

RESEARCH ARTICLE

Open Access



# BBot, a hopping two-wheeled robot with active airborne control

Huei Ee Yap\* and Shuji Hashimoto

## Abstract

Most two-wheeled robots have algorithms that control balance by assuming constant contact with the ground. However, such algorithms cannot confer stability in robots deployed on non-continuous ground terrain. Here, we introduce BBot, a robot that can hop as well as move over stepped terrains. BBot has a two-wheeled lower body platform and a spring-loaded movable upper body mass. Hopping results from the impact force produced by release of pre-tensed springs. An inertia measurement unit detects the angle of body tilt, and an ultrasonic distance sensor records the height above ground. An accelerometer in the inertia measurement unit measures the impact force to determine the beginning and end of the phases of hopping and landing. Torque generated from rotation of the drive wheels controls the airborne robot's body angle. Sensors detect the impact of landing, and controls immediately switch to ground balance mode to stay upright. Experiment results show that BBot is capable of traversing down a 17 cm step, enduring manual toss landing and hopping 4 cm above ground.

**Keywords:** Two-wheeled robot, Step traverse, Attitude control, Inverted pendulum, Hopping

## Background

Mobile robots often need to navigate non-continuous terrains with obstacles. Crawler-type robots can move over spacious uneven terrains but are usually heavy and slow. Biped robots that can navigate through natural environments are currently expensive to make and deploy. Lunar rovers such as the NASA Soujourner and the Ecole Polytechnique Federale de Lausanne (EPFL) Shrimp robot combine adaptive legs with the efficiency of wheels to traverse stepped terrains. This approach relies on redundancy of static points of support on the robot's base to stabilize climbing and descending steps. These robots, albeit efficient, have complex structural designs and driving systems. Currently, there are many robots capable of jumping. Examples include 7g [1], Grillo [2], MSU [3] and mowgli [4]. These robots use linkage leg systems and springs to leap over large obstacles. However, the horizontal movements of such electro-mechanical machines are inefficient and inaccurate. Mini-Whigs [5], Jollbot [6], Scout [7] integrate mobility and jumping to cover

large areas and negotiate stair-like obstacles. But, jumping is passive, and there is no control of airborne attitude.

In 2008, Kikuchi et al. [8] introduced a wheeled-based robot that climbs up and down stairs dynamically. A spring-loaded movable upper body mass allows their robot to land softly and double-hop in midair. Kikuchi robot consists of a statically stable wheel base. One of the drawbacks of this is that the robot has to land with minimum body tilt angle to ensure a successful landing. Safe landing is not guaranteed if there exists any external disturbances during airborne. Another jumping robot named iHop[9] is a transformable two wheeled robot. In hoping mode, it uses both wheels as weights and has a lockable hopping mechanism. iHop pushes its wheels upward while balancing on the central chassis. iHop exhibits hoping capability but it is not shown that the robot is capable of climbing up or down step terrains.

The tradeoff between mobility and system complexity is the main challenge faced by many mobile robots. For example, Boston Dynamics' BigDog [10] moves easily over rough terrains, but complexity and construction costs limit the wide adoption of such machines. By contrast, simpler designs such as the 7g have mobilities that are too restricted for practical use. Unlike existing robots,

\*Correspondence: huei@shalab.phys.waseda.ac.jp  
Department of Applied Physics, Graduate School of Advanced Science and Engineering, Waseda University, Tokyo, Japan

humans and animals negotiate stepped terrains dynamically and efficiently. For example, when jumping up or down a step, we tend to bend our knees to absorb the impact during landing and use the momentum to push ourselves forward and continue our next step.

In this research, we introduce a prototype robot named BBot that utilizes the movement momentum to achieve hopping motion. BBot is an improved version of our previous prototype [11] with additional hopping capability. Our goal is to develop a mobile robot that is able to traverse common terrains such as steps and stairs in an efficient manner. BBot consists of a lower body with wheels and a movable upper body connected to the main chassis by springs. Previous prototype suffers from impact recoil upon landing due to the rigid body structure. The movable upper body acts as an absorber to mitigate the impact force during landing. In addition, BBot uses the spring loaded movable upper body to generate lifting force to hop. We have chosen the two-wheeled structure due to the dynamic nature and simple design. During airborne, BBot uses the drive wheels to generate torque to control the angle of body tilt. As a result, it is resistant to external disturbances when traversing step terrains. Here we describe the hopping mechanism, motion characteristics, sensors, and control algorithms. We discuss the experimental results compare to the theoretical findings, as well as the limitations of the current model.

### Mathematical model

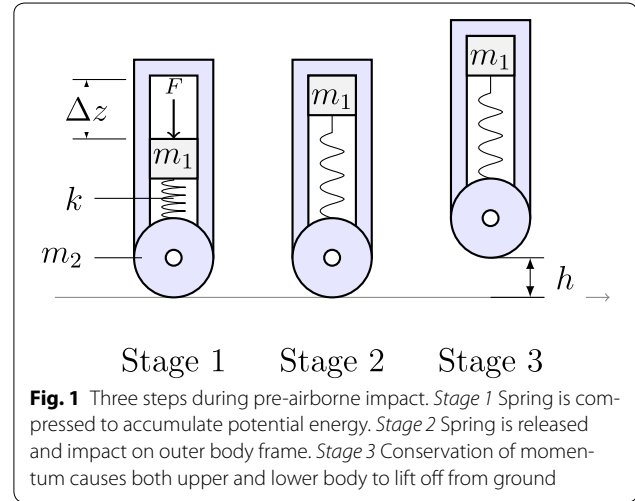
In this section, we develop the mathematical models and analyze the behavior of the robot during a hopping motion. The hopping cycle has three distinctive phases: (1) ground balance; (2) pre-airborne impact; and (3) airborne balance. The robot goes from ground to air and back in cycles. Figure 1 shows the three stages during a hop. To simplify the problem, we derived and analyzed separate models for each phase.

#### Ground balance phase

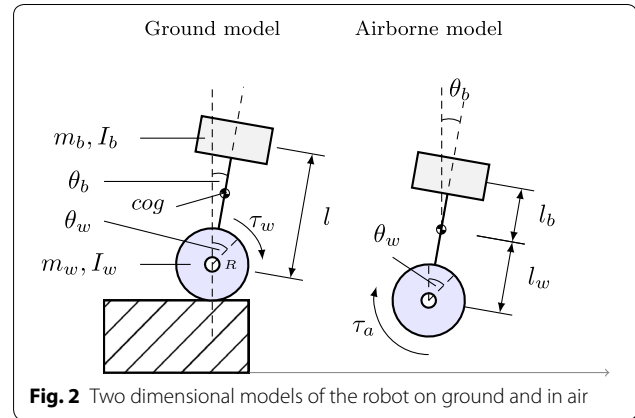
Figure 2 depicts the dynamic behavior of the robot on the ground and in the air in simplified two-dimensional models. Our 2D model consists of an upper body with mass  $m_b$  and moment of inertia  $I_b$  as well as a lower body of mass (wheel)  $m_w$  and moment of inertia  $I_w$ . The upper body rotates about the center axis of the wheel. A motor connected to the wheel generates torque  $\tau_g$ . The equations of motion on the ground are as follows:

$$(I_w + (m_b + m_w)R^2)\ddot{\theta}_w + m_b l_w R \cos\theta_b \ddot{\theta}_b - m_b l_w R \dot{\theta}_b^2 \sin\theta_b = \tau_g \quad (1)$$

$$(I_b + m_b l_w^2)\ddot{\theta}_b + m_b l_w R \cos\theta_b \ddot{\theta}_w - m_b g l_w \sin\theta_b = -\tau_g \quad (2)$$



**Fig. 1** Three steps during pre-airborne impact. Stage 1 Spring is compressed to accumulate potential energy. Stage 2 Spring is released and impact on outer body frame. Stage 3 Conservation of momentum causes both upper and lower body to lift off from ground



**Fig. 2** Two dimensional models of the robot on ground and in air

#### Pre-airborne impact phase

Three steps during pre-airborne impact. Stage (1) Compression of the spring increases potential energy. (2) Release of the spring impacts the frame of the outer body. (3) Conservation of momentum causes the upper and lower body to lift off from ground. Hopping results from conversion of the potential energy stored in the pre-compressed spring into vertical kinetic energy. The transition from ground to air takes place in three stages.

1. External force  $F$  pushes down on the upper body mass to increase potential energy in the spring. Stored potential energy is directly proportional to the spring constant  $k$  and the square of the compressed distance of the spring  $\Delta z$ .  $F$  is proportional to  $k$  and  $\Delta z$ .
2. Release of the upper body mass allows upward acceleration by converting spring potential energy into kinetic energy. Motion of the upper body mass is brought to a stop upon impact with the body frame ( $\Delta z$ ), unlike the free motion of the upper body mass

proposed in [8]. This design limits the magnitude of change in the center of mass that might affect the balance algorithm.

3. A perfectly inelastic collision occurs at the moment of impact. The Law of Conservation of Momentum mandates changes in the velocities of the upper body mass and the lower body according to the following equation:

$$m_1 v_1 = (m_1 + m_2) v_2 \quad (3)$$

If the initial velocity of the upper body generates enough energy, the lower body lifts off the ground. When the wheels rise above ground, the pre-airborne phase ends and the airborne phase begins.

