Oncilla Robot: A Versatile Open-Source Quadruped Research Robot With Compliant Pantograph Legs

Alexander T. Spröwitz [1,2]*, Alexandre Tuleu [1], Mostafa Ajallooeian [1], Massimo Vespignani [1], Rico Möckel [1,3], Peter Eckert [1], Michiel D'Haene [4], Jonas Degrave [4], Arne Nordmann [5], Benjamin Schrauwen [4], Jochen Steil [5] and Auke Jan Ijspeert [1]

[1] Biorobotics Laboratory, Institute of Bioengineering, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, [2] Dynamic Locomotion Group, Max Planck Institute for Intelligent Systems, Stuttgart, Germany, [3] Department of Knowledge Engineering, Maastricht University, Maastricht, Netherlands, [4] Department of Electronics and Information Systems, Ghent University, Ghent, Belgium, [5] CoR-Lab, Research Institute for Cognition and Robotics, Bielefeld University, Bielefeld, Germany

We present Oncilla robot, a novel mobile, quadruped legged locomotion machine. This large-cat sized, 5.1 kg robot is one of a kind of a recent, bioinspired legged robot class designed with the capability of model-free locomotion control. Animal legged locomotion in rough terrain is clearly shaped by sensor feedback systems. Results with Oncilla robot show that agile and versatile locomotion is possible without sensory signals to some extend, and tracking becomes robust when feedback control is added (Ajallooeian, 2015). By incorporating mechanical and control blueprints inspired from animals, and by observing the resulting robot locomotion characteristics, we aim to understand the contribution of individual components. Legged robots have a wide mechanical and control design parameter space, and a unique potential as research tools to investigate principles of biomechanics and legged locomotion control. But the hardware and controller design can be a steep initial hurdle for academic research. To facilitate the easy start and development of legged robots, Oncilla-robot's blueprints are available through open-source. The robot's locomotion capabilities are shown in several scenarios. Specifically, its spring-loaded pantographic leg design compensates for overdetermined body and leg postures, i.e., during turning maneuvers, locomotion outdoors, or while going up and down slopes. The robot's active degree of freedom allow tight and swift direction changes, and turns on the spot. Presented hardware experiments are conducted in an open-loop manner, with little control and computational effort. For more versatile locomotion control, Oncilla-robot can sense leg joint rotations, and leg-trunk forces. Additional sensors can be included for feedback control with an open communication protocol interface. The robot's customized actuators are designed for robust actuation, and efficient locomotion. It trots with a cost of transport of 3.2 J/(Nm), at a speed of 0.63 m s^{-1} (Froude number 0.25). The robot trots inclined slopes up to 10°, at

$0.25 \, \text{m s}^{-1}$. The multi-body Webots model of Oncilla robot, and Oncilla robot's extensive software architecture enables users to design and test scenarios in simulation. Controllers can directly be transferred to the real robot. Oncilla robot's blueprints are open-source published (hardware GLP v3, software LGPL v3).

1. INTRODUCTION

Emerging technologies allow to improve our understanding of legged locomotion, and its underlying principles. We argue that custom-designed, bioinspired legged machines like the presented Oncilla robot have the potential to provide valuable insights into biomechanics and neuromuscular control of animal legged locomotion. We show initial robot locomotion examples in simulation and hardware, directed with minimal control effort.

Technological progress shapes research, for instance traditional dissection tools are being expanded by computer guided imaging. 3D scans provide spatial data of the animal's morphology even in motion (Walker et al., 2014). Advanced computer simulations give insights into complex interactions between muscles, tendons, and skeletal structures during locomotion (Delp et al., 2007). This allows estimating movements, forces and interactions of otherwise hidden and unaccessible structures in animals. With the help of new technologies, it is the goal to find necessary and sufficient biomechanical and neuromuscular control components for legged locomotion, by identifying form and function.

However, mapping morphologies to function is not trivial. Evolutionary byproducts such as spandrels can mask function of structures (Gould and Lewontin, 1979). The animal legged apparatus developed for locomotion—a complex task with hybrid dynamics. Balancing, carrying and accelerating the animal's body requires precise movement coordination through feedback, and redirection of loads. Forces and movements largely differ in swing and stance phase, separated by harsh impacts at landing, and high velocities during toe off (Söhnela et al., 2017).

Our approach includes custom-made, bioinspired legged robot hardware, and its locomotion controller. Oncilla robot allows us to tackle locomotion research from a new perspective, and directly evaluate functional morphologies. With robotic setups we have the freedom to implement and test simplified "blueprints," mimicking aspects of animals' biomechanics and neuromuscular control. We understand such "blueprints," as building plans which are transferred from Biology to Robotics, for the purpose of testing and characterizing them. The pantographic leg configuration described by Witte et al. (2003) is an example for a mechanical blueprint, that was transferred from animal biomechanics. It is based on the observation of the mostly parallel orientation of distal and proximal limb segments in three-segmented mammalian legs, during locomotion. Mathematical models of the spinal cord activation i.e., in lampreys, so called "Central Pattern Generators," are an example for a neuromuscular control blueprint. They were hypothesized and observed in animals, and tested and implemented in robots (Ijspeert et al., 2007).

Robotic implementations can swiftly be altered, i.e., to test effects of scaling, while longitudinal animal studies are significantly more time consuming. Testing features in robot hardware allows one to quantify function; components' states (position, velocity etc.), forces linking components, and internal loads can be measured with dedicated sensors. Signals for locomotion control and sensing are observable, and mostly controlled. In contrast, recording of deep tissue movements and displacements, and their interacting forces in animals is often not feasible. At present recording of animal nerve signals is limited to simpler setups (Daley et al., 2009).

Computer simulated models of legged animals and legged robots forcibly introduce simplifications, compared to real world physics. Several aspects of physics are notably hard to simulate both precisely and efficiently: leg-ground impacts, wobbling masses, damping, compliance and friction (Schmitt and Günther, 2011). Correctly simulating a legged animal or robot walking through soft substrate is difficult (Li et al., 2009; Falkingham and Gatesy, 2014). In comparison, robotic hardware can be a realistic physical (as opposed to numerical) model of an animal—legs and feet are physically interacting with locomotion substrates through contact forces. Still, computer models are immensely insightful, also because they are relatively easy to derive and apply. Here, we derived a computer simulated Oncilla robot model to rapidly test controller parameters, and create locomotion data.

A future iterative robotic approach to legged research can be composed of four steps: (a) implementing blueprints of mechanics and control of legged animals, in a legged robot system, (b) creating locomotion case scenarios with legged robots, (c) collecting and analyzing locomotion data, and eventually (d) implementing alterations to initial blueprints. If successful, improved blueprints will decrease the discrepancy between robot and animal locomotion characteristics. Changes and observed invariants allow insights into underlying aspects of legged locomotion.

From a design and testing perspective, the cyclic iteration works best if tested blueprints can be altered swiftly. Oncilla robot is designed modularly, i.e., its legs and trunk, actuators, sensors, and controllers are relatively easy to replace. Research in legged locomotion progresses iteratively, and fewer researchers are interested in designing entire robotic systems. The Oncilla robot open source project allows researchers to i.e., test sub-functions of a new locomotion controller. Reproducing parts requires a low level of technology; access to a generic 3D printer, and a basic machining workshop. Components such as brushless

motors are off-the-shelve items. Printed circuit board layouts and firmware code are included in the project source code. If required, they can be replaced by custom solutions—Oncilla robot's custom interfaces are open sourced.

A novice user requires a legged robot that is easy to start up and run. A robot with a large basin of robust locomotion patterns is therefore beneficial. Ideally, the robot's locomotion characteristic would be sufficiently "docile". At the same time, the robot should have the potential for complex gait scenarios, such as rapid acceleration, turning, and locomotion on slopes.

Oncilla robot features a spring-loaded leg design enabling simplified locomotion control. As a consequence, this class of legged robots including "Bow-leg robot," "Cheetah-cub robot," "Bobcat-robot," and "Ken" (Zeglin, 1999; Narioka et al., 2012; Khoramshahi et al., 2013; Spröwitz et al., 2013) can run and hop with feed-forward (FFW) control and does not require model-based control or feedback mechanisms, for basic locomotion on flat terrain. These robots run in FFW mode at comparatively low control frequencies: Spröwitz et al. (2013) report RC servo motor control with as little as $f = 50$ Hz control frequency, also in case of step-down perturbations during higher speed running.

In comparison, robots with leg designs like MIT Cheetah (Seok et al., 2013) or ANYmal (Hutter et al., 2016) actively extend leg joints against gravity. Actively extending legs require dedicated controllers for leg length force control (Hyun et al., 2014).

With Oncilla robot we extend the capabilities of its robot class with its passively, spring extending legs; the robot turns rapidly, with minimal turning radii, walks slopes up and down, and outdoors on natural substrates fully mobile, with low computational effort and without the need for sensory feedback.

The remaining paper is organized as follows. In the next section, we briefly present related work. Section 2 describes the robot's mechanical hardware. Section 3 explains the software architecture of Oncilla robot. Results from hardware and simulation experiments are shown in section 4. We discuss design, results and future directions and provide links to the open-source files in section 5, and conclude in section 6. The Supplementary Material provides further details on robot design, kinematics, and experiments.

2. MECHANICAL HARDWARE

This section explains Oncilla robot's design, and its actuator design process. Details are provided for its leg design, the active degree of freedom (DOF) for steering, force sensors, chassis design, and electronic circuits.

2.1. Physical Robot Dimensions
Oncilla robot has the size and weight of a large house cat (**Table 1**). It is 0.4 m long in total, with 0.23 m between front and hind legs, has a maximal hip height of 0.18 m with its legs hanging in-air, a hip standing height of 0.16 m, and an overall height of 0.31 m, from bottom to handle (**Figures 1**, **2**). The robot has 0.14 m of lateral spacing between left and right adduction/abduction axes, and a maximum width of 0.25 m. Total weight is 4.5 kg without, and 5.1 kg with a 4,500 mAh

TABLE 1 | Physical parameters of Oncilla robot, leg masses without adduction/abduction (AA) motors. Abbreviations: front hip (FH), fore-aft (FA) plane, lateral plane (LAT), vertical plane (VERT), leg length (LL), gear box (GB).

