# Agile and stable running locomotion control for an untethered and one-legged hopping robot

Barkan Ugurlu[1] · Emre Sariyildiz[2] · Takao Kawasaki[3] · Tatsuo Narikiyo[3]

## Abstract

This paper is aimed at presenting a locomotion control framework to realize agile and robust locomotion behaviors on conventional stiff-by-nature legged robots. First, a trajectory generator that is capable of characterizing angular momentum is utilized to synthesize reference CoM trajectories and associated force inputs, in accordance with the target locomotion profile. Second, the controller evaluates both force and position errors in the joint level, using a servo controller and an admittance control block. The trade-off between the position and force errors is naturally adjusted via admittance control coefficients. Implementing the controller on a 4-link, 3-jointed one-legged robot, we conducted several balancing and running experiments under challenging conditions; e.g., balancing on a moving cart, balancing on a surface with varying orientation, running on a flat surface, running on an inclined surface. The experimental study results indicated that the locomotion controller enabled the robot to perform untethered one-legged running and to maintain its balance when subject to disturbances.

**Keywords** One-legged running · Active compliance · Angular momentum · Jumping robot

## 1 Introduction

Agility and robustness are the two main desired locomotion characteristics that are expected in a contemporary legged robot. Recently, we have witnessed the grow of humanoids with such qualities; e.g., on top of the existing Petman architecture (Nelson et al. 2012), Boston Dynamics developed a new humanoid robot that has advanced locomotion capabilities. Companies such as Honda Motor Company and Toyota Motor Corporation, and research institutes such as IIT (Istituto Italiano di Tecnologia), AIST (National Insti-

tute of Advanced Industrial Science and Technology) and KAIST (Korea Advanced Institute of Science and Technology), also significantly contributed to the legged locomotion literature with their state-of-the-art robots (Tajima et al. 2009; Takenaka et al. 2009; Cho et al. 2011; Kajita et al. 2007; Tsagarakis et al. 2017).

Generally speaking, agility is defined as the robot's ability to generate acceleration since it enables a certain change in its velocity (Bowling 2011). The change in velocity triggers the mechanisms that regulate gait phase transitions in mammals as well, i.e., from walking to running or vice versa (Jordan and Newell 2008; Diedrich and Warren 1995; Cavagna et al. 1988). In doing so, they are able to optimally manage mechanical energetics and load determinants (Kung et al. 2018). Therefore, from the agility perspective, the running locomotion appears to be an important property when it comes to imitate the locomotion of mammals. Furthermore, the running locomotion includes flight phases which may help the legged systems to maintain its balance when perturbed (Kamioka et al. 2017).

In emulating the untethered running motion on legged robots, Raibert's seminal work stressed the importance of actuation as he achieved continuous one-legged running with a hydraulically actuated robot (Raibert et al. 1984). Ahmadi and Buehler (2006) adapted a similar strategy for a hopper

✉ Barkan Ugurlu
barkan.ugurlu@ozyegin.edu.tr

Emre Sariyildiz
emre@uow.edu.au

Tatsuo Narikiyo
n-tatsuo@toyota-ti.ac.jp

[1] Department of Mechanical Engineering, Ozyegin University, 34794 Istanbul, Turkey

[2] School of Mechanical, Materials, Mechatronic and Biomedical Engineering, University of Wollongong, Wollongong, NSW 2522, Australia

[3] Department of Advanced Science and Technology, Toyota Technological Institute, Nagoya 468-8511, Japan

that had a spring-supported leg; therefore, they proved that the use of passive compliance could compensate for the need of high power actuation. Upon these impressive pieces of research, researchers demonstrated untethered and continuous one-legged running locomotion behavior either via the utilization of novel actuators (Wensing et al. 2017), or via passive compliance (Batts et al. 2016; Ugurlu et al. 2014). This result is not surprising; both springs and high-bandwidth actuators are capable of doing negative work that is needed mitigate impacts at the touch-down.

Being quite useful in their own functionality domain, a great majority of the above mentioned robots possess custom-made actuators or they make use of running-specific design arrangements; i.e., they may not fully respond to low-speed tasks such as walking and loco-manipulation. Therefore, our first research objective in this work is to design a running locomotion controller that can be implemented to a classical humanoid with conventional off-the-shelf actuators, e.g., Cho et al. (2011) and Kajita et al. (2007).