Jumping height is directly proportional to the initial potential energy stored in the spring. Assuming potential energy is zero for the uncompressed spring ( $\Delta z = 0$ ), the potential energy of the system is

$$E_0 = \frac{1}{2} k \Delta z^2 - m_1 g \Delta z \quad (4)$$

Neglecting energy lost from friction, the total energy just before impact is

$$E_1 = \frac{1}{2} m_1 v_1^2 \quad (5)$$

The Law of Conservation of Energy dictates that the velocity of  $m_1$  prior to impact is

$$v_1 = \sqrt{\frac{2}{m_1} \left( \frac{1}{2} k \Delta z^2 - m_1 g \Delta z \right)} \quad (6)$$

Assuming a perfectly inelastic collision occurs after impact, the Law of Conservation of Momentum (e.g. (3)) requires that the velocity of the robot obeys the following equation:

$$v_2 = \frac{m_1}{m_1 + m_2} v_1 \quad (7)$$

The energy right after impact is

$$E_2 = \frac{m_1}{m_1 + m_2} E_0 \quad (8)$$

$$= \frac{m_1}{m_1 + m_2} \left( \frac{1}{2} k \Delta z^2 - m_1 g \Delta z \right) \quad (9)$$

The maximum height of the jump is

$$h = \frac{E_2}{(m_1 + m_2)g} \quad (10)$$

$$= \frac{m_1}{(m_1 + m_2)^2 g} \left( \frac{1}{2} k \Delta z^2 - m_1 g \Delta z \right) \quad (11)$$

Jump height increases in direct proportional to  $k$  and  $\Delta z$ .

### Airborne phase

We modelled the airborne robot as a downward pointing reaction wheel pendulum with a pivot at its center of mass (Fig. 1).

$$(I_b + m_b l_b^2 + m_w l_w^2) \ddot{\theta}_b = -\tau_a \quad (12)$$

$$I_w \ddot{\theta}_w = \tau_a \quad (13)$$

These two equations show that wheel acceleration is directly proportional to the torque  $\tau_a$  generated by the motor. Rearranging and solving the differential equations reveals that the wheel angle and the angle of body tilt are related as follows:

$$\theta_b = -\frac{I_w}{I_b + m_b l_b^2 + m_w l_w^2} \theta_w \quad (14)$$

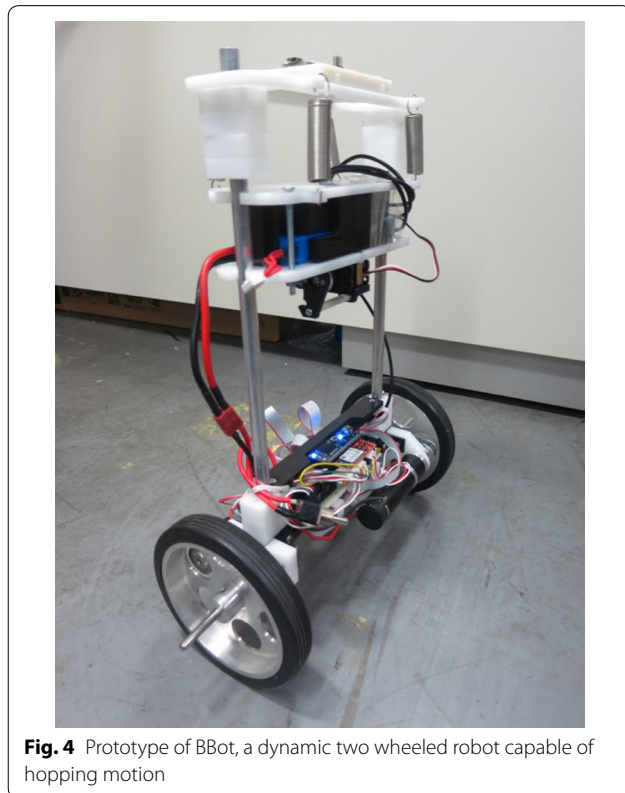
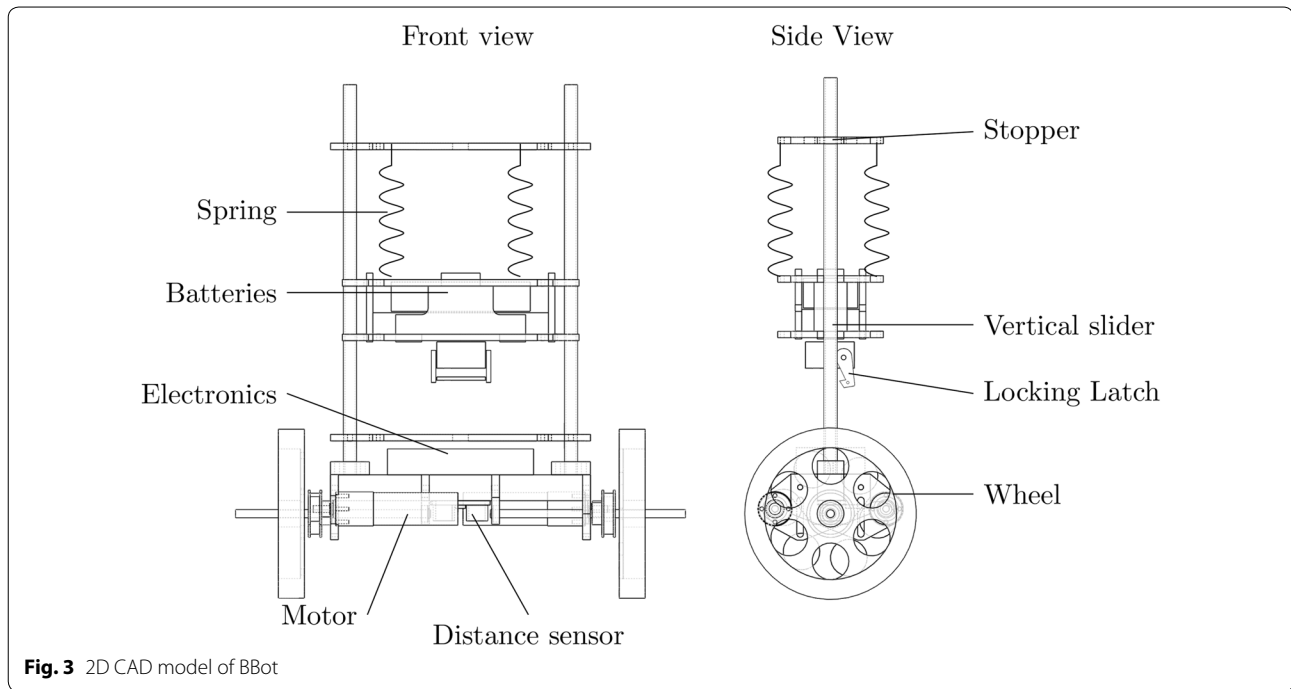
Accordingly, the body tilts in the direction opposite to the rotation of the wheels. Increasing the wheel's moment of inertia permits effective control over body tilt.

### Design and implementation

The jumping mechanism of the stair climbing robot developed by Kikuchi et al. [8] uses an upper body mass that is free to oscillate along the vertical axis to create momentum for lift. Maximum extension of the attached spring limits the movement of the upper mass. This design requires a tall body frame to accommodate the full movement of the mass. Large movement of the upper body mass would result in a major shift in the center of mass. Such a shift makes it harder to keep the robot stable and requires a strict control scheme to maintain balance. By contrast, our design limits the vertical movement of the upper body mass to reduce the magnitude of the shift in the center of mass. Figure 3 shows the 2D CAD model of BBot. Figure 4 shows the actual prototype of BBot. The upper body incorporates two 11.1 V lipo batteries for powering the motors and electronic components respectively. The upper body can slide vertically along a bar, and it is connected to the lower body frame with tension springs that pull upwards.

Tension springs instead of compression springs embedded in vertical slider eliminate possible force from friction. Spring tensioning mechanism is not implemented in this version of the prototype. In this prototype, the upper body is manually compressed and locked in place to prepare the robot in a jump-ready state. A servo activated latch controls the locking and releasing of the upper body. Weights ranging from 0.5 to 1 kg are attached to the upper body to test their effect on the jumping ability.