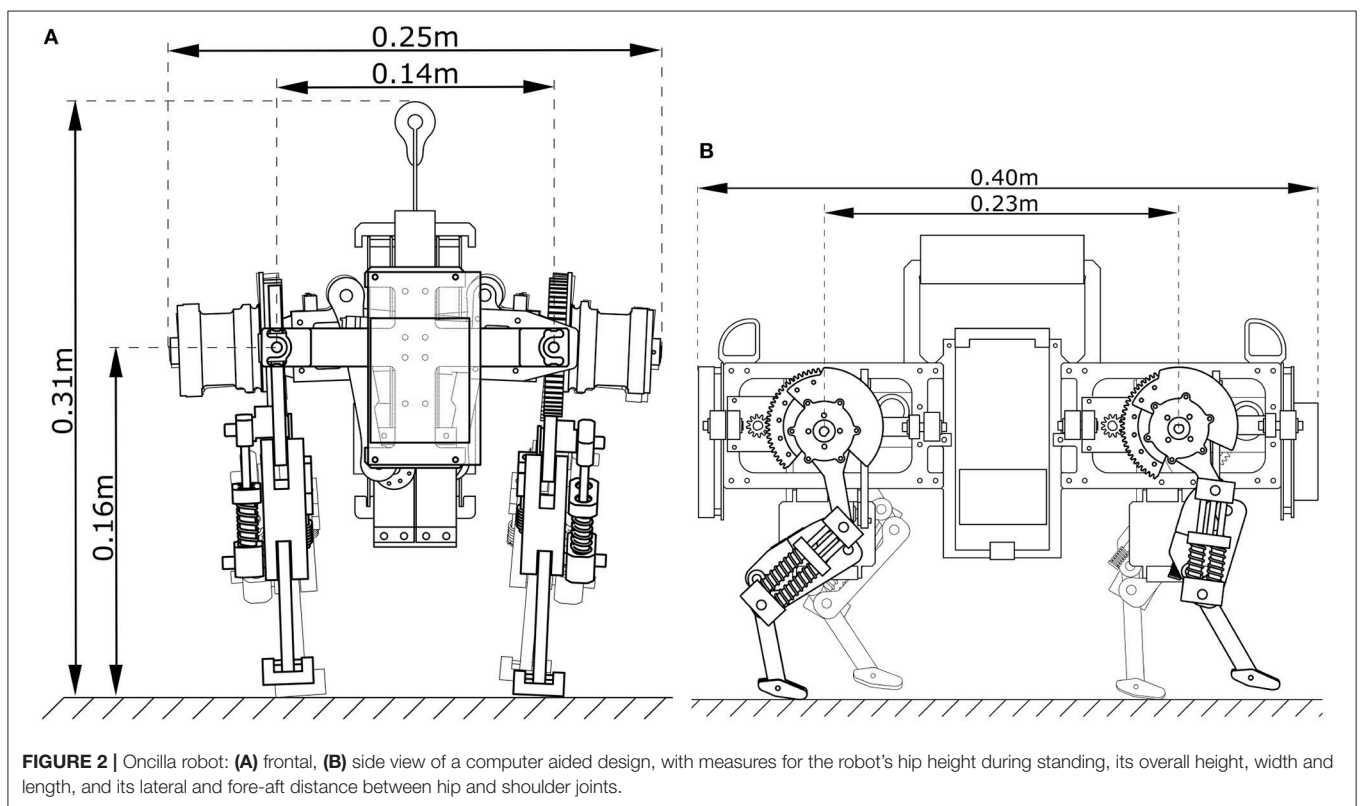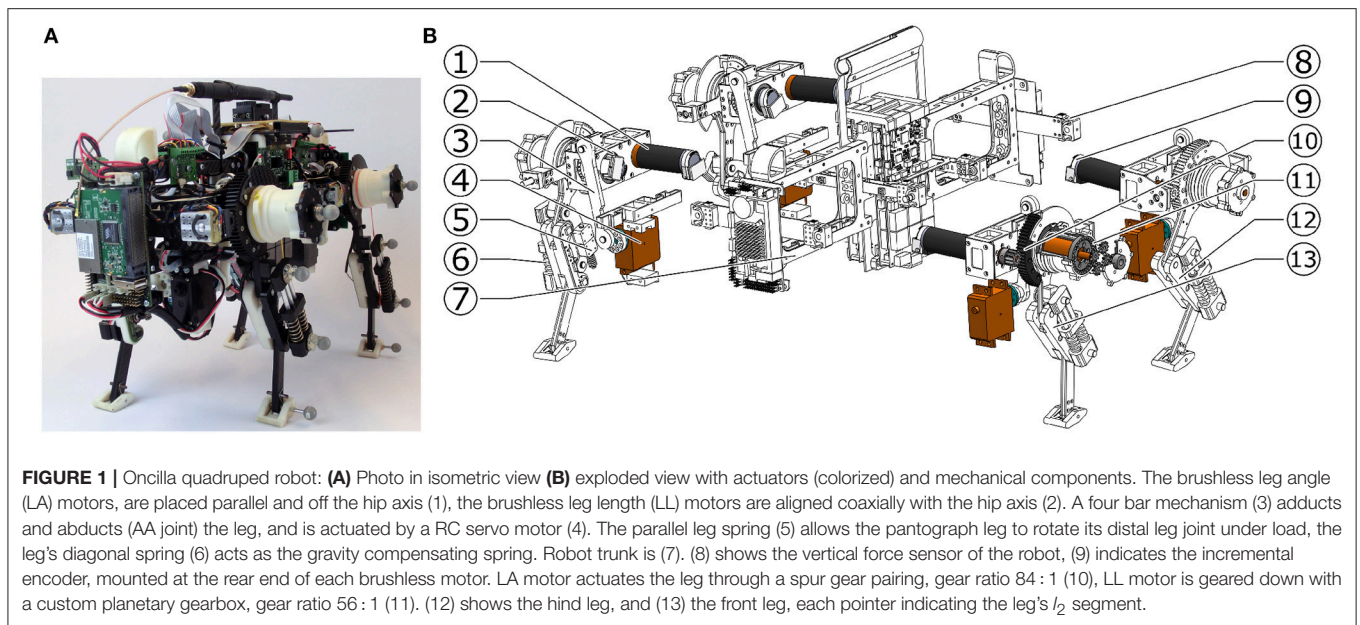| Parameter [unit]	Value
Overall length/height/width [m]	0.40/0.31/0.25
Leg length min/standing/max [m]	0.11/0.16/0.18
Hip spacing (FA) [m]	0.23
Hip spacing AA axes lateral [m]	0.14
Leg angle range (FA) [°]	±34
Leg angle range AA [°]	±8
Robot mass without/with battery & cables [kg]	4.5/5.1
COM position vertical [m]	−0.036
COM position FA, from FH [m]	−0.030
Leg mass without/with actuators [kg]	0.22/0.6
Hip/LA torque at 6 A at GB ratio $n = 84$ [Nm]	7.1
LL torque at 6 A at GB ratio $n = 56$ [Nm]	4.7
Hip (AA) stall torque [Nm]	2.9

Center of mass (COM) position relative to the robot's geometric center, based on the computer aided design.

lithium polymer battery pack. The trunk was designed with a center of mass (COM) 36 mm below the hip-shoulder axis. Each leg weighs 0.2 kg without, and 0.6 kg including leg angle and leg length actuators.

2.2. Actuator Design
Choosing a robust actuator consisting of motor and gearbox is not trivial, when designing robots for agile legged locomotion. Here we only considered electromagnetic motors, for their ease of use in laboratories.

Mobile robots carry their own weight. This includes the power source, i.e., in form of batteries, but also printed circuit boards (PCB), actuators, chassis, and sensors. To reduce power consumption and to increase agility and load carrying capability, low weight and efficient actuators are beneficial. At least two DOF per leg are required for versatile legged locomotion. The leg length (LL) DOF inserts forces to maintain vertical posture. This LL DOF is also used for leg flexion during swing phase. The sagittal leg angle (LA) DOF supports movements and torques in fore-aft direction, and retracts and protracts the leg. Legged robot locomotion is a harsh application for electrical motors. Oscillating locomotion patterns at high cycle frequencies lead to high velocities in both directions and at high peak torques. Those present "external loads." Motor and gearbox combinations lead to additional "internal loads," and can largely exceed external, required loads (Roos et al., 2006). This effect showed strongly with Cheetah-cub robot, a small quadruped robot actuated by high-geared RC servo motors. RC servo motors would work well for 1 Hz motion frequencies. However at 4 Hz motion frequencies and larger amplitudes, the over 300 : 1 gearbox ratio led to overheating problems. Only brief experiments with cool-down phases were possible in a repeatable manner (Spröwitz et al., 2013).

FIGURE 1 | Oncilla quadruped robot: **(A)** Photo in isometric view **(B)** exploded view with actuators (colorized) and mechanical components. The brushless leg angle (LA) motors, are placed parallel and off the hip axis (1), the brushless leg length (LL) motors are aligned coaxially with the hip axis (2). A four bar mechanism (3) adducts and abducts (AA joint) the leg, and is actuated by a RC servo motor (4). The parallel leg spring (5) allows the pantograph leg to rotate its distal leg joint under load, the leg's diagonal spring (6) acts as the gravity compensating spring. Robot trunk is (7). (8) shows the vertical force sensor of the robot, (9) indicates the incremental encoder, mounted at the rear end of each brushless motor. LA motor actuates the leg through a spur gear pairing, gear ratio 84 : 1 (10), LL motor is geared down with a custom planetary gearbox, gear ratio 56 : 1 (11). (12) shows the hind leg, and (13) the front leg, each pointer indicating the leg's l_2 segment.

FIGURE 2 | Oncilla robot: **(A)** frontal, **(B)** side view of a computer aided design, with measures for the robot's hip height during standing, its overall height, width and length, and its lateral and fore-aft distance between hip and shoulder joints.

To improve actuation compared to Cheetah-cub robot, we simulated external load case scenarios for LL and LA actuator, based on a simplified foot locus. We assumed cycle frequencies between 0.5 and 3.5 Hz, calculated for a trotting quadruped with Oncilla robot's dimensions. This external load case data was piped through an actuator optimization framework (Roos et al., 2006), creating a simplified-load dynamical motor model. From this, we chose gearbox ratios in LL (56 : 1) and LA (84 : 1) direction, for a given brushless motor. Details for the external and internal load case simulation setup are provided in supplementary documentation. With an estimated 80 % effective stride length, the modeled robot's cost of transport was
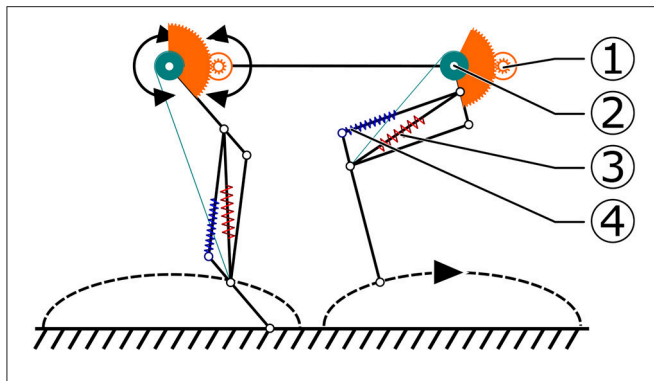
FIGURE 3 | Schematic presentation of Oncilla robot's foot locus movement, created for the simplified-load dynamic-motor (SLDM) model scenario. The SLDM model was applied in the robot's pre-design phase, to estimate required motor and gearbox characteristics. This foot-locus profile was used to calculate leg length (LL, 2) and (LA, 1) loads, for trot gait. The diagonal, gravity compensating leg spring (red, 3) is compressed by flexing the leg through a cable mechanism. Load dependent displacement of the parallel spring (4) during stance phase was ignored in the SLDM model.

calculated (Tucker, 1970). Results are shown in **Figure 7**. The model predicts an asymptotically declining COT, with 7.8 J/(Nm) at a low speed of $0.05 \, \mathrm{m\,s^{-1}}$, a lowest COT of 2.4 J/(Nm) at $0.41 \, \mathrm{m\,s^{-1}}$, and a slightly increasing COT at higher speeds [2.7 J/(Nm) at $0.71 \, \mathrm{m\,s^{-1}}$]. We later recorded the hardware robot's instantaneous power during locomotion, which shows a good match. While underestimating the hardware COT by about 25 %, the simplified-load model did qualitatively predict the asymptotically declining COT characteristics of the hardware robot.

2.3. Leg and Foot Design

Oncilla robot's leg design is a continuation of Cheetah-cub robot's. One active DOF was added for adduction/abduction ("AA joint"), to give the robot steering capabilities. A set of springs with stiffness 5.8 N/mm extends the pantograph leg (red "diagonal spring," **Figure 3**). One pantograph-segment is replaced with a tensile spring (blue "parallel spring," stiffness $k = 7.4$ N/mm). This spring allows flexion of the distal leg segment under load. The leg extends by 7 cm, from its shortest length i.e., during mid-swing, to a fully extended leg length of 18 cm in air. A short foot is mounted, spring tensioned by a torsion spring ($k = 1.21 \, \mathrm{Nmm/°}$). Absolute position encoders are placed at hip axes, and two leg joints (Supplementary Material, Figure S7, q_0, q_1, q_2). The difference between angles q_1 and q_2 indicates the external load torque acting at q_2, i.e., this joint sensor also acts as analog contact sensor.

2.4. Leg Length and Leg Angle Actuation

The robot was designed with mechanical improvements in mind compared to Cheetah-cub robot: increasing load carrying capabilities, incorporating an additional DOF per leg for efficient and fast turning, adding multiple modes of sensing, and adding batteries and on-board power for mobile application. Each additional feature increased the weight, from a 1.1 kg Cheetah-cub robot to a 5.1 kg Oncilla robot. For Oncilla robot's larger

weight, a stiffer leg spring was mounted, which in turn required a higher-torque actuator for the leg flexing mechanism. The LL actuator is mounted serially to the leg angle actuator, and shortens the leg through a cable (**Figure 3**). This design has positive consequences: (a) The LL actuator is decoupled from external forces, such as sudden impacts. Hence, the mounted gearbox requires a lower safety rating, and can be designed with a lower module gear design. (b) During stance phase, parallel leg springs act passively, i.e., the design is a close approximation of the mechanism underlying the spring loaded inverted pendulum (Blickhan, 1989, SLIP template). The sagittal leg angle actuator is directly attached to the proximal leg segment. The motor's long and narrow shape blocked mounting left and right leg motors in line. Instead we applied a non-symmetrical actuator placement, with legs relatively close to each other (0.14 m, **Figure 2** and Figure S6). The short distance leads to a short moment arm i.e., for diagonally touching legs, and is meant to reduce rolling motions. Leg angle motor and hip axis are connected by a large module spur gear (**Figure 2**). Relative encoders are directly mounted to both brushless motor axes. One absolute encoder is mounted at the main LA axis. Incremental motor encoders support precise motor control, and absolute encoders read leg segment positions, and leg spring loading.