In synthesizing a controller for legged systems, whole-body control approaches are proved to be very useful. For instance, Wensing and Orin (2015, 2013) proposed to regulate the angular momentum and simulated a humanoid that can run while being subject to disturbances. Prior to their work, Kajita et al. (2007) synthesized a controller, named Resolved Momentum Control, to achieve bipedal running. Bellicoso et al. (2018) addressed whole-body control by means of an online nonlinear optimization algorithm. Ott et al. (2011) developed a posture controller for a torque controlled humanoid via the utilization of contact force optimization. Takenaka et al. (2009) proposed a cascaded controller methodology that achieves dynamics error compensation for the real-time control of ASIMO.

The use of whole-body control approach is well validated for reliable legged locomotion; however, these controllers mostly rely on online optimization which is computationally expensive to run in real-time (Zhu et al. 2020; Neunert et al. 2016). Considering the fact that the built-in computers could need to run other computationally heavy tasks, e.g., vision sensing, taking up a certain portion of computational power may not be a viable solution. Therefore, the second research objective in this work is to design a running locomotion controller that can be easily executed in real-time, preferably within a 1 ms sampling time while using an off-the-shelf embedded computer.

When it comes to designing efficient locomotion controllers, it is important to capture the dominant characteristics of running locomotion. To that end, the importance of compliance is addressed to characterize basic dynamics of walking and running for legged systems (Ferris et al. 1998; Gollhofer et al. 1984; Hurst and Rizzi 2008). This is due to the fact

that the use of compliance enables the regulation of inertial forces acting on the CoM (Center of Mass) (Geyer et al. 2006). The regulation of inertial forces was well verified for robot locomotion control, in particular, for landing the feet (Sakka and Yokoi 2005; Kim et al. 2007). With this in mind, the researchers made use of various active compliance schemes to attain effective and feasible locomotion controllers (Buschmann et al. 2009; Fujimoto and Kawamura 1998; Buchli et al. 2009; Boaventura et al. 2012).

Inspired by the efficacy of the active compliance schemes, we propose a computationally inexpensive compliant locomotion control scheme in which the trade-off between the position and force constraints can be managed while ensuring stability. The novelty of the presented study is three-fold: (i) a compliant locomotion controller with proven stability is proposed. In the literature, compliant locomotion controllers exist (Caron et al. 2019; Kajita et al. 2010; Kamioka et al. 2017); however, a stability proof that can indicate the interplay between stability and rendered compliance is provided in this paper. (ii) A trajectory generator that can characterize the varying inertia and associated angular momentum is proposed. Compared to flywheel based methods, the method encapsulates the angular momentum of the robot in a more consistent manner. (iii) The proposed control algorithm enabled a conventionally built one-legged robot to exhibit agile balancing and locomotion behavior, e.g., untethered running on an inclined surface, a newly-addressed locomotion task for stiff-by-nature legged systems. In comparison to existing methods, the proposed controller constitutes a good balance between the computational complexity and locomotion control efficacy to achieve stable and compliant running locomotion for untethered legged robots.

A preliminary version of this work reporting early results was presented by Ugurlu et al. (2012). The current article is a significantly improved version to provide an archival value with the following qualities: (i) a stability proof for the compliance control to indicate the boundedness of position tracking and interplay between the control gains, (ii) elaboration of the trajectory generator and admittance controller, (iii) a comprehensive set of newly-added experimental results, e.g., one-legged running on an inclined surface, balancing within the presence of various disturbances.

The remainder of this paper is organized as follows. The mechatronic hardware of our experimental one-legged robot is succinctly introduced in Sect. 2. One-legged running trajectory generator is disclosed in Sect. 3. Section 4 reveals the details of the proposed locomotion control framework. Experimental results are presented in Sect. 5, including thorough discussions on the possible impacts, limitations and future work. Finally, the paper is concluded in Sect. 6.
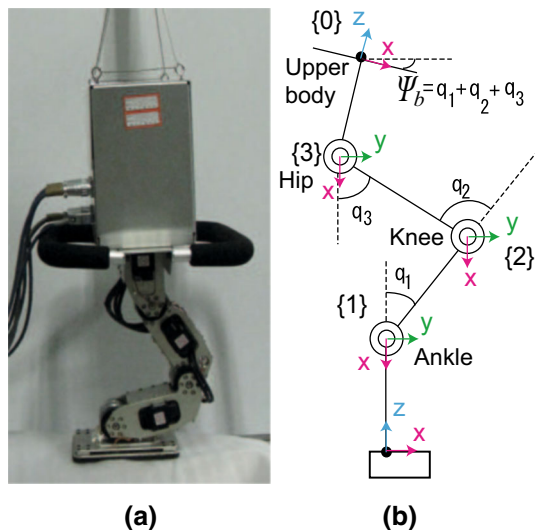
**Fig. 1** The actual robot and its joint configurations