The lower body consists of two differential wheels powered separately by dc motors, a main electronic



control board, and an ultrasonic distance sensor. An inertia measurement unit (IMU), consisting of a 3-axis accelerometer and a 3-axis gyroscope, measures the angle of

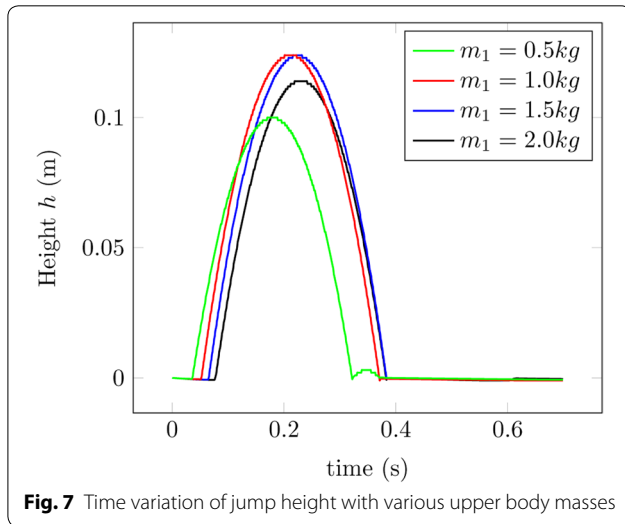
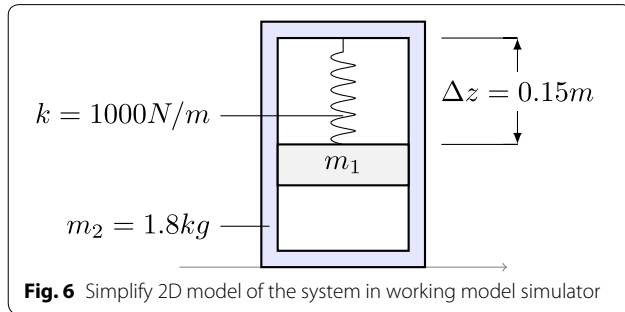
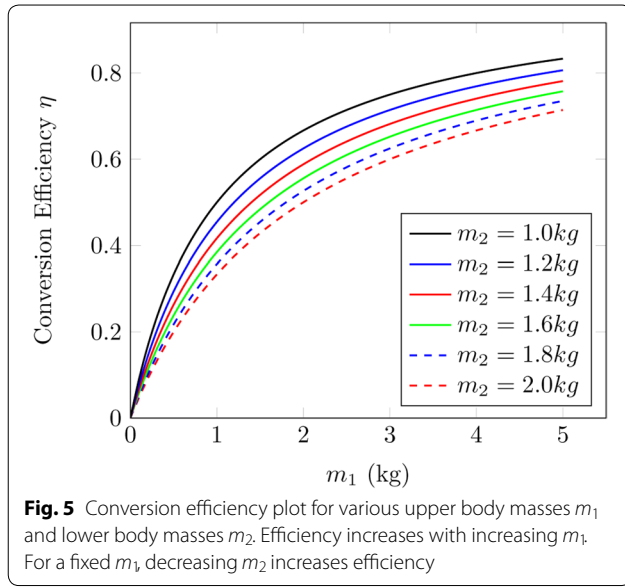
tilt. An ultrasonic distance sensor attached to the bottom measures the height the robot attains above ground. Lower body mass is minimal to make the hops as large as possible. We calculated conversion efficiency  $\eta$  as the ratio of the kinetic energy at takeoff to the energy stored in the compressed spring before takeoff [12]. The equation for conversion efficiency is

$$\eta = \frac{E_2}{E_0} = \frac{1}{1+r} \quad (15)$$

where  $r = m_2/m_1$ .

Figure 5 plots the conversion efficiency for various upper and lower body masses. Increasing upper body mass  $m_1$  increases the efficiency of energy conversion and height of jumping. For a fixed  $m_1$ , lower body mass  $m_2$  is inversely related to the efficiency of energy conversion. Consequently, increasing  $m_1$  while minimizing  $m_2$  leads to a higher height of jumping that is directly proportional to spring constant  $k$ .

Figure 6 shows a simplified 2D model constructed in Working Model 2D simulation software to simulate the effect of upper body weight on jump height. In this model, upper body mass  $m_1$  is constrained to move vertically inside  $m_2$ .  $m_2$  can move freely horizontally and vertically above ground.  $m_2$  and spring constant  $k$  are held constant. Gravitational force is set to  $9.8 \text{ m/s}^2$  and air resistance is neglected. For each simulation spring extension  $\Delta z$  is initialized to 0.15 m. Upon released the jump height of the system is recorded. Figure 7 plots the time variation of the



jump height with upper body masses varying from 0.5 to 2.0 kg. The results reveal that jump height is not linearly proportional to upper body mass. An upper body mass

larger or smaller than the optimal decreases the height of the jump. Larger upper body masses delay the transition from ground to the airborne state.

Figure 8 plots jump height  $h$  versus upper body mass  $m_1$ . Optimum  $m_1$  occurs at the maximum point of the curve:

$$m_{1\text{optimal}} = \frac{m_2 k \Delta z}{4m_2 g + k \Delta z} \quad (16)$$

The 12 V dc motor including gearing produces a maximum torque of 0.2 Nm. Body height  $l$  is 250 mm. Assuming the robot can recover from an angle of  $30^\circ$  when balancing on the ground, the maximum allowable weight of  $m_1$  is 1.6 kg. Our design uses four separate springs arranged in parallel with a spring constant of 250 N/m and  $m_2$  of 1.8 kg.  $m_1$  is set at 1.2 kg, close to the optimum of 1.22 kg. The H × D × W dimensions are 300 × 160 × 420 mm.

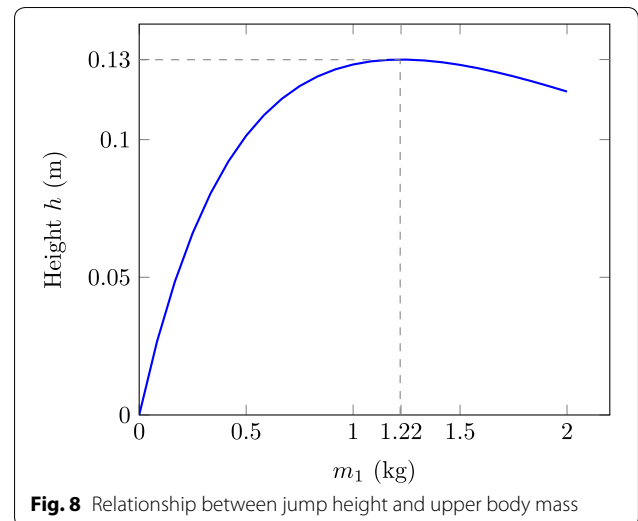
### Balance and attitude control

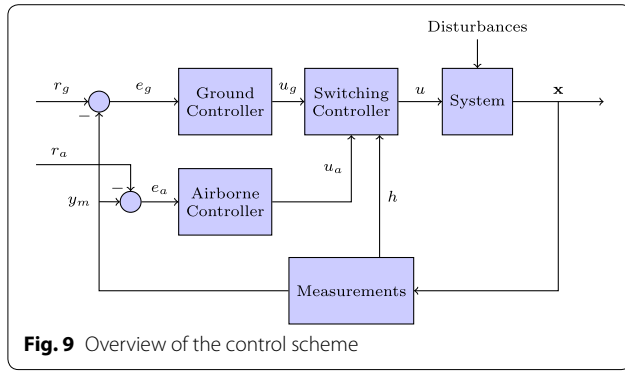
Static instability requires active control of body position during the airborne phase to ensure an upright landing. Two independent control schemes balance the robot on the ground and in the air (Fig. 9). We linearized the mathematical models derived in the previous section to design a full state-feedback controller [11]. We use the linear quadratic regulation method to determine the state-feedback control gain matrix  $K$  for both ground and airborne controllers.

### Ground balance control

On the ground, the linearized systems are in state-space form:

$$\dot{\mathbf{x}}_g = \mathbf{A}_g \mathbf{x}_g + \mathbf{B}_g \mathbf{u}_g \quad (17)$$





where  $\mathbf{x}_g = [\theta_w \ \dot{\theta}_w \ \theta_b \ \dot{\theta}_b]^T$  represents the states of the system,  $A_g$  is the state transition matrix,  $B_g$  is the input matrix and  $u_g$  is the input control torque from the motor:

$$u_g = \tau_g = K_g(\mathbf{r}_g - \mathbf{x}_g) \quad (18)$$

where  $\mathbf{r}_g = [\theta_{wref} \ \dot{\theta}_{wref} \ \theta_{bref} \ 0]^T$  represents the target references to the feedback controller and  $K_g$  is a vector of controller gains. The quadratic cost function to be minimized is defined as

$$J(u) = \int_0^\infty (\mathbf{x}^T Q \mathbf{x} + u_g' R u_g) dt \quad (19)$$

The values of weight matrix  $Q$  and  $R$  are manually tuned to yield a satisfactory response of a mathematical simulation constructed in Matlab.

Figure 10 shows the feedback control block diagram detailing the input and output of the system. BBot's differential drive requires separate controllers for left and right wheels. The input to the system consist of four state references, left wheel velocity  $\dot{\theta}_{wleft}$ , right wheel velocity  $\dot{\theta}_{wright}$ , body tilt angle  $\theta_b$  and body tilt angular velocity  $\dot{\theta}_b$ . Reference  $\theta_b$  and  $\dot{\theta}_b$  is set zero to keep the robot upright. Varying  $\dot{\theta}_{wleft}$  and  $\dot{\theta}_{wright}$  reference controls the forward, backward and turning movement on ground.

The yaw PD feedback controller controls yaw movement:

$$u_{yaw} = K_{yaw}(\mathbf{r}_{yaw} - \mathbf{x}_{yaw}) \quad (20)$$

$$\mathbf{r}_{yaw} = [\theta_{yawref} \ \dot{\theta}_{yawref}]^T \quad (21)$$

$$\mathbf{x}_{yaw} = [\theta_{yaw} \ \dot{\theta}_{yaw}]^T \quad (22)$$

where the yaw angle  $\theta_{yaw}$  and yaw velocity  $\dot{\theta}_{yaw}$  are the differences between left and right wheel angles and wheel velocities. The calculated yaw output is added to left and

right wheel input torque to yield the final input torque to the system.