2.5. Hip Adduction and Abduction Joint

The added hip adduction and abduction (AA) actuation was installed for efficient turning. The chosen AA actuator is a strong, position controlled RC servo motor (Kondo KRS 2350 ICS). It was selected for its compact design, high holding torque, and standardized control interface. The servo horn connects through a four bar mechanism with the leg, with a movement range of ±8° (**Figure 2**).

2.6. Force Sensors

Custom-made, two-axis force sensors were implemented proximally, within the AA suspension between trunk and legs (⑧, **Figure 1**). Force sensors were designed based on double cantilever bending beams, with foil strain sensing resistors in full bridge configuration. Sensors measure forces in vertical and fore-aft direction, in the trunk's coordinate system. Oncilla robot's force sensors were not utilized for the gaits shown in this work. Instead, all gaits shown are generated in feed-forward mode. Force sensing is however necessary for either model-based control based on leg loading information, or non-model based controllers with i.e., reflex-like feedback (Righetti and Ijspeert, 2008). Because similar locomotion experiments are planned with Oncilla robot, custom designed force sensors are included here.

We considered two mounting places: distally, i.e., as feet, or proximally, between the robot's trunk and legs, as finally chosen. When mounting force sensors distally, almost no effects from unsprung masses are measured, but direct contacts between leg and ground. Consequently, the resulting force signal requires less noise filtering. However, distally and foot-mounted sensors rotate with the foot frame. In case the direction of force is of interest for control, the sensor's orientation should be recalculated through the serial chain of trunk, leg segments, and foot sensor. Mechanically, a distal sensor placement moves the leg's center

of mass further distally, especially for larger and more complex sensors. It also limits mounting spring-loaded feet. To cope with harsh touch down impacts, a miniaturized but also robust sensor design is required.

In comparison, proximal force sensor mounts are relatively independent from leg and foot design. In our case sufficient mounting space was available proximally. Standard-sized, strain gage-based sensors were chosen and implemented. Drawbacks exists; proximally sensed force signals are influenced by mass and inertia of the moving leg, and data post-processing is required. Latter will introduce sensor signal delays. Such delays can present an obstacle especially for fast loop locomotion controllers. Proximal force sensors require no recalculation of the sensor's orientation, as they are fixed in the trunk frame.

2.7. Electronics and PCB Mounting

Oncilla robot carries its actuators and sensors, a battery pack, and PCBs for motor control, power supply, and communication. Brushless motor driver boards were custom designed, and each of four boards provides control and power for two brushless motors. One board weighs 0.15 kg, and all are mounted low, at the robot's geometric center. This placement helps keeping the robot's COM below the virtual hip-shoulder axis—the robot's trunk "hangs" in-between. The remaining PCBs are placed at the front of the robot (RB-110, main computing and communication), and at the rear end (power supply), for easy access. An inertia measurement unit (MicroStrain 3DM-GX3-35 IMU) is mounted above the motor driver boards. Further details on the electronic layout, components, operation system are available in the Supplementary Material (i.e., Figure S9).

3. SOFTWARE

In this section we describe software related concepts developed for the Oncilla robot; namely the communication bus connecting main control boards and periphery, the motor controller, Oncilla robot's software architecture, and the simulated Oncilla robot model in Webots.

3.1. Communication Layer and Protocol

Communication between main electronic modules is performed over a RS-485 physical layer, organized in a master/slave point-to-multipoint half-duplex configuration (**Figure 4**). The data link and network layer on the bus is implemented through a custom Simple Binary Communication Protocol (SBCP). Latter implementation is an extension of the Bioloid Dynamixel Communication Protocol version 1 (Robotis Ltd., Co, 2018). The main modification is the change of the two byte packet preamble (value 0xFFFF). This preamble was separated; one single byte preamble (0XFF), and a ClassID byte to designate the device type we want to address, i.e., the motor driver board, power board, or master control board. By reserving the class value for Bioloid devices, Oncilla robot's hardware and driver are capable of directly incorporating these.

A major challenge in the SBCP design was the high communication bandwidth requirement. We initially aimed at a 1 kHz control loop on the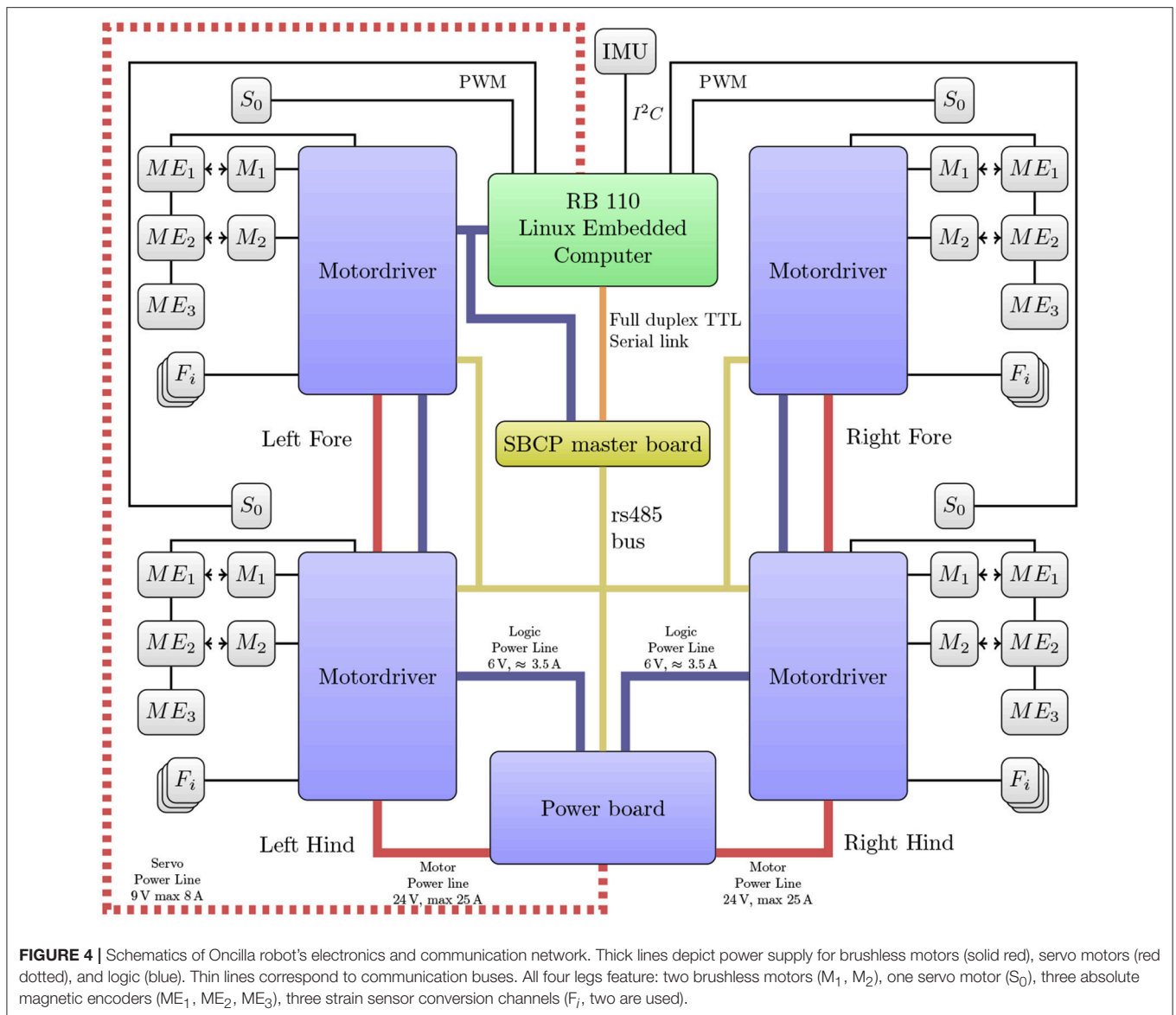 embedded computer level, this translates to a RS-485 baud rate of 3.3 Mbps. Recently such high speed UARTs became available in embedded computers (< 12 Mbps at the moment), through an USB to serial Integrated Circuit (IC). However, USB bus communication is scheduled by 1 ms frame sizes, and a naive bus implementation would be limited to reach a device every 2 ms. Instead, we implemented a bus control data flow in a dedicated device (SBCP master board) controlled by a dsPIC33FJ128MC802 Digital Signal Processor (DSP), mounted after the USB to serial IC. This SBCP master board is able to: a) Handle a group of up to 8 combined packets sent over a full duplex UART connection. b) Manage communication over the half duplex RS-485 bus with a latency of 12 μs and maximal jitter of 2.2 μs. c) Detect slave timeouts and react accordingly with user customizable delays. To reduce jitter and latency to a minimum, we utilized many optimization features of the dsPIC33FJ processor family such as Direct Memory Access (DMA), reduction of interruption stress on the processor, and anticipated packet precomputation and pre-buffering to reduce latency between packet responses. All implementation details are abstracted by a reusable library, SBCP-uc 4. This new open source library enables fast development of slave device SBCP interfaces for this processor family.

3.2. Motor Control

The custom designed motor driver boards implement a Proportional-Integral-Derivative (PID) controller for two brushless motors on a dsPIC33JF128MC804 DSP. It controls two A3930 motor driver ICs which are used to drive a three phase inverter made of six IRFR48Z MOSFETs. These ICs are able to limit the motor winding current. DSP PID controller parameters were hand tuned. In order to reduce motor jitter during position tracking, a velocity profile interpolation is implemented on the motor driver DSPs. The user sets the desired tracking frequency, and the tracking velocity is then computed on-board. Initially we observed trajectory tracking losses every several seconds. We found the cause in a combination of communication jitter and real-time clock frequency mismatch. Tested RB-110 boards had crystal frequency deviations of up to +0.3 %, from their standard frequency. As a workaround, we implemented precise buffering and real-time clock re-synchronization heuristics on the motor driver DSP. The maximum tracking frequency is 500 Hz, half of the DSP internal control loop (1 kHz).