**Table 1** Mechanical specifications of TTI-Hopper

| Size | Upper tip-Hip length | 370 mm |
|------|----------------------|--------|
| | Hip-Knee length | 15 mm |
| | Knee-Ankle length | 150 mm |
| | Ankle-sole length | 40 mm |
| | Toe-heel length | 166 mm |
| Weight | Leg | 5.46 kg |
| | Upper body | 5.37 kg |
| | Total | 10.83 kg |

## 2 One-legged runner robot: TTI-Hopper

In order to investigate running dynamics, we developed a 4-link 3-DoF (Degrees of Freedom) one-legged robot, named TTI-Hopper. It is actuated via brushless DC motors to power hip, knee and ankle joints, making its motion constrained in the sagittal plane. The actual robot can be viewed in Fig. 1, together with its joint configurations. Note that the robot is completely untethered and it can move in other planes if it is subject to disturbances. Table 1 summarizes its mechanical dimensions.

The main controller is an Intel TMD Duo 2 CPU processor (3.33 GHz, 4.0 GB RAM), executing the controllers via a MATLAB xPC target system, running at a sampling frequency of 1 kHz. The robot was equipped with an attitude reference sensor (Lord MicroStrain 3DM-GX5-25 AHRS) in the torso which was synchronized with the main controller using the method described in Olson (2010). Four force sensing units were employed and distributed to each corner of the foot (Kormushev et al. 2012; Lebosse et al. 2011). Using this force sensing topology, we can measure the z-axis GRF (Ground Reaction Force) and x-axis ZMP (Zero Moment Point), namely $F_z$ and $X_{zmp}$. Furthermore, the x-axis GRF, $F_x$, can also be estimated as below (Hyon and Cheng 2007).

$$F_z = \sum_{i=1}^{4} f_i; \ X_{zmp} = \frac{\sum_{i=1}^{4} f_i r_{ix}}{F_z}; \ F_x = \frac{(x - X_{zmp})F_z}{z}$$
(1)

In above equations, $f_i$ is the $i^{th}$ force value ($i = 1, 2, 3, 4$); $r_{ix}$ refers to x-axis coordinate of the $i^{th}$ sensor with respect to the FSC (Foot Sole Center). y-axis ZMP can also be com-

puted; however, we are not able to make use of it since the robot cannot move through y-axis, as stated above.

## 3 Running trajectory generation

Running trajectory is composed of two components that have distinct characteristics: (i) CoM trajectory during support phase, (ii) CoM and feet trajectories during flight phases. For each case, x-axis (horizontal) and z-axis (vertical) trajectories are separately planned in a way to meet corresponding conditions, e.g., dynamic balancing, jumping, feasible landing. Once the trajectory is generated in Cartesian space, we use an analytical inverse kinematics algorithm to obtain joint angle references.

### 3.1 Support Phase Trajectory

#### 3.1.1 Vertical CoM trajectory generation

Throughout the support phase, we observe three main events: (i) touch-down (td), bottom (b), and lift-off (lo); see Fig. 2. Touch-down is the initial instant whereas lift-off stands for the terminal instant. In order to realize a successful jump, the z-axis CoM acceleration must reach the gravitational acceleration, $g$, at the lift-off. Moreover, the z-axis CoM velocity at this instant determines the flight time period; therefore, we need a certain lift-off velocity which can be calculated by considering the planned flight time period. Hence, it is possible to determine two of the boundary conditions: lift-off velocity and acceleration, namely, $\dot{z}_{lo}$ and $\ddot{z}_{lo}$.

Due to projectile motion dynamics, the z-axis CoM trajectory follows a parabola during the flight phase. Because of this fact, the z-axis CoM acceleration is equal to $g$ at the touch-down. By the same token, the touch-down velocity is equal to lift-off velocity in amplitude but opposite in direction. This fact allows us to determine two of the boundary conditions: touch-down velocity and acceleration, namely, $\dot{z}_{td}$, $\ddot{z}_{td}$.