### Airborne attitude control

When in the air, the robot behaves like a reaction wheel pendulum with a pivot at its center of mass. The drive wheels act as reaction wheels to create reaction torque that changes the angle of tilt. Equation 14 shows the angle of tilt  $\theta_b$  is directly proportional to the wheel angle  $\theta_w$ . Manipulating torque generated by the motor directly controls  $\theta_b$ . The controller is designed in a similar manner as the ground controller. The following control law controls the attitude in the air:

$$u_a = \tau_a = K_a[\theta_{bref} - \theta_b - \dot{\theta}_b - \dot{\theta}_w]^T \quad (23)$$

where  $K_a$  is the vector of feedback gains and  $\theta_{bref}$  is the reference body tilt angle in the air. Angular positions of the wheels  $\theta_w$  are not controlled, so they are not included in the feedback loop.

### Phase transition control

Data from the sonar distance sensor and the accelerometer determine the phase transition. Details of the approach are in the next section. A switching controller activates the corresponding feedback controller based on the current phase of the robot.

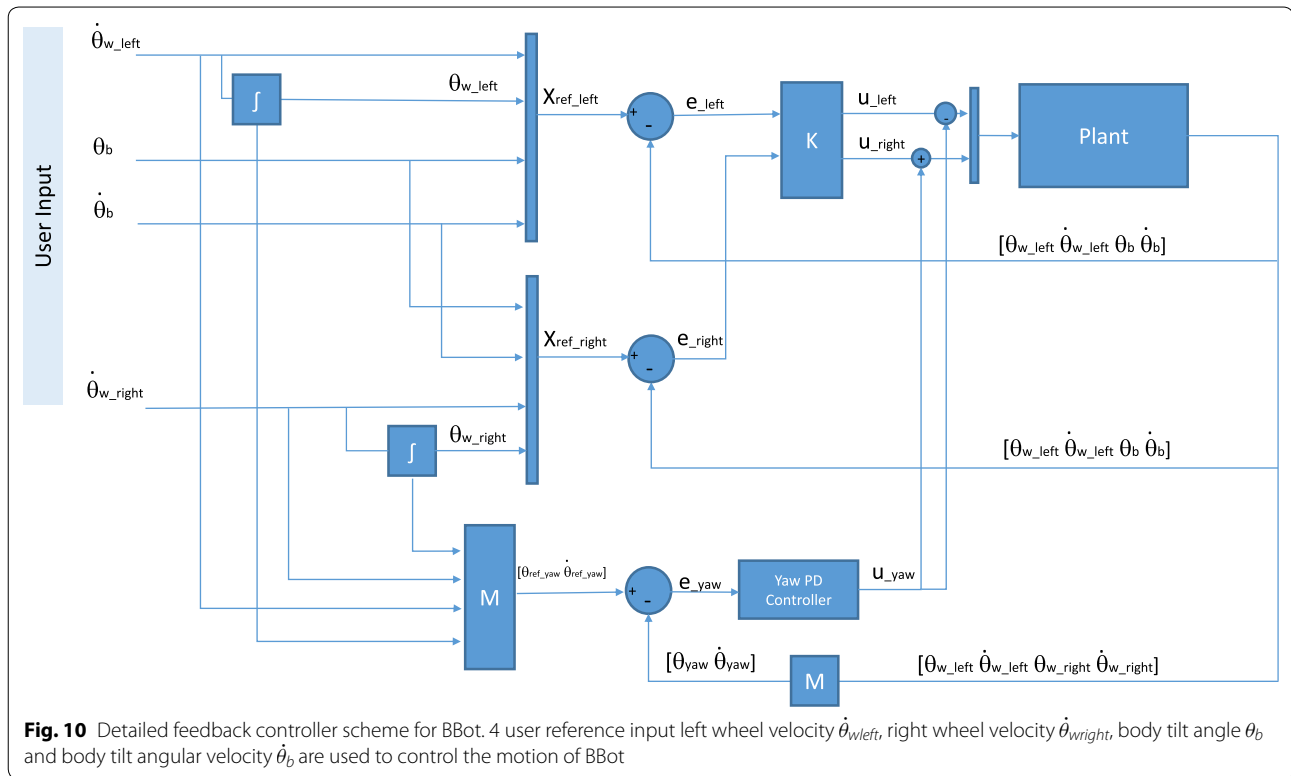
$$u = \begin{cases} u_g & \text{(if current phase = ground phase)} \\ u_a & \text{(if current phase = airborne phase)} \end{cases} \quad (24)$$

### Sensors and sensing approach

To have full-state feedback control for balance, the robot must know its state at every point in time in the control loop. An IMU measures body angle, and a combination of an ultrasonic sensor and an accelerometer measures height and state transition.

### Body angle detection

The IMU sensor comprises a 3-axis accelerometer and a 3-axis gyroscope to measure the angle of body tilt. In principle, for a known initial state, direct integration of the gyroscope measurement gives the angle of tilt. In reality, however, the MEMS gyroscope is subject to white noise and fluctuating bias over time. So, errors accumulate if direct integration is performed on gyro measurements. This is known as gyro "drift". One solution is to implement a sensor fusion scheme using an accelerometer as an additional reference measurement. Schemes such as a complementary filter [13], Kalman filter [14], or Direct Cosine Matrix (DCM) filter [15] can help estimate



rotational information of an IMU. Each type of filter has its pros and cons regarding speed, accuracy and complexity of implementation. We found that a Kalman filter sensor fusion scheme was more resilient and performed better during hopping and at the impact of landing, where accelerometer measurements tend to peak and overshoot. The Kalman filter also performed better under the influence of linear acceleration, especially when the robot platform was in motion.

Figure 11 compares the estimated angles for the Kalman, DCM, and complementary filters when the IMU sensor is accelerated back and forth in a fixed direction at a horizontal angle of zero degrees. The lower graph shows the corresponding raw accelerometer data. We found that a complementary filter that estimates the angle based on weighted angles from accelerometer and gyro sensor performed poorly during acceleration. The DCM filter fluctuated less than the complementary filter, but the influence from linear acceleration was still obvious. By contrast, the Kalman filter showed little fluctuation in angle estimation.

#### Height and phase transition detection

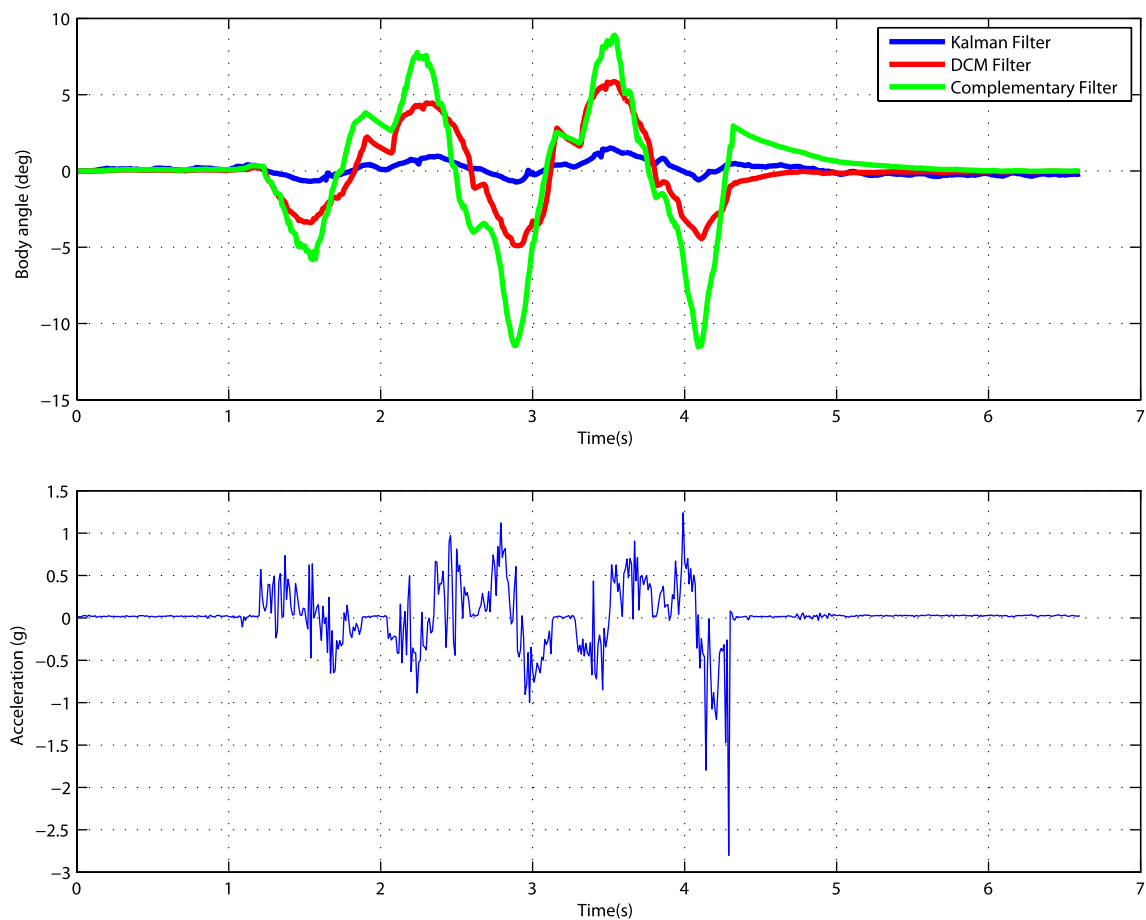
We have mounted an ultrasonic distance sensor on the bottom of the body, facing downwards towards the ground to measure the height of the robot from the ground. Detection of the transition from ground

to airborne phase and vice versa is crucial to maintain balance throughout the jumping motion. A simple approach is to set a threshold to determine the phase transition. This approach, however, has its drawback. Specifically, the sensor reading increases when the robot leans forward during acceleration. Increased sensor reading triggers the threshold settings and causes a false detection in phase transition. For larger tilt angle ( $>60^\circ$ ), the ultrasonic waves reflects away from the sensor and causes incorrect readings.

