Towards Jumping Locomotion for Quadruped Robots on the Moon

Hendrik Kolvenbach, Elias Hampp, Patrick Barton, Radek Zenkl and Marco Hutter

Abstract— Jumping locomotion has the potential to enable legged robots to overcome obstacles and travel efficiently on low-gravity celestial bodies. We present how the 22 kg quadruped robot SpaceBok exploits lunar gravity conditions to perform energy-efficient jumps. The robot achieves repetitive, vertical jumps of more than 0.9 m and powerful single leaps of up to 1.3 m. We present the implementation of a reaction wheel, which allows for control of the robots pitch orientation during the flight phase. We also demonstrate the implementation of a parallel elasticity in the legs providing the capability of temporarily storing and reusing energy during jumping. The jumping and attitude controller are subsequently presented. Finally, we analyze the energetics of the system and show that jumping with the integrated elasticity significantly reduces energy consumption compared to non-elastic jumps.

I. INTRODUCTION

The combination of walking and jumping is a powerful form of locomotion in the animal kingdom. Small animals such as fleas, locusts, and specific bugs make frequent use of jumps in order to traverse distances and heights multiple times their body size [1]. This allows the animal to overcome enormous obstacles, to move efficiently in their respective environment, and to escape predators [2].

Most tiny jumping animals, specifically insects, slowly charge elastic elements and rapidly release the kinetic energy to perform a single, powerful jump [3]. However, this ‘pause and leap’ motion tends to lead to crash-landings at the target location. Because of the square-cube law, such impacts are not necessarily harmful to the animal and can be absorbed by, for example, a rigid exoskeleton.

A different jumping approach is chosen by larger animals, such as Springboks, Kangaroos, and Galagos. Due to gravitational load, the jump height and distance decrease rapidly with size, weight, and consequently available muscle mass of the animal. Additionally, jumping of large animals tends to be more precise, since uncontrolled landing might cause injury, and continuous, to increase the efficiency of the locomotion [4] [5]. Some periodic gaits have evolved from such motions which are commonly known as pronking or bounding.

This form of locomotion becomes especially interesting when developing robots for lower-than-earth gravity which are not bound to terrestrial scaling laws. One extreme example is the unwillingly performed jump by the Rosetta lander Philae, which allowed the ~100 kg probe to traverse one kilometer on the surface of comet 67P/Churyumov-Gerasimenko while remaining functional [6]. Naturally, several hoppers have been investigated for traversing asteroids and small moons such as the Phobos Hopper, JAXA’s Minerva, DLR’s MASCOT and other prototypes [7] [3]. Their locomotion principle follows that of small animals, where kinetic energy is accumulated and rapidly released to perform a single jump. Similar to insects, the structure absorbs the landing energy and little emphasis is put on precise attitude control.

With a sixth of earth’s gravity, the Moon is an appealing target for combined jumping and walking robots. A periodic gait with extended flight phases can be used to traverse long distances fast and in an energy-efficient manner, high jumps can be used to overcome obstacles, and walking gaits can be used where precise positioning is required [8]. In this context, the Apollo astronauts have famously demonstrated how to exploit lunar gravity and naturally adapted to an efficient skipping gait with extended flight phases [9]. Several capable lightweight legged robots have been developed which mimic the capability of small jumping animals on earth: A 7 g jumper with parallel elasticity was designed at the EPFL [10], UC Berkeley’s Salto showed impressive performance by exploiting a series elastic actuation principle [11], and UPenn developed the direct drive actuated legged robots Minitaur and the Jeorba hopper [12]. While the small robots demonstrate the capability of reaching high jumping heights...
compared to their size, only a few of these systems show repetitive high-jump ability. The updated version of Salto, Salto-1p makes use of a reaction wheel and fans to stabilize and steer the robot [13]. The use of a tail was explored with the 2.5 kg Jeorba hopper [14] and the 1.25 kg Cheetah-Cub [15]. A reaction wheel was proposed during a simulation study to stabilize the pronking motion of a 10 kg quadruped on compliant terrain [16]. However, these systems are not capable of carrying a significant payload, which makes them of limited use in the proposed application.

In previous work, we have introduced SpaceBok*, a quadruped robot optimized for jumping locomotion [17]. The robot was built to investigate how walking robots can be used in space exploration to explore scientifically interesting targets in areas which are difficult to reach for wheeled platforms [18]. However, SpaceBok was not able to stabilize the flight phase during high jumps, and jumping was generally not very efficient due to missing elasticities. This became especially apparent when testing in a lower-than-earth gravity where jumping heights easily exceeded the body height.