3.3. Software Architecture

The software architecture of Oncilla robot is based on two primary design decisions tied to the requirements of open research projects (Nordmann et al., 2013): (a) It provides a common interface for the simulator and for the hardware. This allows for easy and fast transition of experiments between simulation and real-world experiments. (b) It provides a local interface for fast control loops running on the embedded PC, as well as a remote interface to allow more complex applications to control the robot over the network. **Figure 5** shows the software architecture that exposes the abstracted interface for both simulation and hardware at different application programming interface (API) levels according to application requirements.
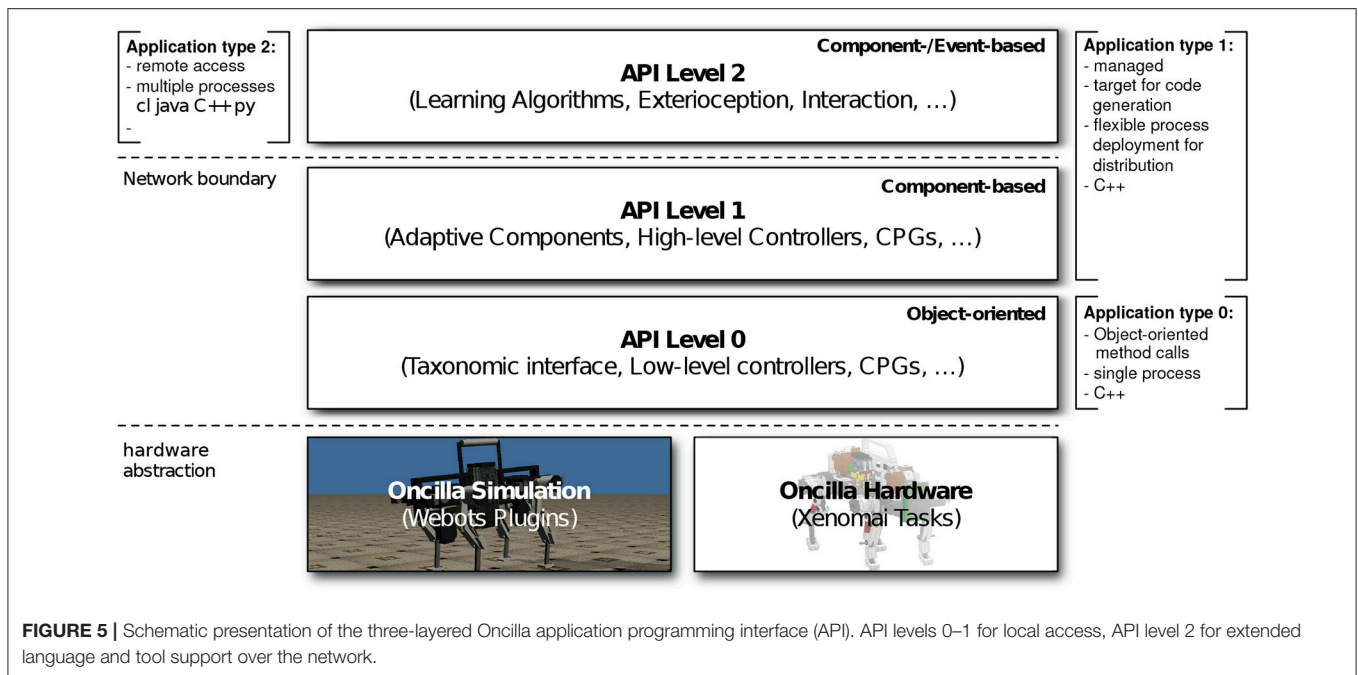
FIGURE 4 | Schematics of Oncilla robot's electronics and communication network. Thick lines depict power supply for brushless motors (solid red), servo motors (red dotted), and logic (blue). Thin lines correspond to communication buses. All four legs feature: two brushless motors (M_1, M_2), one servo motor (S_0), three absolute magnetic encoders (ME_1, ME_2, ME_3), three strain sensor conversion channels (F_i, two are used).

3.4. Simulation and Robot Interface

One of the main design goals was to implement a common abstraction between hardware and simulation, with binary compatibility, to facilitate an easy transfer between hardware and simulation. The abstraction also allows to exchange the currently Webots-based (Michel, 2004) simulation back end. Here we chose Webots software as our multi-body simulation software for Oncilla robot. Since the lowest API level 0 is implemented for the simulation and the hardware back end, applications implemented against any of the API level 0-2 can be switched transparently between simulation and hardware (**Figure 5**). By providing a common abstracted interface for both simulation and hardware, it enables fast and easy transfer of experiments between these two domains. It is also possible to replay real experiment recordings (e.g., joint angles) in simulation, and vice-versa.

3.5. Local and Remote Interface

The Oncilla interface is accessible through a local C++ interface (API levels 0 and 1, **Figure 5**) and remotely via the open-source middleware for extended language and tool support (API level 2). The low-level sensors and actuators are locally accessible through a C++ interface, using multiple inheritance to expose the node taxonomy. The local interface enables light-weight applications with fast sensor feedback running on the embedded PC without dealing with network latency. The Oncilla interface is also remotely available by the open-source middleware Robotics Service Bus (RSB) with C++, Java, Python, and Common Lisp bindings and therefore allows applications in all four languages (Wienke and Wrede, 2011). This enables extended tool support, e.g., external logging, monitoring and recording of experiments. It supports more complex and computationally expensive applications to run on distributed

FIGURE 5 | Schematic presentation of the three-layered Oncilla application programming interface (API). API levels 0–1 for local access, API level 2 for extended language and tool support over the network.

PCs and control of the robot over the network. This was successfully applied and demonstrated in experiments that were specified in domain-specific languages. The experimental source code was automatically generated, as well as machine learning applications that can not run on the embedded PC due to resource limitations (Nordmann et al., 2013).

3.6. Webots Model of Oncilla Robot

A simulated model of Oncilla robot was created in Webots (Michel, 2004). It is based on the mechanical properties extracted from Oncilla robot's computer aided design (CAD) model, i.e., weight, center of mass, inertia, link dimension, and spring constant. The motivation to use a physics engine intended for games such as Open Dynamic Engine (ODE, physics engine of Webots) is a balance between simulation speed and required simulation accuracy. Internal kinematic loops in Oncilla robot's pantograph leg, and its asymmetric actuation can be expressed as a constraint in the underlying Linear Complementary Problem (LCP). There is a drawback to this approach; Open Dynamic Engine favors stability over accuracy, which results in a poor constraint resolution. Likely, such a simulator is well suited for fast prototyping, but less suited for on-board, model-based control. We used the default Webots actuator model for the proximal joints (hip and shoulder). This model applies a PID controller, with a maximum output velocity and torque limited by the theoretical motor limits, i.e., a maximum torque of $53.5 \, \text{mNm} \cdot 84 = 4.5 \, \text{Nm}$ and a maximum speed of $16{,}300 \, \text{rmp} / 60 \, \text{s} \cdot 2 \cdot pi / 84 = 20.3 \, \text{rad/s}$ (53.5 mNm constant motor torque, overall gear ratio $n = 84$, 16,300 rpm no-load motor speed). For the leg's flexion and extension DOF we are directly manipulating the LCP constraint to control the mechanical stop of the diagonal spring. Since there is no easy way to specify the maximum torque or linear force in ODE to satisfy a given constraint, we

transformed this force constraint into a velocity constraint using a linear motor model with internal resistance. We simplified and assumed that the motor torque required for actuating the leg length cable was caused by the compressive force of the diagonal spring. In reality, this torque is an upper bound since external forces could induce additional leg flexion. We further limited the maximum speed at which motors move to the mechanical stop constraint for the next simulation step, proportionally to the instantaneous torque. This limit is ranging from maximum speed with no torque required, to zero speed at maximum motor torque requested. The resulting model corresponds to a simple, linear motor model. Finally an implementation of the C++ interface level 0 back end was developed. It provides the user with the ability to use the same API to seamlessly switch control between either the real hardware or the simulated robot.

4. EXPERIMENTS AND EXPERIMENTAL RESULTS

This section describes experimental results with the hardware Oncilla robot utilizing the here described open-loop controller for straight level locomotion indoors, locomotion descending and ascending slopes, and turning strategies with and without AA joints. Links to videos of Oncilla robot locomotion are provided in Table S1 . Further, we provide experimental results with the simulated Oncilla robot running in Webots. Advanced experiments with Oncilla robot's closed loop control framework are documented in Ajallooeian (2015).

4.1. CPG Controller for Straight Locomotion

For the robot's locomotion control, we applied morphed oscillators (Ajallooeian et al., 2013c), to implement a Central

Pattern Generator model (Ijspeert, 2008). Morphed oscillators are nonlinear oscillators which can encode arbitrary limit cycles defined as phase-dependent functions. Given a desired joint trajectory, a morphed oscillator can be implemented to encode this trajectory as a stable limit cycle. This provides a smooth trajectory generator with the capability of feedback integration. A morphed oscillator utilizes a simple oscillator as base and morphs it to obtain the desired limit cycle behavior. Here we use a unit radius amplitude controlled oscillators as base:

$$\dot{\theta}_i = \Omega_i \tag{1}$$

$$\dot{r}_i = \Omega_i f_i'(\theta_i) + \gamma \left(f_i(\theta_i) - r_i \right) + \xi_i \tag{2}$$

$$\Omega_i = \omega + \sum_{j=1}^{N} c_{ij} \sin(\theta_j - \theta_i - \phi_{ij}) \tag{3}$$

where θ_i, Ω_i and r_i are the phase, the coupling dynamics, and the radial output of the ith oscillator, respectively. γ is the rate at which the dynamics converge to the limit cycle, ω is the locomotion frequency multiplied by 2π, and c_{ij} and ϕ_{ij} are the coupling strength and phase difference between the ith and jth oscillator. The phase difference is exploited to implement inter-joint coordination. $f_i(\theta)$ defines the shape of the limit cycle of the ith oscillator and $f_i'(\theta) = \partial f_i(\theta)/\partial\theta$. ξ_i is the additive feedback. It can be designed through strategies explained in Ajallooeian et al. (2013b,c). r_i, the time-integration of $\dot{r}_i$, is the joint angle reference for the ith DOF.

To design locomotion gaits, we define foot trajectories with respect to the hip frame, similar to Maufroy et al. (2010). We use simplified closed-form inverse kinematics to convert those to joint trajectories. These joint trajectories define $f_i(.)$ functions. The definition of ϕ_{ij} is gait dependent, for example for a trot gait $\phi_{ij} = \pi$ for adjacent hips, and $\phi_{ij} = 0$ for diagonal hips. Finally, $c_{ij} = 5$ for all the oscillators. We applied solely open-loop gaits in this work. However, Oncilla robot's hardware and software architecture allows to utilize the robot's internal sensors and apply closed-loop control, i.e., with reflexes and posture control (Ajallooeian, 2015).

4.2. Level Trotting

Figure 6 shows experimental data of Oncilla robot trotting forward at an average speed of 0.55 m/s (Fr = 0.19), on level terrain, on a standard laboratory surface and with the CPG controller described in the previous section, in open-loop mode. Kinematic data was recorded at 240 Hz with a motion capture system (Naturalpoint, Inc., 2011). Position and velocity data of the robot are shown, ordered by its left-right, fore-aft, and its up-down direction. The robot's center of mass (COM) oscillated vertically about ±5 mm, left-right velocity stayed in the range of ±0.05 m/s. Peak forward speed of this recording was 0.78 m/s. The recorded roll angle during trotting stayed symmetrically around 0 ± 0.02 rad (0°±1.2°), the pitch angle of the robot oscillated around 0.06 ± 0.04 rad (3.4° ± 2.3°).

In further tests, Oncilla robot's best forward trotting speed at 3.5 Hz was v = 0.63 m/s. This is equivalent to 2.7 body

lengths per second, with a body length of 0.23 m (shoulder-to-hip distance, 2). The best average speed backwards was 0.78 m/s (3.4 BL/s), at 4 Hz locomotion frequency. Backwards trotting with Oncilla robot showed less slippage, leading to a larger effective stride length. Forward locomotion frequencies above 3.5 Hz yielded no speed gain, due to slippage. Links to videos of the robot trotting both indoors and outdoors (gravel, step-down, flat terrain) are provided in Table S1.