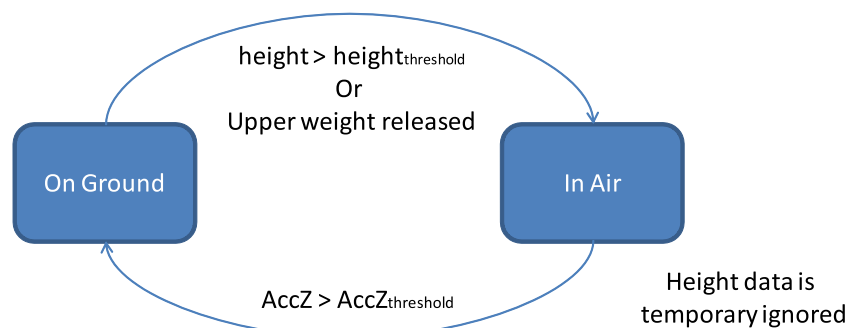
In order to build a more reliable phases detection scheme, we use the vertical axis accelerometer data to detect impact indicating a phase transitions. Figure 14 shows the accelerometer data at the instant of an impact. We can determine an impact event by detecting a sudden spike in accelerometer data. Figure 12 shows a flow diagram of a more reliable approach in detecting phase transition using both ultrasonic sensor data and accelerometer data.

#### Experiments and discussion

BBot is a self contained, with electronics and power supply encased in the chassis. A host pc controls and communicates with BBot via bluetooth connection. Realtime sensor data is streamed to the host pc and logged at a rate of 100 Hz.



**Fig. 11** Angle estimation comparison between Kalman filter, DCM filter and complementary filter. The sensor is accelerated back and forth in a fix direction at tilt angle of  $0^\circ$ . Kalman filter shows little fluctuation in angle estimation under the influence of linear acceleration noise. The lower graph shows the raw linear acceleration observed by the sensor

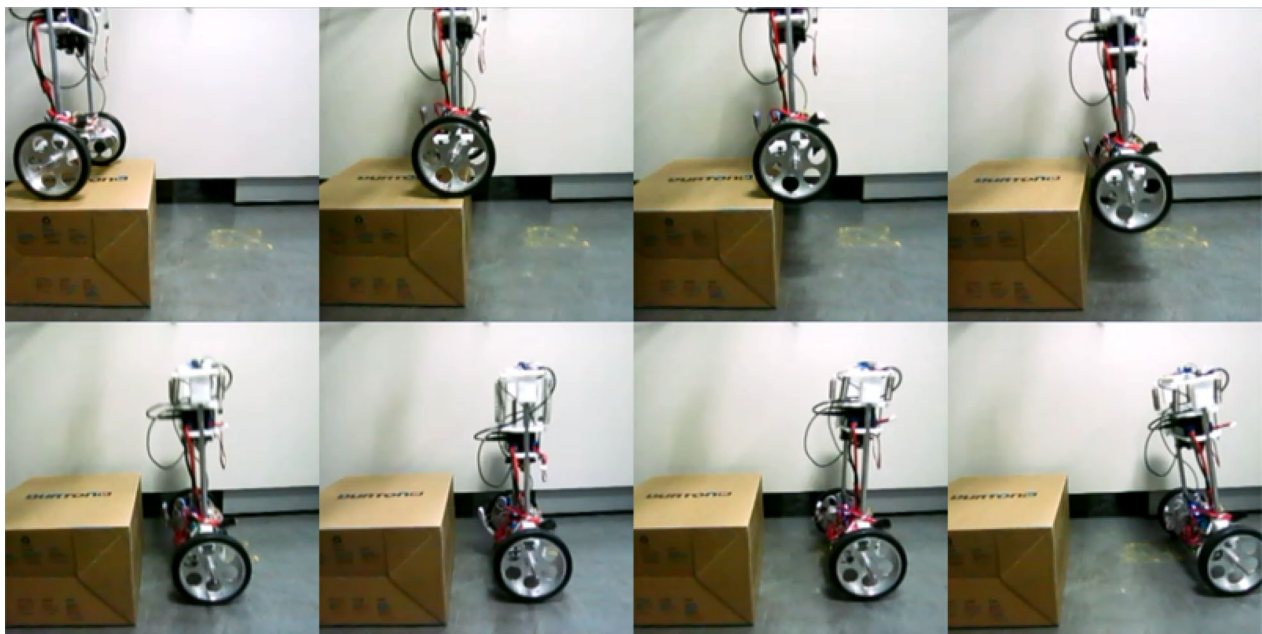


**Fig. 12** Flow diagram showing the phase transition detection scheme using both ultrasonic sensor and accelerometer data

### Step traversing experiment

In this experiment, the robot travels down a 17 cm step. The step height is equivalent to the average height of a

stair step. Additional file: 1 (Bbot-StepTraverseExperiment) shows the video of the step traversing experiment. Figure 13 shows the snap shots of the motion. Figure 14



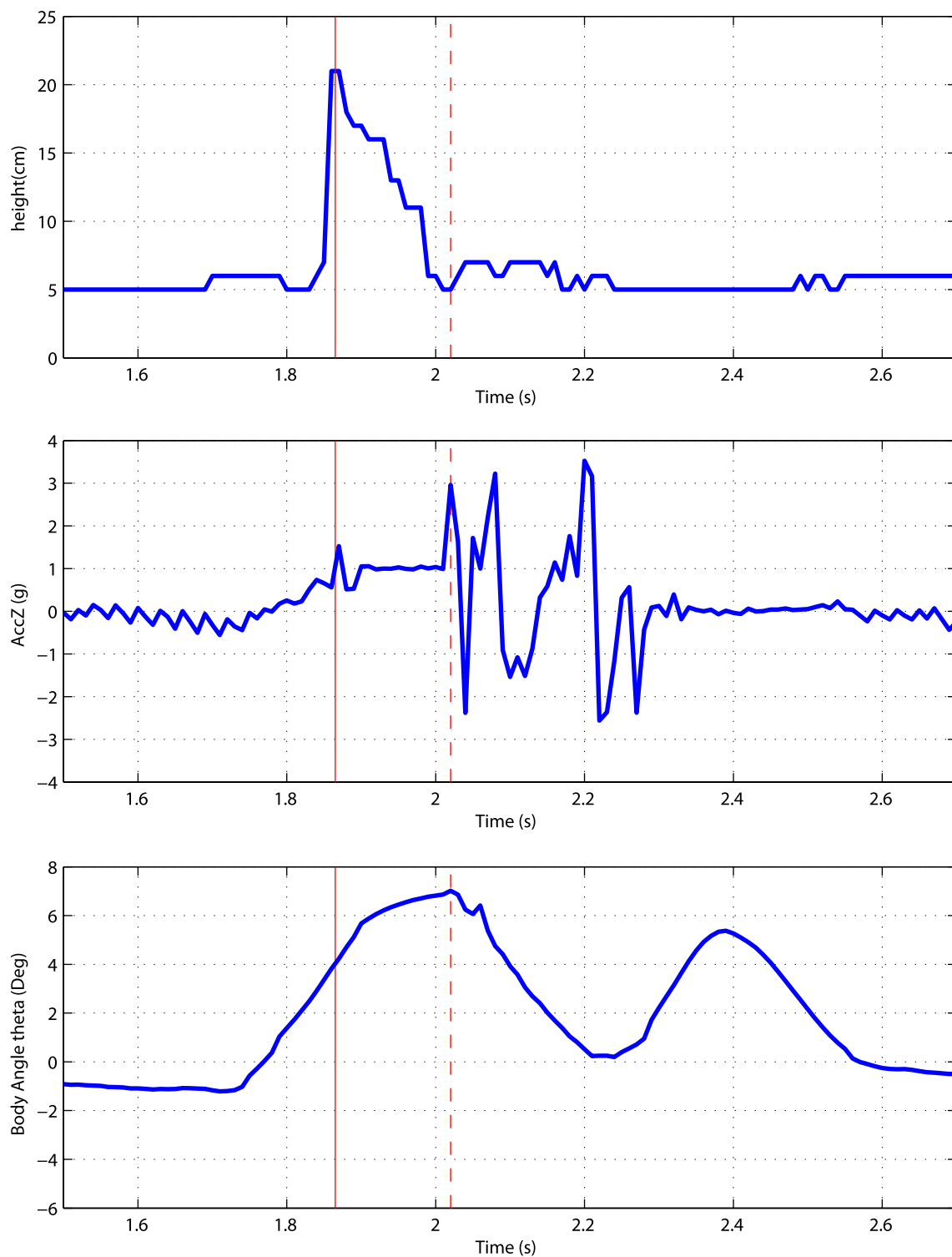
**Fig. 13** The robot is driven down a step terrain of height 17 cm. While in air, the robot uses the drive wheels to maintain its body angle close to vertical. Using this approach, the robot prevents its motor from saturation and hence possesses sufficient torque to balance itself upon landing