To obtain a symmetrical pattern, touch-down and lift-off positions are kept the same for the z-axis: $z_{td} = z_{lo} = z_a$; see
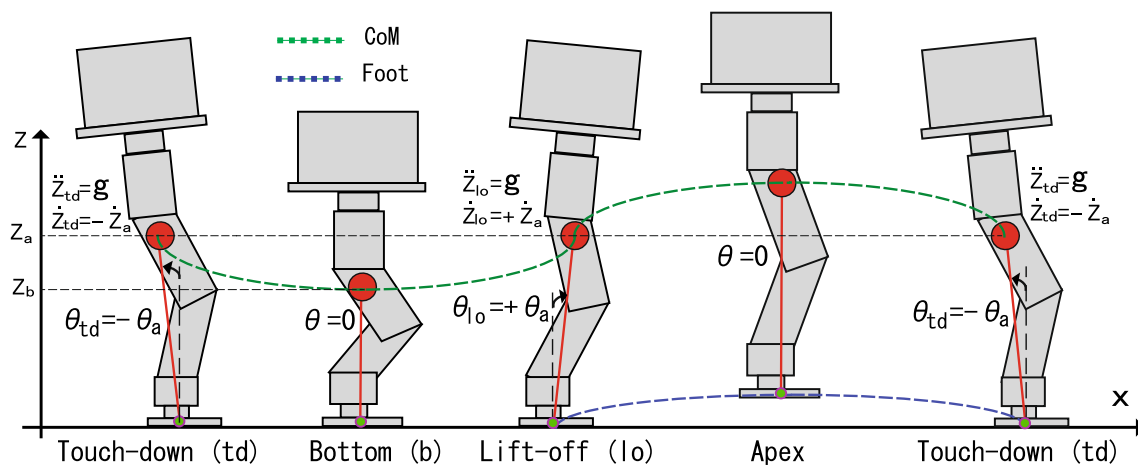
**Fig. 2** The main instants during a complete jumping cycle. The support phase is defined between touch-down and lift-off, and the flight phase is defined between lift-off and touch-down

Fig. 2. We also assign a position at the bottom, $z_b$, to ensure a smooth trajectory since the touch-down and lift-off velocities are equal in magnitude but have opposite directions. One can assign these position values by considering the joint angle limits and kinematic structure.

In light of these facts, we can assign 7 boundary conditions, namely, z-axis CoM position, velocity, acceleration at the lift-off ($z_{lo}$, $\dot{z}_{lo}$, $\ddot{z}_{lo}$), z-axis position at the bottom ($z_b$), z-axis position, velocity, acceleration at the touch-down ($z_{td}$, $\dot{z}_{td}$, $\ddot{z}_{td}$). With these 7 constraints in mind, we utilized $6^{th}$ order polynomials to synthesize smooth and time-varying z-axis CoM trajectoies to ensure jumping conditions.

At the beginning of an experiment, the initial condition is the bottom instant at which initial velocity and acceleration are zero. Similarly, the robot can be stopped after exhibiting a number of jumps by assigning the bottom instant as the terminal condition with zero velocity and acceleration.

### 3.1.2 Horizontal CoM trajectory generation

To meet the dynamic balance criterion, the ZMP concept is utilized in x-axis trajectory generation (Vukobratovic and Borovac 2004). Since TTI-Hopper is a planar robot, its motion is constrained in x-z plane and only the x-axis ZMP needs to be analyzed (Tajima et al. 2009).

$$mX_{zmp}(\ddot{z} + g) = mx(\ddot{z} + g) - mz\ddot{x} - \dot{L}_y \qquad (2)$$

In (2), $x$ and $z$ are the x-axis and z-axis components of the CoM trajectory, $m$ is the total mass, $X_{zmp}$ is the x-axis ZMP, and $L_y$ is the pitch axis angular momentum that is associated with the rotational inertia. In short, the term $\dot{L}_y$ refers to rotational motion while the other terms arise due to inertial forces.

While defining the ZMP, the conventional approach is to make use of an inverted pendulum model with a point mass and a massless telescopic leg (Harada et al. 2004). Despite its practical value, it cannot characterize rotational inertia; therefore, the term $\dot{L}_y$ is often omitted. However, angular momentum is an important criteria for the balancing and its incorporation is of importance (Orin et al. 2013). To this end, researchers propose different pendulum models that can encapsulate rotational inertia while avoiding the complexity of multi-DoF approaches (Pratt et al. 2006; Ugurlu et al. 2016; Sanyal and Goswami 2013; Ugurlu and Kawamura 2012).