Cost of transport (COT) was measured on the tethered robot (m = 4.5 kg). Stand-by power consumption with no actuator movement was subtracted from all runs (19.6 W), the remaining power consumption (P) was used to calculate the cost of transport (Tucker, 1970): $COT = \frac{P}{mgv} = [J/(Nm)]$ with g = 9.81 m/s², and average speed v over at least 4 cycles. Although COT is often given without units, we use [J/(Nm)], to avoid mix-up with COT values given in [J/(kg m)]. Froude numbers are calculated as $Fr = v^2/(gl)$, with l = 0.16 m standing hip height. Duty factor values are given as ratio of leg stance to cycle time.
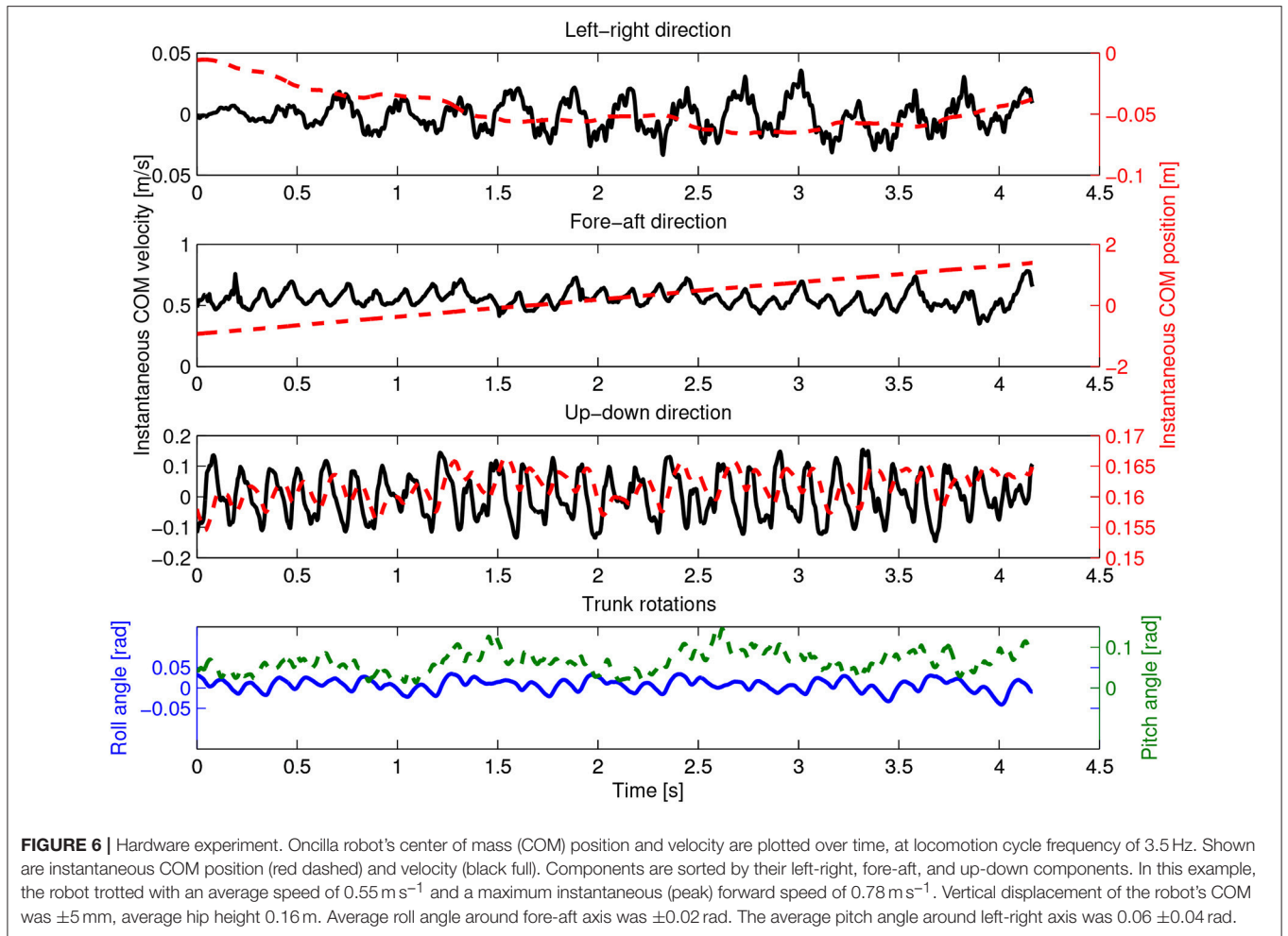
Figure 7 displays recordings for speed and cost of transport (COT) for the simplified modeled ("SLDM," section 2.2) and the hardware ("real") Oncilla robot, during forward (square markers) and backward (round markers) trotting locomotion, over 5 tested speeds. At very low speed (0.07 m/s) the robot trotted forward with a high COT of 20.4 J/(Nm). It reached its best forward COT at its maximum recorded speed on level terrain: 0.63 m/s with a COT of 3.2 J/(Nm). Backwards locomotion was more efficient at low speed, with a COT of 9.6 J/(Nm) at Fr = 0.01. At higher backwards speed (0.63 m/s) the robot trotted with a COT of 3.8 J/(Nm).

Figure 7 shows the SLDM model underestimates the real robot's COT. Considering its simplicity, the SLDM model does approximate the hardware robot's COT well, with 2.8 J/(Nm) at 0.71 m/s. The model also qualitatively captures the asymptotic decrease of COT over speed.

4.3. Slope Up and Down Locomotion

Table 2 illustrates results for locomotion on inclined surfaces, and **Figure 8** depicts a slope descending locomotion in forward direction. All slope experiments were conducted without changes to the standard level-trotting locomotion controller, or changes to the hardware. This allows better comparison between level and slope trotting, although adapting the foot friction and gait patterns, and applying closed loop control would improve the robot's speed (Ajallooeian et al., 2013b). For the experiment documented in **Table 2**, Oncilla robot trotted forward and backward, level up and down slopes up to 10°.

For inclinations of more than 4°, the robot could only climb slopes when going backwards. The highest inclination climbing was recorded at 10°, with the robot going backwards at 0.25 m/s, at 0.4 m/s commanded speed. The robot trotted forward onto slopes of 4°, with a speed of 0.15 m/s. Down, the robot kept the commanded speed of 0.4 m/s when pointed forward. Its speed increased by 5 % when going down backwards (0.42 m/s). From video footage we observed that runs with strong speed deviations coincided with strong robot feet slippage.

FIGURE 6 | Hardware experiment. Oncilla robot's center of mass (COM) position and velocity are plotted over time, at locomotion cycle frequency of 3.5 Hz. Shown are instantaneous COM position (red dashed) and velocity (black full). Components are sorted by their left-right, fore-aft, and up-down components. In this example, the robot trotted with an average speed of $0.55\,\mathrm{m\,s^{-1}}$ and a maximum instantaneous (peak) forward speed of $0.78\,\mathrm{m\,s^{-1}}$. Vertical displacement of the robot's COM was $\pm 5\,\mathrm{mm}$, average hip height $0.16\,\mathrm{m}$. Average roll angle around fore-aft axis was $\pm 0.02\,\mathrm{rad}$. The average pitch angle around left-right axis was $0.06\pm 0.04\,\mathrm{rad}$.

4.4. Turning Maneuvers

We utilized two strategies to implement turning. For the first method (adductor/abductor amplification: "AA-amp"), a sine-wave was embedded into the oscillator nodes controlling the adduction and abduction joints. Turning was achieved by setting the amplitude of the fore and hind AA joints ($a_{l,AA}$) with opposite signs, proportional to the desired turning rate ($\Delta\psi_{yaw,des}$). In AA-amp, turning time is only a variable of AA amplitude.

$$f_{l,AA} = a_{l,AA}\sin(\theta_{l,AA}) \quad l = 1..4 \quad (4)$$

$$a_{l,AA} = \lambda\Delta\psi_{yaw,des} \quad (5)$$

$$\lambda = \begin{cases} +1 & l \in \{LF, RF\} \\ -1 & l \in \{LH, RH\} \end{cases} \quad (6)$$

For the second method (asymmetrically shortening stride length: "ASL turning"), step length between the left and right leg was modified in order to implement turning, without the use of AA joints. We implemented a turning strategy typically used for two-wheeled mobile robots; the step length was shortened for the legs on one side of the body, to turn in the same direction. The

robot turns in-place if step lengths are equal, but with opposing signs.

$$a_{l,asym} = \begin{cases} 2\varpi + 1 & \varpi < 0 \,\&\, l \in \{LF, LH\} \\ 1 & \varpi > 0 \,\&\, l \in \{LF, LH\} \\ 1 & \varpi < 0 \,\&\, l \in \{RF, RH\} \\ 1 - 2\varpi & \varpi > 0 \,\&\, l \in \{RF, RH\} \end{cases} \quad (7)$$

$a_{l,asym}$ is the step length amplifier, for leg l, and ϖ is the turning factor. This approach produced a small amount of slippage because the ground-contacting legs (diagonal pair of legs, during trot gait) are not on the same axis. Note that in this approach the turning time is a variable of both the commanded forward velocity and the turning factor.

AA amplification was used for most of the turning cases, however fast turning with this method resulted in considerably larger forces at AA joints. The ASL method was mostly used to turn when the ground was uneven. This method does not depend on lateral foot movement, and therefore poses no danger of sideways stumbling against obstacles. Quantitative results from experiments with these two turning strategies are provided in **Table 3**. They show that AA-amp allows to turn on the spot,
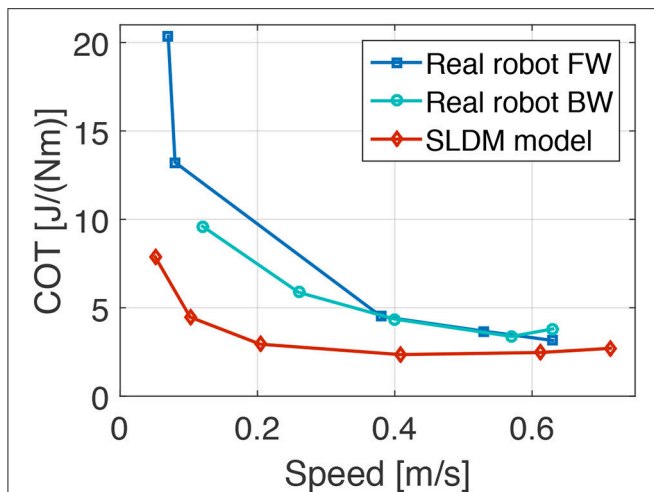
FIGURE 7 | Hardware experiment and results from simplified-load dynamical motor (SLDM) model. Plots show the cost of transport (COT) of the real Oncilla robot and the SLDM model over different speeds. The full power consumption minus the stand-by robot power consumption (19.6 W) was used for the COT calculation. Red diamond (SLDM model) marks show the estimated COT values calculated prior to the construction of Oncilla robot based on a simplified, dynamically articulated robot model. Dark blue data points show the COT-speed values for the real Oncilla robot during level trotting in forward (FW) direction. Round marks indicate the hardware robot's COT during backwards (BW) level trotting. The FW locomotion shows a higher COT up to a speed of $0.4 \, \mathrm{m\,s^{-1}}$, compared to the BW locomotion. The SLDM model continuously underestimates the real robot's COT, but provides a good estimate of the asymptotic decline of COT over speed. The best recorded COT with the real Oncilla robot is 3.2 J/(Nm) at $0.63 \, \mathrm{m\,s^{-1}}$, during FW trotting.

TABLE 2 | Hardware experiment.

Inc [°]	$v_{cmd} \, \mathrm{[ms^{-1}]}$	$v_{avg} \, \mathrm{[ms^{-1}]}$
+4	0.4	0.15
+4	−0.4	−0.33
+7	0.4	ES
+7	−0.4	−0.27
+10	0.4	ES
+10	−0.4	−0.25
−10	0.4	0.40
−10	−0.4	−0.42

Results for locomotion on inclined surfaces of different slopes while ascending or descending. Locomotion on up slopes was more successful when trotting backwards. Abbreviations: commanded velocity (v_{cmd}), average velocity (v_{avg}), inclination (Inc), excessive slippage (ES).

taking 10 s for a full turn. The AA strategy leads to small speed losses during turning, between 20 and 30 %. AA-amp turning radius was between 0.23 and 0.46 m and depended on turning speed. In comparison, the ASL method showed higher speed losses at turning. The best parameter configuration led to a 52 % speed loss, at a turning radius of 0.5 m. The smallest ASL turning radius recorded was 0.03 m. Turning maneuvers, including on-spot turning, are available as Supplementary Videos (Table S1).

4.5. Leg and Force Sensor Characterization

The recorded sensor data of the robot's proximal, trunk-mounted force sensors shows mixed results (**Figure 9**). Especially the horizontal force signal is sensitive to parasitic stresses and strains of the mounting brackets, caused by forces between legs and trunk. To keep its weight low, the robot is designed from lower stiffness materials such as 3D printed plastic. As a consequence, the AA suspension bracket deflects under load, and influences forces sensor readings. In the horizontal force signal, this appears as high-frequency noise. The robot's vertical force sensor showed good signal quality. After calibration and post-processing, vertical reaction forces can be extracted (**Figure 9**), which can be utilized for future feedback-based controllers.

Stance phase timing, and to some extend leg loading can be estimated by observing the difference of leg joint deflection between knee joints, and spring loaded ankle joints ($p_{knee} - p_{ankle}$). **Figure 10** shows the resulting angular signal, with joints being charged periodically at each stance phase. If required, this angular difference signal can be multiplied by the ankle joint stiffness, creating a source for ankle joint torque sensory feedback (not shown here). Importantly, this method is comparatively cheap, both from a hardware and computational perspective. It requires no additional hardware sensing framework, while providing joint angular, stance timing, and loading information.

4.6. Oncilla Webots Simulation

The Webots (Michel, 2004) simulated Oncilla robot allows one to test control algorithms and locomotion scenarios without access to the real hardware. The controller can then be transferred to the robot, with the help of Oncilla robot's software architecture. Certain limitations for a direct transfer of control parameters exist ("one-to-one transfer") because of the "reality gap" between simulated and real robot, those are also explained in the following.