In this work, we show how the stable, efficient, and repetitive (high) jumps of the 22 kg SpaceBok robot are achieved †. The robot is significantly bigger, heavier, and with four legs, more complex than previous jumping robots and thus imposes unique challenges on the design and control. Our empirical contribution consists of the iteration of SpaceBok, which involves the design of the parallel elasticity, the implementation of a reaction wheel, and the respective control structure. We tested the system on a specially designed testbed, which allows for the simulation of a lunar gravity scenario. Our theoretical contribution is the evaluation of the energetics of the system, comparing the energy consumption for locomotion with and without elastic elements.

We describe the design of the parallel elasticity, reaction wheel, as well as the control structure and the test setup in Sec. II. Subsequently, we present the test results of repetitive jumping, attitude control, and energetics in Sec. III. Finally, we conclude the work in Sec. IV.

II. Method

A. SpaceBok robot

Compared to other quadruped robots, SpaceBok was primarily built to investigate jumping locomotion. We focused on lightweight design, a kinematic chain for the leg that allows the actuators to work together during jumping, and a safe way of integrating tension springs into the structure. Each leg has two Degrees of Freedom (DOF), which allow for hip flexion/extension and knee flexion/extension. Hip abduction/deduction was omitted to save weight and decrease complexity. The hip height of the robot is 500 mm and the platform weighs 22 kg. The system is equipped with an Intel i7 computer (FitPC IPC3) that executes the control software. An Inertial Measurement System (IMU) (Vectornav VN100) provides pose estimates and a lithium-polymer battery (Swaytronic 12S 6000 mAh) with an in-house developed Battery Management System (BMS) powers the robot. The BMS allows monitoring of energy consumption, and charging of the battery in case energy is recuperated. A 20 mF capacitor is placed between the battery and the consumers to smooth the readings. A detailed overview of the platform can be found in [17]. In the following, the upgrades to enable energy-efficient, repetitive jumping are described.

B. Parallel elasticity

The leg kinematic of SpaceBok is described by a four-bar parallel motion linkage, as seen in Figure 2. The link lengths are set to $l_1 = 120$ mm and $l_2 = 250$ mm and each leg is actuated by two brushless DC motors (T-Motor U8 KV85) in combination with a single-stage planetary gearbox with a transmission ratio of 9.55:1. The hand-laminated carbon feet are designed to absorb peak loads during landing and are inspired by prosthetic legs for humans and the RHex robot [19]. A rubber pad is attached to the sole to increase traction on the ground. For this work, we integrated two modified, commercially available tension springs (Gutekunst Z-156K1) in the legs (Figure 2) to recuperate energy during jumping and to exploit the mechanical advantage for powerful single jumps. The springs are placed in parallel to the actuators and are tensioned if the leg is compressed from a default position. The leg kinematic leads to a maximal deflection of each spring by $b = 100$ mm. The undeformed length of the spring is constrained to $a = 120$ mm to ensure that its full length fits into the carbon tubes if extended. The spring constant of the modified spring is $k = 2$ N/mm and it weighs 0.09 kg. Figure 3 shows the characterization of the spring in terms of the exerted spring force vs stroke and the rotational stiffness as a function of joint torque vs joint deflection.

---

*https://youtu.be/oujhXCPSA-k
†https://youtu.be/HTEIcjHlAes
Fig. 3: Diagrams of the spring characterization showing the force $f$ exerted on the environment depending on stroke $s$ and the rotational stiffness defined as the torque of the spring $\tau_s$ acting on the joints in relation to the deflection angle $\phi$ from the untensioned configuration.

The springs can store up to 18 J per leg if fully compressed. When performing a passive jump from crouched position with fully compressed legs and disabled drive actuators, the robot achieves a jump height of 0.6 m at lunar gravity.

A Dyneema cord is attached to the springs, guided by pulleys and connected to a decoupling element, to allow for manual adjustment of the length differences after assembly. An adjustable nylon zip tie fastens the connection, which simplifies handling effort. The lower spring is rigidly connected to the leg.