Among these approaches, we utilized Eulerian ZMP Resolution (EZR) method, which assumes an equivalent pendulum model with a telescopic leg and a single rigid body located at the pendulum tip (Ugurlu and Kawamura 2012); see Fig. 3. This pendulum is called EZR pendulum. Unlike the usual physical bodies, the rigid body in EZR pendulum has a varying rotational inertia. Using the composite rigid body algorithm, we can calculate the rotational inertia of the actual robot at each control cycle, then assign it as the rotational inertia of EZR the pendulum (Balafoutis and Patel 1991). This allows us to mimic the real robot in terms of rotational inertia characterization, an important feature that characterizes multi-DoF behavior.

In EZR method, robot CoM is assumed to be oriented in an identical manner to EZR pendulum model. With this in mind, $L_y$ and $\dot{L}_y$ can be calculated as follows,

$$L_y = I_y \omega_y = I_y \dot{\theta}, \qquad (3)$$
$$\dot{L}_y = I_y \ddot{\theta}_y + \dot{I}_y \dot{\theta}, \qquad (4)$$

where $I_y$ is the diagonal element of the rotational inertia associated with y-axis and $\theta$ is EZR pendulum orientation; see Fig. 3. Since $I_y$ varies with respect to joint configuration, its
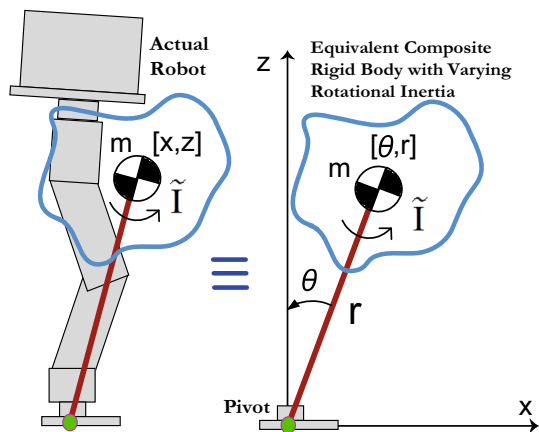
**Fig. 3** Implementation of Eulerian ZMP Resolution method to TTI-Hopper. In this model, the robot is considered to be a single rigid body with a varying rotational inertia. The robot CoM is considered to be oriented identically with respect to this rigid body. Its rotational inertia is updated in each control sample
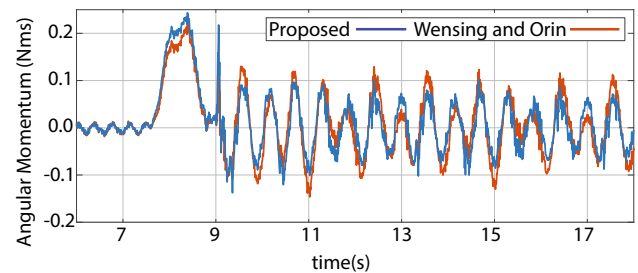


**Fig. 4** Angular momentum is computed for a simultaneous squatting-and-swaying motion using EZR and centroidal angular momentum computation algorithm (Wensing and Orin 2015)

time derivative is nonzero and needs to be computed in each cycle. Recall that TTI-Hopper is a planar robot; therefore, the Euler's motion equations are boiled down to (3).

To express the ZMP equation as a whole, we utilize the polar coordinates $(r, \theta)$ as depicted in Fig. 3.

$$x = r \sin \theta \tag{5}$$
$$z = r \cos \theta \tag{6}$$

As explained in previous subsection, defining the z-axis CoM trajectory is of importance to ensure jumping conditions. Hence, we find it useful to employ the coordinates $(z, \theta)$. With this objective at hand, $x$ and $\ddot{x}$ can be expressed as follows.

$$x = z \tan \theta \tag{7}$$
$$\ddot{x} = \ddot{z} \tan \theta + (1 + \tan^2 \theta)(z\ddot{\theta} + 2\dot{\theta}(\dot{z} + z\dot{\theta} \tan \theta)) \tag{8}$$

Inserting (4), (7) and (8) to (2), the following motion equation is obtained.

$$mX_{zmp}(\ddot{z} + g) = mgz \tan \theta - \ddot{\theta}(I_y + D_a)$$
$$- \dot{\theta}(\dot{I}_y + D_b) - 2\dot{\theta}^2 D_a \tan \theta \tag{9}$$

In (9), the following parameters are used to simplify the computation.

$$D_a = mz^2 \sec^2 \theta; \tag{10}$$
$$D_b = 2mz\dot{z} \sec^2 \theta \tag{11}$$

One can arrange (9) in a way to yield $\ddot{\theta}$ as below.

$$\ddot{\theta} = \frac{mgz \tan \theta - mX_{zmp}(\ddot{z} + g)}{I_y + D_a}$$
$$- \frac{\dot{\theta}(\dot{I}_y + D_b) - 2\dot{\theta}^2 D_a \tan \theta}{I_y + D_a} \tag{12}$$

Equation (12) can be solved numerically to synthesize $\theta$ trajectory in case the following parameters are provided.