We implemented a Webots gait demonstration with the following parameters: locomotion frequency $f = 3.5$ Hz, desired step length of 12 cm, fore and hind touch down angle around 2.85 rad. To maximize the available leg retraction, a foot lift-up height of 4 cm was set. The virtual, commanded duty factor was 0.49, the observed effective duty factor 0.52 and 0.58 for fore and hindlimb, respectively. Remaining control parameters were optimized with a PSO framework (Kennedy and Eberhart, 1995) to maximize covered distance over a 15 s time interval, after a 5 s time window to reach steady-state. Snapshots of the Webots Oncilla robot trotting are shown in **Figure 11**, the corresponding video link is given in Table S1 of the Supplementary Material.

From results we recognize the relatively large speed gap for forward locomotion between hardware and Webots simulation at equal locomotion frequencies (**Figures 6**, **12**, 3.5 Hz). The hardware robot reached a maximum forward speed of 0.63 m/s, with 0.09 m effective stride length. The Webots model reached a forward speed of 0.98 m/s, with 0.14 m effective stride length. The Webots robot's COM oscillated ±2 mm vertically. The simulated robot showed roll angle oscillating around 0 rad ± 0.015 rad, and pitch angles oscillating around −0.03 rad ± 0.015 rad.

Further analysis of the foot locus shows the optimization framework increased the model robot's speed by making precise
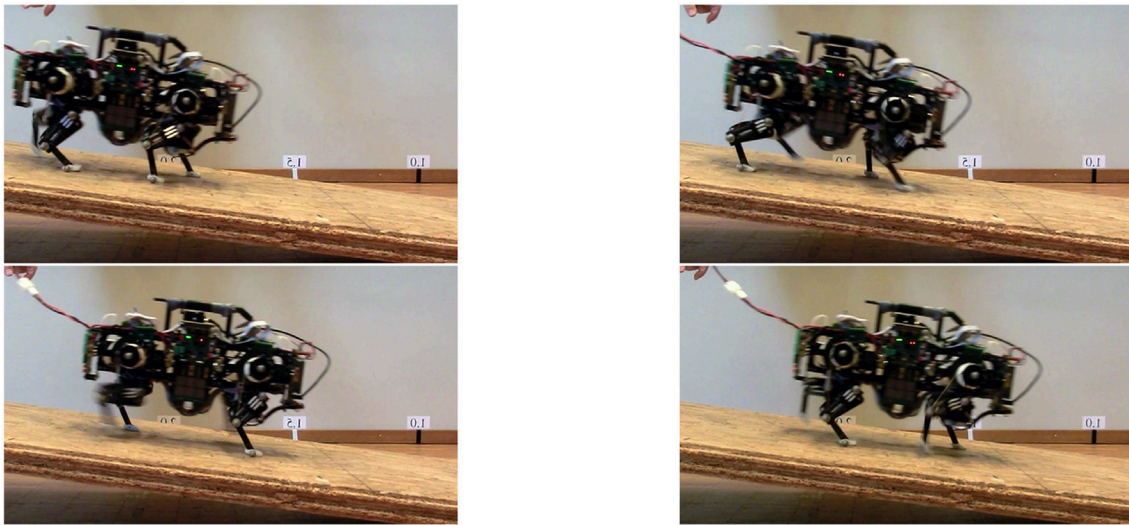
FIGURE 8 | Hardware experiment: snapshots of Oncilla robot descending a slope in forward direction. Further tests were performed with the robot going up the slope, and by letting the robot climbing and descending during backwards locomotion. At steeper slopes, Oncilla robot showed excessive slippage when climbing the slope head on (**Table 2**). Generally, the robot performed better when locomoting backwards. Snapshots here are flipped horizontally, for reading convenience.

TABLE 3 | Results for turning with two different strategies: (a) Amplifying the movement of the shoulder/hip abduction/adduction degree of freedom (AA amp).

Strategy	v_{cmd} [m s^{-1}]	SF []	A_{AA} [rad]	r [m]	t_{FT} [s]	v_{avg} [m s^{-1}]
AA ampl.	0.4	0.0	0.05	0.46	9	0.32
	0.2	0.0	0.05	0.23	10	0.14
	0.0	0.0	0.05	≈ 0	10	≈ 0
ASL	0.4	0.3	0.00	0.50	17	0.19
	0.4	0.4	0.00	0.32	14	0.15
	0.4	0.5	0.00	0.19	9	0.13
	0.4	1.0	0.00	0.03	7	0.03

(b) Asymmetrically shortening the stride length (ASL) on one side of the robot. Scale factor (SF): when 0, stride length on both sides are the same, when 0.5, stride length on one side is zero, and when 1, stride length on one side is reversed. (v_{cmd}), Commanded velocity; (A_{AA}), amplitude of AA movement; (r), turning radius; turn (t_{FT}), time required for a full 360°; and (v_{avg}), average resulting velocity are shown.

use of the robot's toe segment (Figure S1). This increased effective stride by 15 %, from 0.12 m to 0.14 m. In hardware, we observed a decrease of effective stride length, compared to the commended stride length. At frequencies above 3.5 Hz the hardware robot started slipping, and the effective stride length reduced to 0.09 m.

The gap between simulation and real-hardware speed makes it hard to one-to-one transfer control parameters for open-loop gaits. However, in parallel to this work, Oncilla robot's Webots model was successfully applied to prototype a closed loop controller (Ajallooeian et al., 2013a; Ajallooeian, 2015). The transfer of the closed-loop controller onto the hardware was done with minimal effort, and showed a better performance matching compared to the open-loop parameter transfer shown here.

5. DISCUSSION

Oncilla robot is a small, light-weight, quadruped legged robot with compliant, spring-loaded pantograph legs and three active degrees of freedom per leg. The robot trots open-loop indoors and outdoors, with forward speeds up to 0.63 m s^{-1} (Fr = 0.25). It climbs up to 10° slopes at a speed of 0.25 m s^{-1} backwards, and 4° slopes at a forward speed of 0.15 m s^{-1}. All hardware experiments shown are with a single controller, for trotting, in open-loop. The robot descends slopes of 10° at a speed of 0.42 m s^{-1} backwards, and 0.40 m s^{-1} forwards. Oncilla robot's adduction/abduction (AA) joints permit fast turning maneuvers, with smaller turning radii compared to turning maneuvers utilizing only leg angle and leg length joints. Utilizing AA joints enables turning on the spot, with 10 s for a full turn. Furthermore, AA turning allows the robot to turn while mostly maintaining the commanded speed, i.e., the robot would lose as little as 20 % speed while turning on a radius of 0.46 m. Oncilla robot showed best cost of transport (COT) at a locomotion speed of 0.63 m s^{-1}, with COT = 3.2 J/(Nm). A set of sensors is available to monitor the robot's posture, ground contacts, and trunk-leg reaction forces for feedback control. A high bandwidth, real-time (500 Hz) network allows communication between the robot's electronic boards. The robot's main controller runs at 200 Hz in real time. The robot's software architecture allows to transparently control the robot either onboard, or remotely, with multiple programming interface options. It further allows to control hardware and Webots simulation with the same controller architecture. Oncilla robot's multibody dynamics simulation in Webots can be used for fast prototyping controller architectures.

Oncilla robot incorporates several features of Cheetah-cub robot (Spröwitz et al., 2013). We implemented additional turning capabilities through Oncilla robot's AA DOF, with a set of RC servo motors rotating the robot's leg in an axis parallel to the
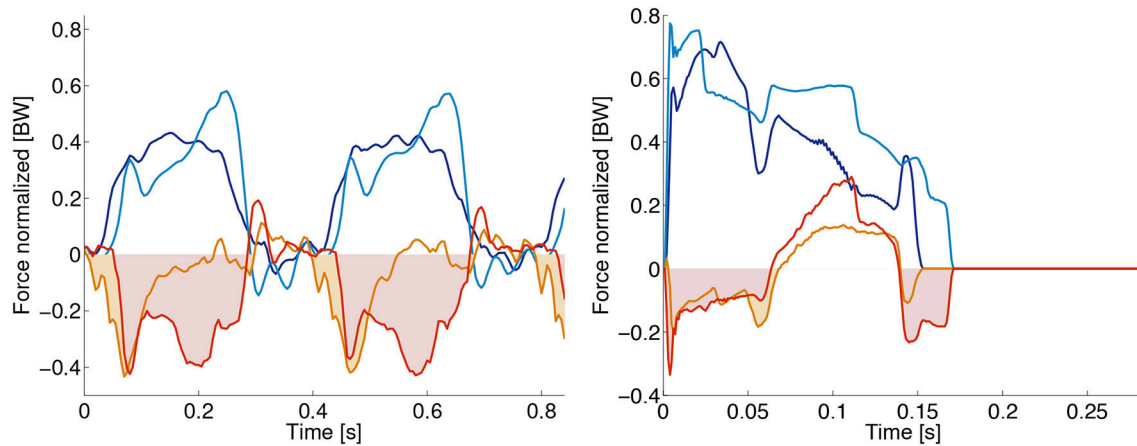
FIGURE 9 | Oncilla robot's force sensor signals for **(A)** the hardware robot, and **(B)** its simulated Webots model. Forces are given in multiples of body weight (BW). Recorded are hind right (HR) and left front (LF) legs. Vertical forces $F_{ver,LF}$: dark blue line, $F_{ver,HR}$: light blue line. Horizontal forces $F_{hor,LF}$: orange line, $F_{hor,HR}$: red line. The hardware robot gait is a 2.5 Hz trot, the simulated robot trotted at 3.5 Hz. Hardware data was post-processed by an offset correction, and a 18 Hz low pass filter. The hardware experiment indicates vertical forces around 0.5 BW. Time-wise integration of horizontal forces over stance phase would be zero at constant velocity trotting. Hardware recorded force data however shows in-sum negative impulse i.e., the data indicates a decelerating robot, while the actual robot was trotting at quasi continuous speed. We assume that offsets are created by internally deflected mounting brackets, around the horizontal force sensors. Positive and negative components of horizontal forces extracted from Webots are about equal.
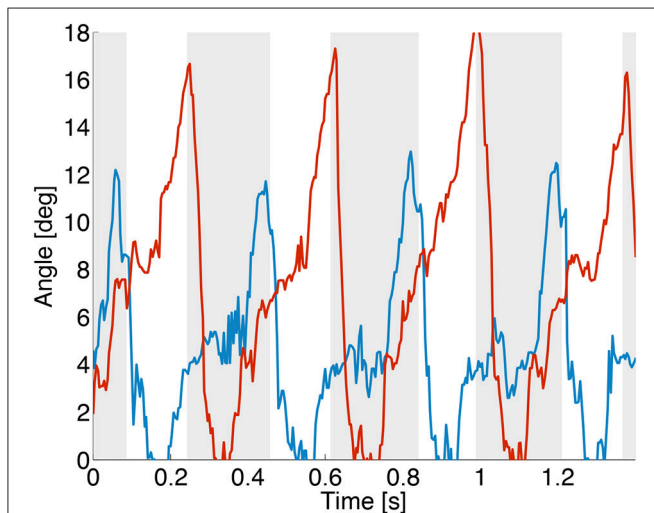
FIGURE 10 | Hardware experiment. The difference between knee and ankle joint position ($p_{knee} - p_{ankle}$) is plotted, for the front left (blue line) and a hind left (red line) leg. Stance phases of the front leg are shown with a gray background, white background indicates the swing phase. Leg loading information is imperfect, and joint friction leads to delayed and damped joint movements. This overestimates loading contact times, but can be filtered. The here shown data is not filtered.

robot's rolling axis. Due to the nature of serial joints, the entire leg structure and its actuators are rotated by the AA joint. This worked satisfactorily, for the shown examples. In case of harsh perturbations or for agile side stepping motions, a more powerful AA joint will become necessary.

Oncilla robot's spring-loaded, pantograph legs performed well for level locomotion indoors and outdoors, slope trotting upwards and downwards, turning during locomotion, and turning on the spot. The leg's intrinsic compliance resolves overdetermined kinematic loops, and all experiments could be performed open-loop. Closed-loop control with Oncilla robot is possible for unstructured terrain (Ajallooeian et al., 2013b; Ajallooeian, 2015). The current design has a limitation which we are planning to resolve in the future: the leg extends only by releasing its charged extensor springs. For maneuvers demanding higher leg power, such as jumping or leaping, an additional actuator, or a redesign of the robot's flexor actuator into a flexor-extensor actuator will become necessary.