A miniature one-axis load cell (YGX YGX601L-13) connects the upper spring to the structure of the leg. The additional weight for including the parallel elasticity (springs, force sensors and spring adapters) sums up to 0.86 kg and accounts for roughly 4% of the total mass of the robot. The analog signals are read by a 24-Bit Analog-Digital Converter (SparkFun HX711) placed in close proximity to the leg inside the main body. The force measurements are published to the FitPC via a micro-controller (Arduino Mega 2560) at 80 Hz.

C. Reaction wheel

The goal of the flight stabilization system is to control the attitude of the robot in the flight phase to (re-)orient the robot for a safe landing. There exist many suitable concepts for this task: An actively controlled boom which acts as a tail, the use of swinging legs, or a reaction wheel. A tail is a good solution for one jump, but due to its limited operational range, it is not very well suited for multiple jumps [20]. The low inertia of SpaceBok’s legs and their kinematic constraints makes the use of swinging legs difficult for the achievable jump height. Hence, we chose to use a reaction wheel as a flight stabilization system. It allows for high maneuverability during flight and can be easily integrated into the robot’s main body.

The inertia and maximum angular velocity of the reaction wheel define the angular momentum available to correct the orientation of the robot. Thus, the inertia and maximum angular velocity should be maximized while optimizing the reaction wheels weight and volume. We defined the following boundary conditions based on ROS Gazebo simulations with hand-tuned gains to design the reaction wheel:

- The maximum initial angular velocity of the robot after take-off is $70^\circ/s$
- Orientation errors between $20^\circ$ and $-20^\circ$ should be compensated

With these requirements and an estimated maximum flight time of 3 s in lunar gravity, the necessary torque acting on SpaceBok which ensures adequate control is 1.6 Nm. The angular acceleration of the robot depends solely on the torque exerted by the motor on the reaction wheel. The reaction wheel is driven by a brushless DC motor similar to those found in the legs (T-Motor Antigravity 7005 KV115), which is integrated with a motor controller (ELMO Twitter Gold Solo) and a small on-axis encoder (RLS RM08) to allow for direct torque control. The 0.536 kg reaction wheel is manufactured out of brass and designed for high inertia. It has an outer radius of 75 mm and a width of 14.4 mm which yields a combined inertia of 0.003 kgm$^2$ (Reaction wheel: 0.0026 kgm$^2$, Rotor inertia: 0.0004 kgm$^2$). The assembly (including 0.1 kg for the motor controller) weighs 1.2 kg and can be seen in Figure 4. The reaction wheel is mounted to the side of the robot and can be adjusted in the plane to align with the principal axis of (pitch-) rotation.

D. Control structure

SpaceBok’s control software makes use of the Robot Operating System (ROS), which serves as a middleware. An overview of the key modules can be seen in Figure 5. The state estimator is acquiring IMU data, spring forces, absolute orientation, and joint information. During testing (Sec. II-G), the robot was lying sideways on an inclined rail, and the orientation estimates were therefore supplied by the Vicon motion capture system.
Fig. 5: Dataflow visualization in SpaceBok during the low-gravity jumping tests. The green boxes show software modules that are executed on the on-board computer while yellow boxes show peripheral components.

Fig. 6: Definition of jump height $h$, compression depth $d$ and total height $h$.

During normal operation, the orientation estimates are supplied by the IMU. The highlevel controller processes the state and computes desired joint torques and velocities (Sec. II-E and Sec. II-F) based on user commands from an operator PC or a gamepad. Lastly, the motor commands are forwarded to the motor controllers via the lowlevel controller, which takes care of the EtherCAT communication.

E. Jumping controller

We chose the periodic pronking gait and a robust but straightforward control strategy to demonstrate the continuous jumping ability of SpaceBok. During pronking, the actuators are used in conjunction and allow for the highest jumps. For the following, we use the definitions as visualized in Figure 6. If we take the height of the center of mass of the body when standing with fully extended legs as a baseline, then the jump height is the distance to the highest point reached. The compression depth is used to refer to the maximum distance of the body below that baseline. The total height is the compression depth plus the jump height. The pronking controller is based on a state machine with four different phases as shown in Figure 7. We implemented a virtual model controller (VMC) [21] to control the robot during stance and a position controller to control the feet positions during flight. For the first jump, the robot compresses to a crouch position determined by the desired jump height. When commanded by the user, the system switches from the crouch position to the thrust phase. During this phase, a defined thrust is applied by the motors until the legs are fully extended. Once the legs are extended, the jump height is estimated based on the take-off velocity of the robot:

$$h_{\text{est}} = \max(\tau z, k) \cdot \left( g - \mu \right)$$

The take-off velocity is calculated from the leg odometry and the robot’s orientation and is defined as the maximum foot velocity in the vertical direction at the end of the thrust phase. A small friction term has been added to compensate for rail friction $\mu < 0.1$.