shows the corresponding raw sensor data and the body tilt angle plot against time. Height data plots the height calculated from the ground to the sonar sensor. The height data has an offset of 5 cm above ground when the robot is balancing still. AccZ graph plots the acceleration in the robot frame z axis (pointing up) with gravity subtracted. At time  $t = 1.86$  s, sudden increased in height measurement indicates that the robot is currently airborne. The controller switches into airborne control mode and applies balancing torque generated through the rotating wheels. We have set the reference body tilt angle for airborne controller to a small positive value (+6 degrees), to tilt the robot backwards during airborne. The reason for this is to compensate the forward momentum during landing and reduce the torque needed to balance upon landing. This is analogous to landing with feet in front and uses momentum to bring the body to a neutral position. At time  $t = 2.02$  s, the wheel hits the ground and causes large fluctuation in the accelerometer readings. Phase changed is detected and the robot switches into ground mode to keep its balance. From the figure, the fluctuation in accelerometer reading does not affect the body angle estimation due to robustness of the Kalman filter. At the moment of impact upon landing, we observe the upper body mass moves downward. This motion acts as an absorber to absorb the impact force from the ground and mitigate any recoil effect which is observed in the previous rigid prototype. From the height vs time plot, there is a false positive indication of increase

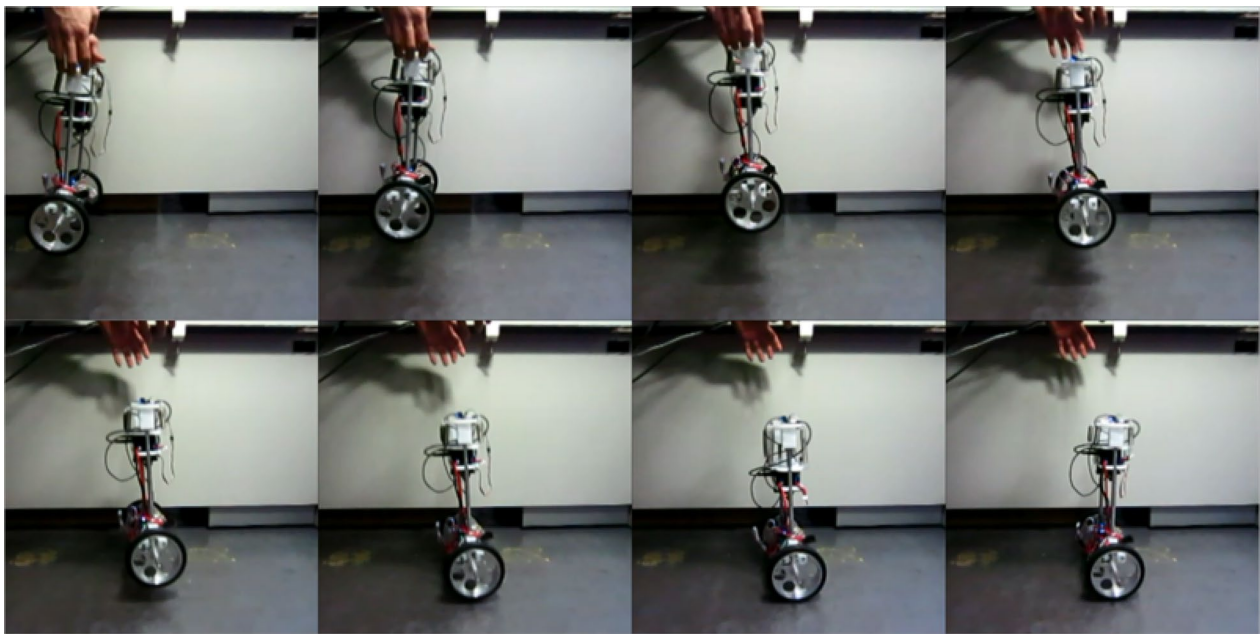
height right after landing. This is the effect explained in the previous section where the ultrasonic waves bounds away from the ground when the body tilt angle is large.

#### Toss landing experiment

We have devised another experiment to confirm the effectiveness of attitude control during airborne. In this experiment, the robot is manually tossed with an initial angular rotation to test the capability to recover from such situation and land safely. Without attitude control, initial rotation causes a tilt away from the vertical. Upon landing the robot is not be able to recover from large tilt angle. With attitude control turned on, the controller constantly generates recovery torque to keep the body angle close to the vertical axis. The small tilt angle on landing prevents the motor from saturating. The motor is able to generate sufficient torque to keep the robot in balance. Additional file: 2 (BBot-TossExperiment) shows the video of the toss landing experiment. Figure 15 shows the snapshots of the motion of the robot in this experiment. Figure 16 shows the height, vertical acceleration, body angle and motor pwm duty plot against time. Red solid line in the graph shows the instant when the robot is released. At time  $t = 4.5$  s the robot reaches peak height and starts to free fall. The airborne controllers maintains a positive body tilt angle in air. Landing impact happens at time  $t = 4.7$  s. The toss experiment introduces large rotational torque to the robot. From time  $t = 4.7$  s  $\sim$  5.5 s, the robot rocks back and forth to



**Fig. 14** Height, raw accelerometer plot against time during step traverse event. The graph body angle (filtered) against time shows the robustness of Kalman filter fusion scheme towards fluctuating accelerometer measurements. *Solid red line* and *dotted red line* indicate the beginning and the end of an airborne phase



**Fig. 15** Snapshot of the robot manually tossed with an initial angular rotation

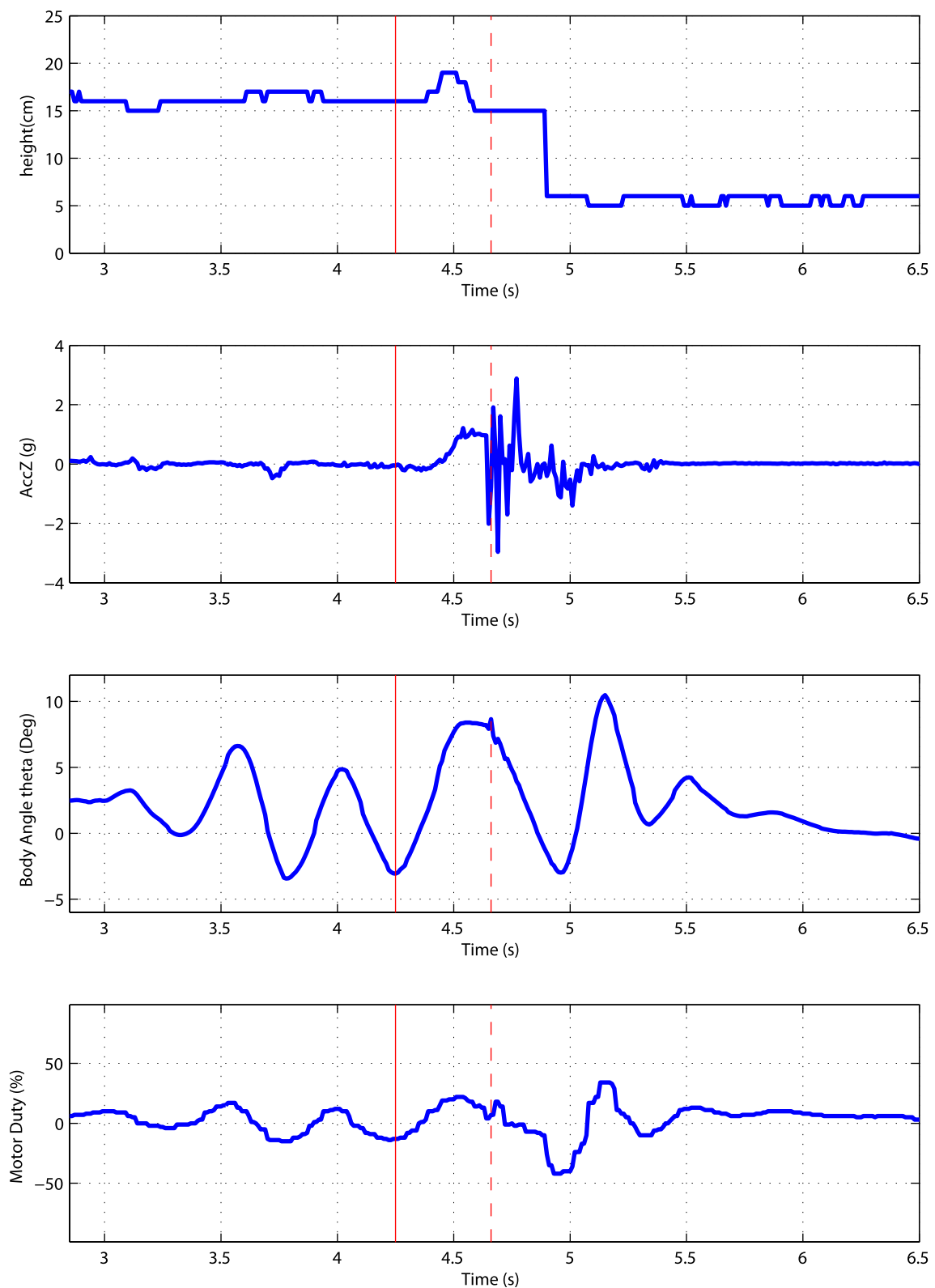
keep balance before settling down on a stable upright position at  $t > 6$  s. The motor duty graph shows that the pwm duty for the left motor. Pwm duty is proportional to the torque apply to the motor. The pwm duty is kept below 50 % at any instant of time indicates that the motor is below saturation limit.