One serious problem with RC servo actuated robots is actuator saturation. At combined loads of high speed and torque, these actuators heat up above their thermal dissipation capabilities and required regular cool-down stops as with Cheetah-cub robot (Spröwitz et al., 2013). Hence, Oncilla robot's LL and LA actuators were designed for a low cost of transport, and for sufficient thermal capacities through larger, brushless motors, and optimized gearbox ratios. This lowers the robot's relative power intake, and allows to run it without exceeding hand-warm motor temperatures, at high load and up to 4 Hz cycle frequency. The relatively low COT allows the robot to run tether-free up to 30 min with a 4500 mAh lithium polymer battery. In open-loop mode, good speed can be achieved, i.e., 0.63 m s^{-1} (Fr = 0.25). Optimizing locomotion control parameters, and introducing feedback will likely yield higher robot speed. We expect that continuous progress with off-the-shelf components will further reduce robot weight and increase robot performance: more compact, powerful, brushless motors of different form factors are becoming available (De and Koditschek, 2015), together with compact, high-power, four-quadrant brushless motor controllers (Vedder, 2018).
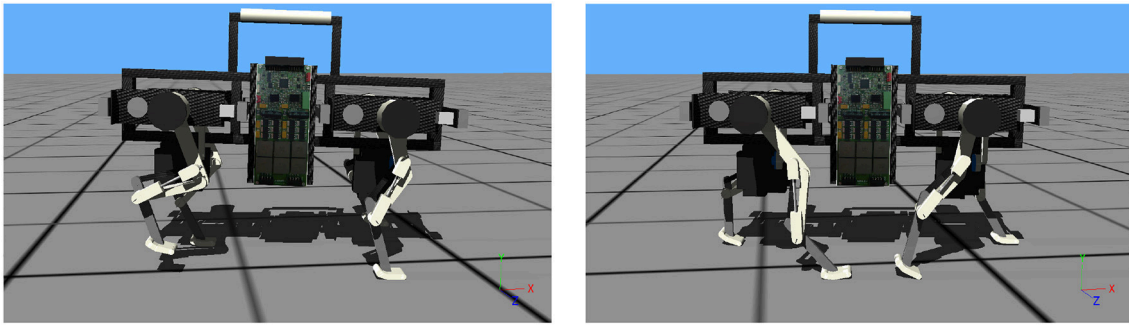
FIGURE 11 | Snapshots show Oncilla robot simulated in Webots, with the robot's heading to the right. The robot trots with an average speed of $0.98\,\text{m s}^{-1}$, at locomotion frequency 3.5 Hz. The commanded duty factor is 0.49, the observed duty factor is 0.52 and 0.58 for front and hind legs, respectively. The commanded step length was 0.12 m, the observed step length 0.14 m. For this run, the robot accelerated from its still standing position, without controlled transition. The corresponding video shows the acceleration as short trunk pitching (link in Table S1).
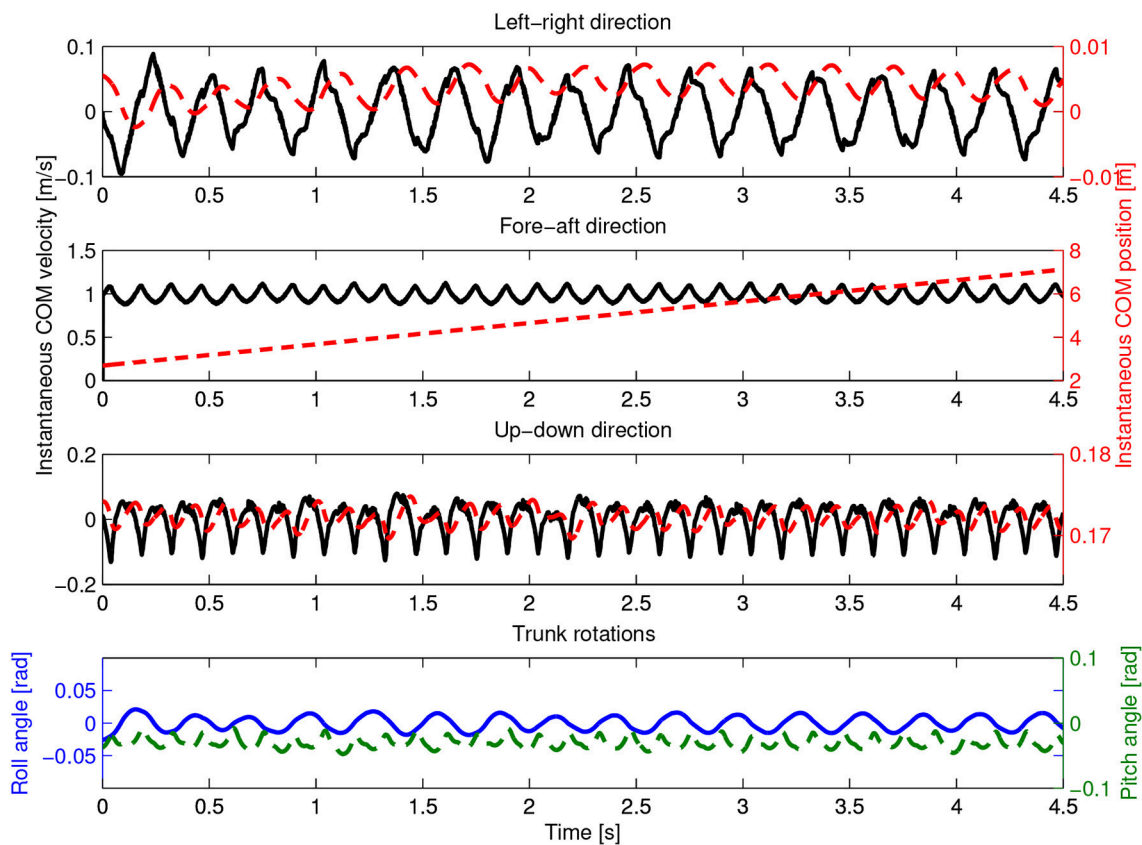
FIGURE 12 | Experiment in Webots. Results are plotted same style as hardware results in **Figure 6**. The simulated robot's center of mass (COM) position and velocity are shown over time, for a cycle frequency of 3.5 Hz. Shown are instantaneous COM position (dashed red) and velocity (full, black line) components sorted by their left-right, fore-aft, and up-down components. The robot reached an average trotting speed of $0.98\,\text{m s}^{-1}$, at peak speeds of $1.12\,\text{m s}^{-1}$. Vertical COM displacement was ±2 mm, at an average hip height of 0.17 m. Average roll angle was ±0.015 rad. The average pitch angle was $-0.03\,\pm0.015$ rad.

Oncilla robot's best measured hardware cost of transport of 3.2 J/(Nm) is 54 % lower than Cheetah-cub robot's COT of 6.9 J/(Nm). Tucker (1970) shows that the metabolic cost of transport of animals decreases with increasing body weight. This heuristically found relationship predicts a COT of 1.0 J/(Nm) for an animal of the weight of Oncilla robot ($m = 5$ kg). The currently most energy efficient, fully active quadruped robot is the 33 kg MIT Cheetah, featuring a COT of 0.5 J/(Nm) at fast, 6 m/s trotting (Hyun et al., 2014). It benefits from custom made brushless actuators, and energy recuperation capabilities.

Oncilla robot trotted slopes with inclination angles up to 10°, faster and with less slippage while going backwards. As nothing else changed, the speed discrepancy must have been caused by leg asymmetry, between forward and backward orientated feet, and pantograph leg joints. More research is required to understand the effects of underactuated, segmented and spring-loaded legs.

By open-sourcing Oncilla robot's mechanical, electrical, software, and simulation blueprints, we aim toward an easily accessible platform for research and education. We hope this project will help spawning improved robots, and allow the field to grow and extend rapidly. All source files are online accessible through permanent repositories, links are provided in **Table 4**. This project was shared between multiple universities in the framework of the EU-FP7 AMARSi project, and five Oncilla robots have been distributed among project partners.

Recently, fluidic and fluidic-hybrid actuation became an interesting choice for articulated robots research (Whitney et al., 2016). Pneumatically actuated quadruped robots like Pneupard (Rosendo et al., 2014) can utilize the actuator's intrinsic compliance, and high frequencies. This keeps robot weight and complexity low. At the same time, large torques, angular amplitudes, and locomotion frequencies even above 7 Hz are feasible (Narioka et al., 2012). This present an

TABLE 4 | Link list Oncilla robot permanent open-source repositories.

Name	License	Format/description/on-line location
Oncilla Hardware Blueprint	GPL-3	SolidWorks 2014, Altium designer, PDF summary Mechanical and Electronics Blueprints of the Oncilla robot https://c4science.ch/source/oncilla-hardware/
sbcp-uc	GPL-3	C sources, MPLAB-X project SBCP protocol slave implementation for dsPIC33FJ DSP processor family https://c4science.ch/source/sbcp-uc/
Oncilla SBCP Master Firmware	GPL-3	C sources, MPLAB-X project SBCP Master Communication board firmware https://c4science.ch/source/oncilla-sbcp-master-fw/
Oncilla Power Management Board Firmware	GPL-3	C sources, MPLAB-X project Power board firmware https://c4science.ch/source/oncilla-power-fw/
Oncilla Motordriver Firmware	GPL-3	C sources, MPLAB-X project Motordriver board firmware https://c4science.ch/source/oncilla-motordriver-fw/
SBCP Low-Level Driver	LGPL-3	C sources, CMake project Low-level SBCP, user-space, device agnostic SBCP drivers, based on libftdi https://c4science.ch/source/sbcpd/
SBCP High-Level Driver	LGPL-3	C++ sources, CMake project Object oriented, high-level, device specific SBCP drivers, Bus management Command Line Utilities https://c4science.ch/source/libsbcp/
liboncilla	LGPL-3	C++ sources, CMake project Level-0 RCI interface, depends on librci (Nordmann et al., 2012) https://c4science.ch/source/liboncilla/
liboncilla Hardware Implementation	LGPL-3	C++ sources, CMake project Level-0 Back-end implementation for the Oncilla hardware https://c4science.ch/source/liboncilla-hw/
liboncilla Webots Interface	LGPL-3	C++ sources, CMake project Level-0 Back-end implementation for the Oncilla Webots based simulator https://c4science.ch/source/liboncilla-webots/
Oncilla Simulation	LGPL-3	Python sources Helper program to setup Oncilla simulation environment https://c4science.ch/source/oncillasim/
CoR-Lab APT repository	n.a.	Apt repository Apt (Debian/Ubuntu) repository for liboncilla dependencies (precise and trusty supported) http://packages.cor-lab.de
Biorob APT repository	n.a.	Apt repository Apt (Debian/Ubuntu) repository for liboncilla and liboncilla-webots (precise and trusty supported) https://ponyo.epfl.ch/packages

Unless specified otherwise (i.e., "apt" or "wget"), users can Git clone repositories. All repositories can be open-source utilized in accordance to the specified license (LGPL-3 or GPL-3).

interesting option for future electric-pneumatic hybrid legged robots.

Soon, legged robots will need to keep up with the claim to perform better than wheeled or tracked vehicles in cluttered environments, and over rough terrain. Boston Dynamics LittleDog robot showed great results as the common robot platform in an earlier DARPA challenge (Righetti et al., 2013). During the tasks of climbing over obstacles, LittleDog robot benefited from its three DOF leg design, and a very large motion joint range. Other legged robots like StarlETH robot (Hutter et al., 2013) or HyQ (Semini et al., 2011) also implemented 3 DOF legs with large joint ranges. Oncilla robot's AA joint allows the robot to turn on the spot, and trot efficiently during turning. For tasks in rougher terrain, or to stabilize the robot from sudden sideways perturbations a faster and stronger AA joint with a larger motion range will become necessary.

We found Oncilla robot's small form factor very helpful. Production and component costs are relatively low, and maintenance is easy. Due to its low weight the robot can safely be handled by a single user, and without gantry installations.

Finally, Oncilla robot provides sensor data, of joint and motor encoders, from gyroscopes, forces sensors, and ground contact sensing based on leg spring deflection. This allows research of closed-loop control (Ajallooeian et al., 2013a,b; Ajallooeian, 2015), and testing bioinspired locomotion controllers. It also makes Oncilla robot a compact and relatively low cost mobile gait analysis tool. Oncilla robot has remaining load capacities for extra sensors like stereoscopic cameras, or distally mounted leg-force sensors. The robot's communication interface and software architecture allows to include such sensors transparently.