The detection of leg extension prompts also the transition to the flight phase. Here, the robot positions its feet such that they are extended and the springs are not compressed. Ground contact is detected once a torque threshold is reached on the drive and the controller switches back to the compression phase. During the compression phase, SpaceBok uses its odometry estimates of torso height and velocity to calculate the force required such that its vertical velocity will be zero at the desired compression depth, mimicking a virtual spring-damper system.

The desired compression depth and the applied thrust controls the jump height during the stance phase. The jump height is defined by:

$$j = h - d$$

where $j$ = Jump height

$h$ = Total height

$d$ = Compression depth

The total height is determined through the conservation of energy. The input energy is the work done by the motors plus the energy stored in the springs. The total height is then given by:

$$h = \frac{F \cdot d + E_s}{m \cdot g}$$

where $m$ = Total mass of robot

$g$ = Gravity

$F$ = Force applied in thrust phase

$E_s$ = Energy stored in springs
The thrust and compression depth are linearly related to the total height and can both be chosen freely. This allows us to use the springs with maximum efficiency since SpaceBok does not need to compress more than necessary for small jumps. Thus, we implemented a proportional controller to adjust the thrust gain based on the difference between the desired and estimated jump height.

Besides the virtual force, a virtual torque is calculated to correct orientation errors of the robot during stance. The calculated virtual wrench is mapped to foot forces using constrained quadratic optimization similar to [22]. The formulation of the problem with \( k \) feet in ground contact is:

\[
\begin{align*}
\min_{x} & \quad \| (Ax - b) \|_2^2 \\
\text{s. t.} & \quad F_k^n \geq F_{\text{min}} \quad (4) \\
& \quad -\mu F_k^n \leq F_k^s \leq \mu F_k^n \quad (5) \\
& \quad -\tau_{\text{max}} \leq J^T \dot{x} \leq \tau_{\text{max}}, \quad (6)
\end{align*}
\]

where

\[
\begin{align*}
A = \begin{bmatrix} I & I & \cdots & I \\
\hat{r}_{\text{BF},1} & \hat{r}_{\text{BF},2} & \cdots & \hat{r}_{\text{BF},k} \end{bmatrix} : \text{Transformation matrix} \\
x = \begin{bmatrix} \lambda_1 \\
\lambda_2 \\
\vdots \\
\lambda_k \end{bmatrix} \in \mathbb{R}^{2k} : \text{2D contact forces of the feet} \\
b = \begin{bmatrix} F^n \\
T_v \end{bmatrix} \in \mathbb{R}^5 : \text{Virtual wrench acting at the COM} \\
F^n: \text{Normal contact force} \\
F^i: \text{Tangential contact force} \\
\tau: \text{Stacked motor torques} \\
J: \text{Stacked actuator Jacobian} \\
I: \text{Identity matrix} \\
\hat{r}_{\text{BF},k}: \text{Position vector from robot base (COM) to foot} k
\end{align*}
\]

The desired joint torques \( \tau_{VMC}^* \) are then acquired by using the Jacobian-transpose mapping of the optimized foot forces:

\[
\tau_{VMC}^* = J^T x \quad (8)
\]

A challenge in multilegged jumping lies in the precise force-control of all feet. Compared to single-leg hoppers, small errors between desired and actual joint torque can create a significant moment around the COM during the stance phase. To avoid saturation of the reaction wheel at an early stage and enable higher jumps, these unwanted moments around the COM should be minimized.