### Hopping experiment

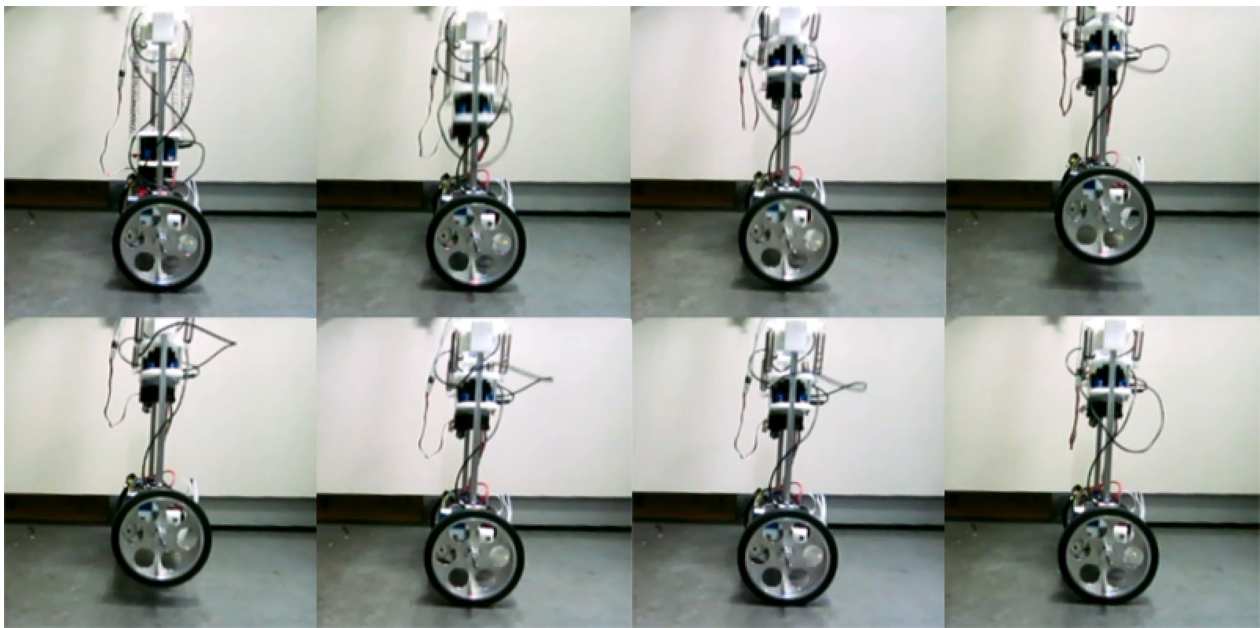
In the last experiment, we present the hopping action using our proposed method. The springs are pre-tensioned manually by pushing down the upper body weight. A catch and release mechanism locks the upper body weight in place. Remote command from the host pc unlocks the locking mechanism to release the upper body weight. Additional file: 3 (BBot-HoppingExperiment) shows the video of the hopping experiment. Figure 17 shows the continuous snapshots of a hopping action. Figure 18 shows the sensor data plots against time. When the lock is released, pre-tensed springs accelerate the upper body weight. Impact on the stopper, occurs at time  $t = 2.38$  s, converts into lifting force and causes the robot to jump. The robot detects the impact event and switches into airborne balancing mode. The rest of the action is similar to step traversing. At time  $t = 2.55$  s, landing is detected and the controller switches back to ground balance mode. The robot successfully performs

a hopping motion. From Fig. 18, the maximum hopping height of this model is roughly 4 cm. The actual jump height is less than the value calculated in the simulation due to the following reasons: (1) Friction from the manually constructed sliding joint connecting the upper and lower body contributes to energy lost. (2) The springs used in the robot do not have the exact spring constant as assumed in the simulation. (3) Contrast to the assumption, the impact collision is not fully inelastic. The impact force does not convert fully into lifting force. (4) The simulation does not take into account friction encountered in physical world. Nevertheless, the prototype robot demonstrated the possibility to achieve hopping motion using proposed approach.

From these experiments, we observed a few short comings on the current prototype. In order to improve the hopping height of the robot, the current sliding joint has to be replaced by a lubricated ball bearing sliding joint to further reduced the friction. Jump height of the current prototype is too small to have practical application in real world environment. In order to be able to jump up a step, the robot needs to be able to have a jump height of at least 17 cm. To achieve higher jump height, we plan to scale up the prototype to accommodate larger springs. We observed recoil effect during impact of the upper body mass and the body frame. The partially inelastic collision differs from the



**Fig. 16** Data plot of robot tossing experiment. *Red solid line* above shows the instant when the robot is released. The robot generates correcting torque to prevent it body angle to tilt away from vertical axis so that the body angle upon landing is within recoverable range. *Dashed line* shows the instant when the robot hits the ground



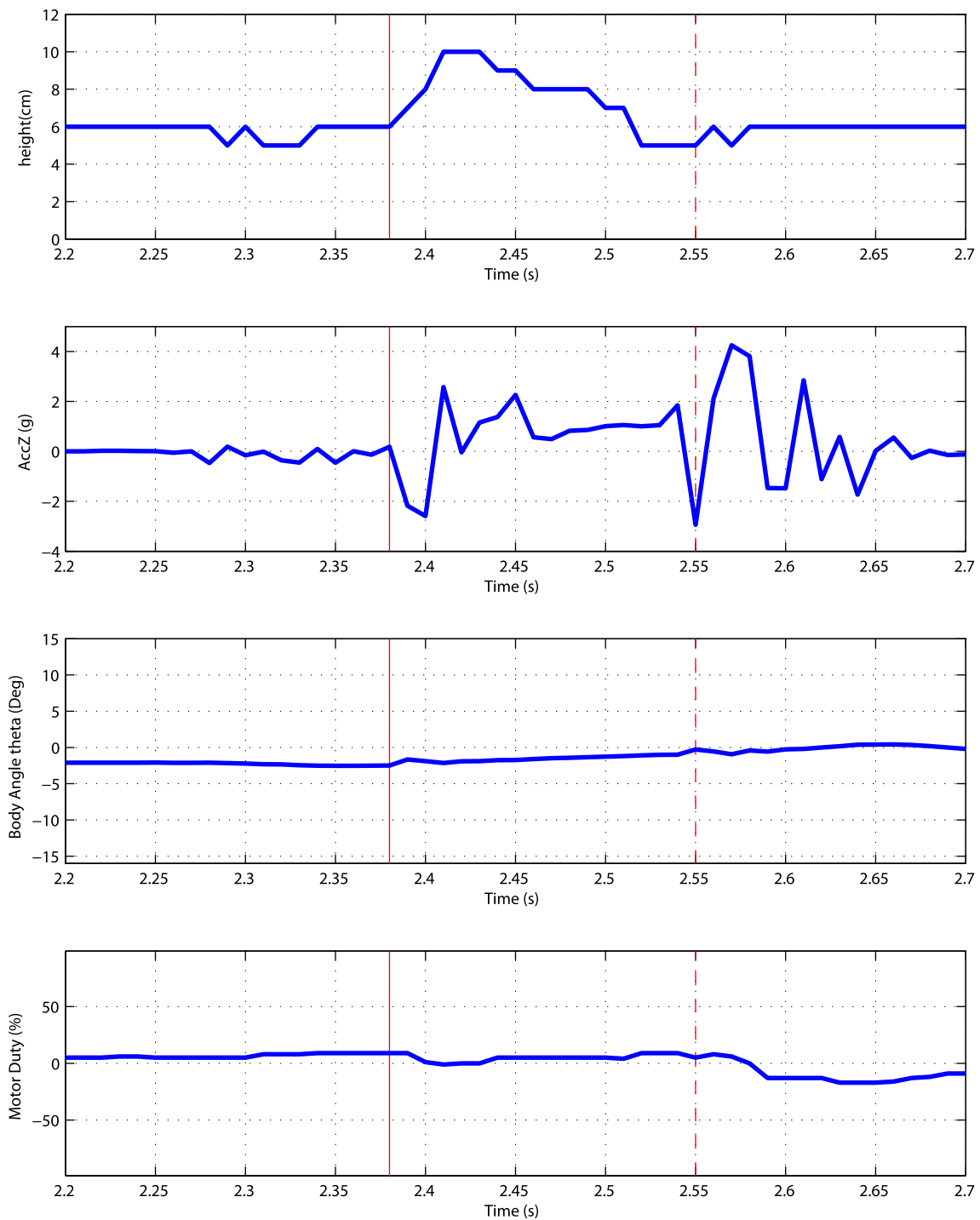
**Fig. 17** Hopping motion demonstrated by the prototype robot

simulation assumption where the impact is perfectly inelastic. For the next version of the prototype, we plan to use a latching mechanism to enable locking of the upper body mass upon impact with the body frame. The reference body tilt angle is manually tuned and hard coded to enable a more efficient landing. As for future work, the relationship between velocity, jump height and body tilt angle needs to be investigated so that the robot can dynamically calculate the optimal tilt angle for different jump height. Current sonar sensor used to detect jump height suffers from false positive readings when the body is tilted away from the vertical. We are investigating the possibility of using laser scanner or small radar chips which are capable of measuring distance at a higher accuracy and reliability.