6. CONCLUSION

We presented a novel compliant quadruped robot, its software and control framework, and its Webots simulation. We characterize Oncilla robot trotting on level ground, climbing and descending slopes, and during turning. Locomotion speed and direction were set by an open-loop controller in all experiments, without the need to track the robot's posture, internal leg constraints, or considering foot slippage. By outsourcing control tasks into mechanics, Oncilla robot can locomote with low control effort, i.e., open-loop and low control frequencies, over different in- and outdoor substrates. Oncilla robot reached a maximum forward speed of $0.63 \, \mathrm{m\,s^{-1}}$ at a transport of COT = $3.2 \, \mathrm{J/(Nm)}$, at $5 \, \mathrm{kg}$ weight. The robot can turn on the spot, or with a very small turning radius, depending on the applied controller and the utilized robot DOF. The presented example locomotion controller implements gait features in a modular manner, through a central pattern generator. Furthermore, Oncilla robot's open software architecture allows testing other control approaches. For future research with closed loop control, sensors are incorporated and ready for use, i.e., reaction force sensors, gyroscopic sensing, leg-ground detection, and absolute joint position sensors. Oncilla robot's mechanical and electrical blueprints, firmware, simulation, and software architecture are open-source available under GPLv3 (Free Software Foundation, 2007a) for hardware and firmware, and

LGPLv3 (Free Software Foundation, 2007b) for driver, simulation and software architecture. Links are available in **Table 4**. The supplementary document for this manuscript includes a more detailed description of Oncilla robot's electronic hardware components, a table with links to videos, schematic figures and kinematic variables of the robot's leg morphology, and details to the knee and hip motor optimization applied.

LINKS TO ONLINE VIDEOS

1. Oncilla robot trots backwards and forwards https://youtu.be/38pX1FBRlEA

2. Oncilla robot trotting up a 4 slope https://youtu.be/c7wudgzZNkc

3. Oncilla robot turning on the spot, real-time https://youtu.be/TH8AB1mdSoY

4. Oncilla robot in outdoor environment https://youtu.be/A20KLlwuwTg

5. Webots simulation of Oncilla robot trotting forward https://youtu.be/0eAhhNvKjGM

AUTHOR CONTRIBUTIONS

FUNDING

ACKNOWLEDGMENTS

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2018.00067/full#supplementary-material

REFERENCES

Ajallooeian, M. (2015). *Pattern Generation for Rough Terrain Locomotion with Quadrupedal Robots: Morphed Oscillators and Sensory Feedback.* PhD thesis, STI, Lausanne.

Ajallooeian, M., Gay, S., Tuleu, A., Spröwitz, A., and Ijspeert, A. J. (2013a). "Modular control of limit cycle locomotion over unperceived rough terrain," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Tokyo: IEEE), 3390–3397.

Ajallooeian, M., Pouya, S., Spröwitz, A., and Ijspeert, A. J. (2013b). "Central pattern generators augmented with virtual model control for quadruped rough terrain locomotion," in *2013 IEEE International Conference on Robotics and Automation (ICRA)* (Karlsruhe: IEEE), 3321–3328.

Ajallooeian, M., van den Kieboom, J., Mukovskiy, A., Giese, M. A., and Ijspeert, A. J. (2013c). A general family of morphed nonlinear phase oscillators with arbitrary limit cycle shape. *Phys. D* 263, 41–56. doi: 10.1016/j.physd.2013.07.016

Blickhan, R. (1989). The spring-mass model for running and hopping. *J. Biomech.* 22, 1217–1227. doi: 10.1016/0021-9290(89)90224-8

Daley, M. A., Voloshina, A., and Biewener, A. A. (2009). The role of intrinsic muscle mechanics in the neuromuscular control of stable running in the guinea fowl. *J. Physiol.* 587, 2693–2707. doi: 10.1113/jphysiol.2009.171017

De, A., and Koditschek, D. E. (2015). The Penn Jerboa: A Platform for Exploring Parallel Composition of Templates. arXiv preprint arXiv:1502.05347.

Delp, S. L., Anderson, F. C., Arnold, A. S., Loan, P., Habib, A., and John, C. T. (2007). OpenSim: open-source software to create and analyze dynamic simulations of movement'. *IEEE Trans. Biomed. Eng.* 54, 1940–1950. doi: 10.1109/TBME.2007.901024

Falkingham, P. L., and Gatesy, S. M. (2014). The birth of a dinosaur footprint: subsurface 3d motion reconstruction and discrete element simulation reveal track ontogeny. *Proc. Natl. Acad. Sci. U.S.A.* 111, 18279–18284. doi: 10.1073/pnas.1416252111

Free Software Foundation (2007a). *Gnu General Public License.* Available online at: http://www.gnu.org/licenses/gpl.html (Accessed April 01, 2013).

Free Software Foundation (2007b). *Gnu Lesser General Public License.* Available online at: http://www.gnu.org/licenses/lgpl.html (Accessed April 01, 2013).

Gould, S. J., and Lewontin, R. C. (1979). The Spandrels of San Marco and the Panglossian Paradigm: A Critique of the Adaptationist Programme. *Proc. R. Soc. Lond. Ser. B. Biol Sci.* 205, 581–598.

Hutter, M., Gehring, C., Jud, D., Lauber, A., Bellicoso, C. D., Tsounis, V., et al. (2016). "Anymal-a highly mobile and dynamic quadrupedal robot," in *International Conference on Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ* (IEEE), 38–44.

Hutter, M., Remy, C., Hoepflinger, M., and Siegwart, R. (2013). Efficient and versatile locomotion with highly compliant legs. *IEEE/ASME Trans. Mechat.* 18, 449–458. doi: 10.1109/TMECH.2012.2222430

Hyun, D. J., Seok, S., Lee, J., and Kim, S. (2014). High speed trot-running: Implementation of a hierarchical controller using proprioceptive impedance control on the MIT Cheetah. *Int. J. Robot. Res.* 33, 1417–1445. doi: 10.1177/0278364914532150

Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: a review. *Neural Netw.* 21, 642–653. doi: 10.1016/j.neunet.2008.03.014

Ijspeert, A. J., Crespi, A., Ryczko, D., and Cabelguen, J.-M. (2007). From swimming to walking with a salamander robot driven by a spinal cord model. *Science* 315, 1416–1420. doi: 10.1126/science.1138353

Kennedy, J., and Eberhart, R. (1995). "Particle swarm optimization," in *Proceedings of IEEE international Conference on Neural Networks* (Perth, WA), Vol. 4, 1942–1948.

Khoramshahi, M., Spröwitz, A., Tuleu, A., Ahmadabadi, M. N., and Ijspeert, A. (2013). "Benefits of an active spine supported bounding locomotion with a small compliant quadruped robot," in *Proceedings of 2013 IEEE International Conference on Robotics and Automation* (Karlsruhe), 3329–3334.

Li, C., Umbanhowar, P. B., Komsuoglu, H., Koditschek, D. E., and Goldman, D. I. (2009). Sensitive dependence of the motion of a legged robot on granular media. *Proc. Natl. Acad. Sci. U.S.A.* 106, 3029–3034. doi: 10.1073/pnas.0809095106

Maufroy, C., Kimura, H., and Takase, K. (2010). Integration of posture and rhythmic motion controls in quadrupedal dynamic walking using phase modulations based on leg loading/unloading. *Auton. Robots* 28, 331–353. doi: 10.1007/s10514-009-9172-5

Michel, O. (2004). Cyberbotics Ltd. Webots^{TM}: professional mobile robot simulation. *J. Adv. Robot. Sys.* 1, 39–42. doi: 10.5772/5618

Narioka, K., Rosendo, A., Spröwitz, A., and Hosoda, K. (2012). "Development of a minimalistic pneumatic quadruped robot for fast locomotion," in *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (Guangzhou: IEEE), 307–311.

Naturalpoint, Inc. (2011). *Optitrack s250e.* Available online at: http://www.optitrack.com/

Nordmann, A., Rolf, M., and Wrede, S. (2012). "Software abstractions for simulation and control of a continuum robot," in *Simulation, Modeling, and Programming for Autonomous Robots* (Tsukuba: Springer), 113–124.

Nordmann, A., Tuleu, A., and Wrede, S. (2013). "A domain-specific language and simulation architecture for the oncilla robot," in *ICRA 2013 Workshop on Developments of Simulation Tools for Robotics and Biomechanics*, 3.

Righetti, L., Buchli, J., Mistry, M., Kalakrishnan, M., and Schaal, S. (2013). Optimal distribution of contact forces with inverse-dynamics control. *Int. J. Robot. Res.* 32, 280–298. doi: 10.1177/0278364912469821

Righetti, L., and Ijspeert, A. J. (2008). Pattern generators with sensory feedback for the control of quadruped locomotion. in *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA 2008)* (Pasadena, CA), 819–824.

Robotis, Ltd., Co. (2018). *Communication Overview Protocol 1.0.* Available online at: http://emanual.robotis.com/docs/en/dxl/protocol1/

Roos, F., Johansson, H., and Wikander, J. (2006). Optimal selection of motor and gearhead in mechatronic applications. *Mechatronics* 16, 63–72. doi: 10.1016/j.mechatronics.2005.08.001

Rosendo, A., Nakatsu, S., Liu, X., Shimizu, M., and Hosoda, K. (2014). "Quadrupedal locomotion based on a muscular activation pattern with stretch-reflex," in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (Bali), 773–778.

Schmitt, S., and Günther, M. (2011). Human leg impact: energy dissipation of wobbling masses. *Arch. Appl. Mech.* 81, 887–897. doi: 10.1007/s00419-010-0458-z

Semini, C., Tsagarakis, N. G., Guglielmino, E., Focchi, M., Cannella, F., and Caldwell, D. G. (2011). Design of hyq - a hydraulically and electrically actuated quadruped robot. *IMechE, I J. Sys. Control Eng.* 225, 831–849. doi: 10.1177/0959651811402275

Seok, S., Wang, A., Chuah, M. Y., Otten, D., Lang, J., and Kim, S. (2013). "Design principles for highly efficient quadrupeds and implementation on the MIT cheetah robot," in *2013 IEEE International Conference on Robotics and Automation*, 3307–3312.

Söhnela, K., Andradaa, E., de Lussanetb, M. H., Wagnerb, H., and Fischera, M. S. (2017). "Kinetics of jumping regarding agility dogs" in *Engineering for a Changing World: Proceedings; 59th IWK, Ilmenau Scientific Colloquium, Technische UniversitÄt Ilmenau, September 11-15, 2017* (Ilmenau), Vol. 59 of 3.2.

Spröwitz, A., Tuleu, A., Vespignani, M., Ajallooeian, M., Badri, E., and Ijspeert, A. J. (2013). Towards dynamic trot gait locomotion–design, control, and experiments with cheetah-cub, a compliant quadruped robot. *Int. J. Robot. Res.* 32, 932–950. doi: 10.1177/0278364913489205

Tucker, V. A. (1970). Energetic cost of locomotion in animals. *Comp. Biochem. Physiol.* 34, 841–846. doi: 10.1016/0010-406X(70)91006-6

Vedder, B. (2018). *Benjamin's Robotics.* Available online at: http://vedder.se/2015/01/vesc-open-source-esc/

Walker, S. M., Schwyn, D. A., Mokso, R., Wicklein, M., Müller, T., Doube, M., et al. (2014). *In vivo* time- esolved microtomography reveals the mechanics of the blowfly flight motor. *PLoS Biol.* 12:e1001823. doi: 10.1371/journal.pbio.1001823

Whitney, J. P., Chen, T., Mars, J., and Hodgins, J. K. (2016). "A hybrid hydrostatic transmission and human-safe haptic telepresence robot," in *2016 IEEE International Conference on Robotics and Automation (ICRA)* (Stockholm: IEEE), 690–695.

Wienke, J., and Wrede, S. (2011). "A middleware for collaborative research in experimental robotics," in *International Symposium on System Integration* (Kyoto), 1183–1190.

Witte, H., Hackert, R., Ilg, W., Biltzinger, J., Schillinger, N., Biedermann, F., et al. (2003). "Quadrupedal mammals as paragons for walking machines. in *Proc AMAM - Adaptive Motion in Animals an Machines* (Kyoto).

Zeglin, G. (1999). *The Bow Leg Hopping Robot.* Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.