Since SpaceBok’s motors are directly torque-controlled, we found it necessary to model and compensate for joint friction \( \tau_f \). Thus, we implemented a friction model containing Coulomb friction \( \tau_c \) and viscous friction \( \tau_v \) on the motor velocity \( \dot{\varphi}_i \) for each leg actuator \( i \) [23], as follows:

\[
\tau_{fr,i} = \tau_{c,i} + \tau_{v,i} = c_{0,i} \cdot \text{sgn}(\dot{\varphi}_i) + c_{1,i} \cdot \dot{\varphi}_i \quad (9)
\]

which yields the friction compensated motor torques \( \tau^* \):

\[
\tau^* = \tau_{VMC}^* + \tau_{fr} \quad (10)
\]

During the compression phase, the torques resulting from the springs \( \tau_f \), have to be subtracted from the commanded motor torque following:

\[
\tau_{\text{compression}}^* = \tau_{VMC}^* - \tau_s + \tau_f \quad (11)
\]

where the spring compensation torques \( \tau_s \) act equally on the two actuators of each leg and are calculated, with the definitions of Figure 2 & 3:

\[
\tau_{s,k} = F_{s,k} \cdot \sin \beta_k \cdot l_2 \quad (12)
\]

Finally, the desired motor commands are sent to the motor controllers via the low-level controller at 600 Hz.

**F. Attitude controller**

The flight stabilization via reaction wheel is realized using a PD controller:

\[
\begin{align*}
\tau_{\text{rw,fi}}^* &= k_{p,fi} \cdot (p^* \otimes p) - k_{d,fi} \cdot \dot{\theta} \quad (13) \\
\tau_{\text{rw,si}}^* &= -k_{p,si} \cdot \dot{\varphi}_{\text{rw}} + k_{d,si} \cdot \ddot{\varphi}_{\text{rw}} \quad (14)
\end{align*}
\]

During the flight phase, the desired reaction wheel torque \( \tau_{\text{rw,fi}}^* \) is calculated based on the robot’s orientation error in unit quaternions and angular rate. During stance, the braking torque \( \tau_{\text{rw,si}}^* \) is commanded to unload the reaction wheel. We hand-tuned the gains during our experiments for fast response while minimizing potential overshoot. To counteract the effect of the braking torque on the robot, \( \tau_{\text{rw,si}}^* \) is added to the pitch torque \( T_v \) of the wrench \( b \) (Subsec. II-E) as a feed forward term during the stance phase.

**G. Test setup**

We performed the validation of the system at the ORBIT testbed located at the Automation and Robotics laboratories at the European Space Research and Technology Centre (ESTEC) [24]. We designed an inclined rail on which we mounted the robot sideways in order to offload the gravitational force of the earth (Figure 8). The rail is inclined to 9.51° which mimics the gravitational acceleration of the Moon \((g_m = 1.62 \, \text{m}/\text{s}^2)\). The robot is attached to a sled via a thrust ball bearing, where the axis of rotation aligns with the pitch axis. The sled is connected to the rail and allows for unconstrained translation with low-friction in the z-direction. The rail is placed within the 45 m² testbed and in good view of 14 Vicon cameras, which provide the ground truth for the experiment.

**III. RESULTS AND DISCUSSION**

**A. Repetitive jumping**

Figure 9 shows the position of the COM of the robot during the jumping experiments. We set the desired \( z \) position from 0.6 m to 1.4 m with increments of 0.2 m. Stable, repetitive jumping is achievable with and without springs. The height estimation method based on the feet velocity at take-off reveals to be a reliable method to predict the jump height and to control it. When jumping with the parallel elasticity, the compression depth is reduced compared to jumping without springs.
This is due to the relatively high stiffness of the springs and the design of the controller, which forces the system to use the springs efficiently and to avoid additional energy towards compression.

Figure 10 shows the ability of the reaction wheel to compensate for orientation errors during the flight phase. Initial angular velocities and orientation errors are quickly compensated. High torques are visible during the stance phase, which brakes the reaction wheel.

### B. High jumps

Figure 1 illustrates the capability of SpaceBok to perform powerful, single jumps with springs. The robot’s COM reaches a height of up to 1.8 m which is more than three times its body size. To achieve these heights, however, the motors are in close range of saturation and eventually trigger the current limit of the motor controllers. Because of the high take-off velocity, small disturbances in the control of the actuators lead to a significant initial angular velocity of the robot, which is challenging to compensate with the reaction wheel. During this jump, for example, the robot reaches an orientation error of almost 15° but manages to return to zero before landing.

The maximum jump height is a sole function of the take-off velocity and thus of the maximum motor speed. The springs contribute to the jumps by reducing the energy required by the motor to achieve a certain speed.

### C. Energetics

We measure the total energy consumption between battery and capacitor by using the BMS. The average standby power consumption of 85 W of the robot was deducted. In standby, the actuators are not powered, and the power consumption can be attributed to the idle power consumption of the logic stage of the motor controllers, the on-board computer, other peripherals such as fans and the inefficiency of the DC-DC converters.