## Conclusions

Tradeoff between robustness and increased system complexity is the main challenges faced by many mobile robots. Highly dynamic and robust robots tend to have increased complexity as well as cost of development and manufacturing. Simpler robots, on the other hand, have limited capability to be put into practical use. In order to tackle this problem, we have created BBot, a dynamically stable two wheeled robot capable of hopping and negotiating step terrains. The dynamic nature and relatively simple mechanical construction enables

the robot to have a good compromise between robustness and complexity. We have presented the theoretical analysis, design and mechanism of the robot, choice of sensors and sensing approach as well as the experiment results of a working prototype. The robot consists of a two wheeled lower body platform and a spring loaded movable upper body mass. Hopping motion is achieved through the impact force produced by releasing pre-tensed springs. An IMU detects the body tilt angle, along with an ultrasonic distance sensor to detect the height above ground. The accelerometer in the IMU measures the impact force to determine the beginning and the end of a hopping and landing phases. Due to statically unstable property of the robot, the attitude has to be actively controlled during airborne to ensure stability upon landing. While in air the body angle is controlled using the torque generated through the rotation of the drive wheels. Upon landing impact detection through onboard sensors, the controller switches to ground balance mode to maintain balance. Current prototype is capable of negotiating step terrains of a height of 17 cm, which is similar to normal stair height, and is capable of hopping 4 cm above ground. The hopping height is significantly less than simulated results due to mechanical constraints and perfect world assumption in the simulation. For future work, we plan to improve the mechanical design,



**Fig. 18** Raw data plots of the hopping motion. Red solid line indicates the instant when the upper body weight hits the stopper ( $t = 2.38$  s). Dashed line indicates the instant when the robot lands ( $t = 2.55$  s). The hopping motion lasted for 0.17 ms

increase the size of the prototype to accommodate larger springs to increase the lifting force and enable the robot to jump at a higher height. We plan to take advantage of the dynamic nature of the robot to travel up and down

step terrains, leap over gap obstacles in a swift and reliable manner. We are also investigating laser scanners or radar sensors in replace of sonar sensor for height detection.

## Abbreviations

IMU: inertia measurement unit; DCM: direct cosine matrix.

## List of symbols

$m_b$	mass of body
$m_w$	mass of wheels
$I_b$	inertia of body around body center of mass
$I_w$	inertia of wheels around wheel center of mass
$R$	wheel radius
$l_b$	length of upper body's center of mass to robot's center of mass
$l_w$	length of wheel axis to robot's center of mass
$l$	length of upper body's center of mass to wheel axis
$g$	gravity
$\theta_b$	tilt angle of robot body
$\theta_w$	rotational angle of wheels
$\tau_a$	motor torque in air
$\tau_g$	motor torque on ground

## Additional files

**Additional file 1.** BBot-StepTraverseExperiment Description: Movie showing step traverse experiment of BBot.

**Additional file 2.** BBot-TossExperiment Description: Movie showing tossing experiment of BBot.

**Additional file 3.** BBot-HoppingExperiment Description: Movie showing the hopping experiment of BBot.

## Authors' contributions

HEY developed the methodology, performed the analysis, constructed the prototype and wrote the manuscript. Both authors read and approved the final manuscript.

## Competing interests

The authors declare that they have no competing interests.

Received: 27 November 2015 Accepted: 17 February 2016

Published online: 03 March 2016

## References

- Kovac M, Fuchs M, Guignard A, Zufferey J-C, Floreano D (2008) A miniature 7g jumping robot. In: 2008 IEEE international conference on robotics and automation. ICRA 2008. pp 373–378. doi:[10.1109/ROBOT.2008.4543236](https://doi.org/10.1109/ROBOT.2008.4543236)
- Scarfogliero U, Stefanini C, Dario P (2006) A bioinspired concept for high efficiency locomotion in micro robots: the jumping robot grillo. In: Proceedings 2006 IEEE international conference on robotics and automation. ICRA 2006. pp. 4037–4042. doi:[10.1109/ROBOT.2006.1642322](https://doi.org/10.1109/ROBOT.2006.1642322)
- Zhao J, Xi N, Gao B, Mutka MW, Xiao L (2011) Development of a controllable and continuous jumping robot. In: 2011 IEEE international conference on robotics and automation (ICRA), pp 4614–4619. doi:[10.1109/ICRA.2011.5980166](https://doi.org/10.1109/ICRA.2011.5980166)
- Niiyama R, Nagakubo A, Kuniyoshi Y (2007) Mowgli: a bipedal jumping and landing robot with an artificial musculoskeletal system. In: 2007 IEEE international conference on robotics and automation, pp 2546–2551. doi:[10.1109/ROBOT.2007.363848](https://doi.org/10.1109/ROBOT.2007.363848)
- Lambrech BGA, Horschler AD, Quinn RD (2005) A small, insect-inspired robot that runs and jumps. In: Proceedings of the 2005 IEEE international conference on robotics and automation. ICRA 2005, pp 1240–1245. doi:[10.1109/ROBOT.2005.1570285](https://doi.org/10.1109/ROBOT.2005.1570285)
- Armour R, Paskins K, Bowyer A, Vincent J, Megill W (2008) Jumping robots: a biomimetic solution to locomotion across rough terrain. *Bioinspir Biomim* 3(3):039801
- Stoeter SA, Papanikolopoulos N (2006) Kinematic motion model for jumping scout robots. *IEEE Trans Robot* 22(2):397–402. doi:[10.1109/TRO.2006.862483](https://doi.org/10.1109/TRO.2006.862483)
- Kikuchi K, Sakaguchi K, Sudo T, Bushida N, Chiba Y, Asai Y (2008) A study on a wheel-based stair-climbing robot with a hopping mechanism. *Mech Syst Signal Process* 22(6):1316–1326. doi:[10.1016/j.jymssp.2008.03.002](https://doi.org/10.1016/j.jymssp.2008.03.002) (Special Issue: Mechatronics)
- Schmidt-Wetekam C, Bewley T (2011) An arm suspension mechanism for an underactuated single legged hopping robot. In: 2011 IEEE international conference on robotics and automation (ICRA), pp 5529–5534. doi:[10.1109/ICRA.2011.5980339](https://doi.org/10.1109/ICRA.2011.5980339)
- Railbert M, Blankespoor K, Nelson G, Playter R (2008) Bigdog, the rough-terrain quadruped robot. In: The international federation of automatic control
- Yap HE, Hashimoto S (2012) Development of a stair traversing two wheeled robot. In: 2012 IEEE/RSJ international conference on intelligent robots and systems (IROS). pp 3125–3131. doi:[10.1109/IROS.2012.6385767](https://doi.org/10.1109/IROS.2012.6385767)
- Burdick J, Fiorini P (2003) Minimalist jumping robots for celestial exploration. *Int J Robot Res* 22(7–8):653–674. doi:[10.1177/02783649030227013](https://doi.org/10.1177/02783649030227013). <http://jrr.sagepub.com/content/22/7-8/653.full.pdf+html>
- Batista P, Silvestre C, Oliveira P, Cardeira B (2010) Low-cost attitude and heading reference system: filter design and experimental evaluation. In: 2010 IEEE international conference on robotics and automation (ICRA), pp 2624–2629. doi:[10.1109/ROBOT.2010.5509537](https://doi.org/10.1109/ROBOT.2010.5509537)
- Lin Z, Zecca M, Sessa S, Bartolomeo L, Ishii H, Takanishi A (2011) Development of the wireless ultra-miniaturized inertial measurement unit wb-4: preliminary performance evaluation. In: 2011 Annual international conference of the IEEE Engineering in Medicine and Biology Society, EMBC, pp 6927–6930. doi:[10.1109/IEMBS.2011.6091751](https://doi.org/10.1109/IEMBS.2011.6091751)
- Madgwick SOH, Harrison AJL, Vaidyanathan R Estimation of imu and mag orientation using a gradient descent algorithm. In: 2011 IEEE international conference on rehabilitation robotics (ICORR), pp 1–7 (2011). doi:[10.1109/ICORR.2011.5975346](https://doi.org/10.1109/ICORR.2011.5975346)

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)