The parallel elasticities contribute significantly to the energy balance and reduces the energy consumption per jump by almost a factor of two regardless of jump height (Figure 11a). This matches similar findings which have been obtained with highly optimized series elastic springs in previous work [25]. It is important to note that the weight of the springs contributed to the overall mass of the robot during all experiments.
Fig. 11: Fitted curve of the average electric energy consumption (a) during repetitive jumping depending on jumping height, measured at the battery. Investigating the electric power consumption per jump with (b) and without (c) springs shows a generally lower power consumption when using springs. The peak power consumption (maximum and minimum indicated by the red and orange lines) is smaller when using elasticities.

Since the mass for the spring assembly is less than 4% of the total mass of the robot, we consider the difference in energy to be negligible.

We analyzed the electric power consumption at different jump heights which revealed that the use of the parallel elasticity lowered the average power consumption and the peak powers (Figure 11b and 11c). The motor torques and velocities during the compression phase show the contribution of the parallel elasticity, to which energy is being stored (Figure 12). Using the springs results in a shorter stance phase and higher torque, yet peak velocity remains the same to achieve the desired jump height.

We also measured the amount of antagonistic energy absorbed by the actuators during jumping, which tends to be significant in parallel mechanisms. Antagonistic energy is defined as the mechanical energy of actuators working against each other. For example, if one actuator moves and the other one brakes, more energy is required than should be necessary for a specific task. This behaviour can be seen during re-positioning of the legs after the thrust phase in Figure 12. We used the definitions for total ($P_{\text{tot}}$), net ($P_{\text{net}}$) and antagonistic ($P_{\text{ant}}$) power of [26],

$$P_{\text{ant}}(P) = \frac{1}{2}(\sum |P_t| - \sum P_t) = \frac{1}{2}(P_{\text{tot}} - |P_{\text{net}}|),$$

(15)

where $P(t)$ is the element-wise product of torque and speed of the leg actuators, yielding the power tuple $P(t) = (τ_1 \dot{ϕ}_1, τ_2 \dot{ϕ}_2, ..., τ_n \dot{ϕ}_n)$. The mechanical energy is then calculated as the integrals of the power terms over time [27]. We averaged the energy consumption over the jump heights from $\sim 0.2 - 0.9$ m to obtain the mechanical energy of a single, average jump. The data showed a relatively high amount of antagonistic energy compared to the net energy required for the task (Figure 13).

Fig. 13: The total, net and antagonistic mechanical energy consumption for an average jump height of $0.54 \pm 0.01$ m.

IV. CONCLUSION

In this paper, we showed how repetitive jumping locomotion is realized on a 22 kg quadruped robot at lunar gravity. We introduce a parallel elastic mechanism in the legs to recuperate energy during jumping, which increases the efficiency when compared to non-elastic jumps.

The implementation of a reaction wheel allows for effective control of the pitch orientation of the robot and stabilizes and corrects orientation errors during the flight phase.
In order to carefully control the torques commanded to the legs, we implemented force sensors to measure spring forces and added compensation terms to account for joint friction. Repetitive jumping is achieved by the implementation of a virtual model controller, through which we can adjust compression depth and thrust gain to control the jump height, as well as an in-flight orientation control for the reaction wheel. Taking into account the take-off velocity from odometry estimates is sufficient to predict the jump height of the robot with high accuracy. The exploitation of the parallel elasticity has significantly reduced the energy consumption and average power demand for locomotion.

While the concept generally works well, further improvements can be made. First, the parallel linkage mechanism absorbs energy due to antagonism, which could be reduced with a more specialized mechanism. Second, to precisely control the forces at the feet, sensor feedback on joint level would improve performance. Additionally, increasing the bandwidth of the spring force measurement would potentially increase the fidelity of the controller. It remains to be evaluated if other dynamic gaits with full flight phases, such as a running trot or gallop could achieve similar or better energetic performances at lunar gravity.

ACKNOWLEDGMENT

The authors would like to thank Aurora Aerospace and the Suisse Space Centre for funding the internship placement of Elias Hampp, Patrick Barton and Radek Zenkl at the European Space Agency. We would also like to thank Gianfranco Visentin and Martin Werner Zwick from the Automation and Robotics section of the European Space Agency for their organizational efforts in realizing the tests.

REFERENCES