- Initial conditions: $\theta_{td}$ and $\dot{\theta}_{td}$; see Sect. 3.2.
- Support time and flight time periods: $T_s$ and $T_f$.
- Vertical CoM trajectory: since $z$ is synthesized via polynomials, $z$, $\dot{z}$, and $\ddot{z}$ are readily available.
- Rotational inertia: $I_y$ and $\dot{I}_y$ are computed at each sampling period (Balafoutis and Patel 1991).
- Reference x-axis ZMP: $X_{zmp}$.

Having computed $\theta$ trajectory, one can use (7) to obtain x-axis CoM trajectory. This procedure allow us to utilize ZMP-based trajectory generation in which the rotational inertia and corresponding angular momentum rate change are characterized.

To validate that the proposed trajectory generator can characterize angular momentum accurately, we computed angular momentum using (3) and experimentally compared it with the centroidal angular momentum algorithm proposed by Wensing and Orin (2015). For this comparison study, the robot squatted with an amplitude of 28 cm and a frequency of 1 Hz while additionally swaying back and forth. The results are displayed in Fig. 4, where dotted red and solid blue lines show angular momentum variations for the proposed method and centroidal angular momentum computation, respectively.

As may be observed, both methods output very similar variations, and therefore, we conclude that the proposed method can characterize the angular momentum of a multi-DoF system adequately. Considering the centroidal angular momentum algorithm as the ground truth, the root mean square of the approximation error was less than 0.02 Nms. Furthermore, the computation time was decreased by a factor of nearly 14.

## 3.2 Flight Phase Trajectory

During a flight phase, the robot is subject to projectile motion dynamics. It has no contact with the environment and therefore the ZMP is undefined. See Fig. 2 for an illustration of a flight phase.

### 3.2.1 Vertical CoM trajectory

The CoM simply follows a parabola formulated below,

$$z = z_a + \dot{z}_a t - \frac{1}{2} g t^2, \tag{13}$$

where $z_a$ and $\dot{z}_a$ are lift-off position and velocity, amd $t$ denotes time. To preserve the symmetry, the lift-off position and velocity are assigned as $z_{lo} = z_a$ and $\dot{z}_{lo} = \dot{z}_a$; see Fig. 2. One can also generate vertical foot trajectory $z_f$ to retract the leg while the robot is in the air; yet, we did not consider it in this study.

### 3.2.2 Horizontal CoM and foot trajectories

When the robot is in the air, the translational momentum through x-axis is conserved. Because this fact, x-axis CoM velocity remains constant and it is equal to horizontal lift-off velocity. With this piece of information, x-axis CoM trajectory can be calculated as in the following.

$$\dot{x} = \dot{z}_a \tan \theta_a + z_a \dot{\theta}_a (1 + \tan^2 \theta_a) \tag{14}$$

$$x = z_a \tan \theta_a + \dot{x} t \tag{15}$$

Recall that $\theta$ angle at the lift-off is equal to $\theta_a$; see Fig. 2. From the same figure, one can observe that $\theta$ angle at the touch-down is set as $\theta_{td} = -\theta_a$, for purpose of symmetrical motion planning. At the apex, i.e., in the middle of flight phase, $\theta$ angle is zero. Using these 3 boundary conditions, one can generate a $\theta$ angle trajectory via a parabola. Since the instant of touch-down is the initial condition for the next support phase, we set $\theta_{td}$ and $\dot{\theta}_{td}$ values as initial conditions to solve the differential equation given in (12).

In order to attain a successful landing, the foot must be progressively translated beyond the CoM; see Fig. 2. Using the previously generated $x$, $z$, $z_f$ and $\theta$ parameters, we can synthesize x-axis foot trajectory.

$$x_f = x - (z - z_f) \tan \theta \tag{16}$$

## 4 Admittance control

As disclosed in Sect. 2, the joints of TTI-Hopper are actuated via brushless DC motors with 1:100 gear ratio strain wave
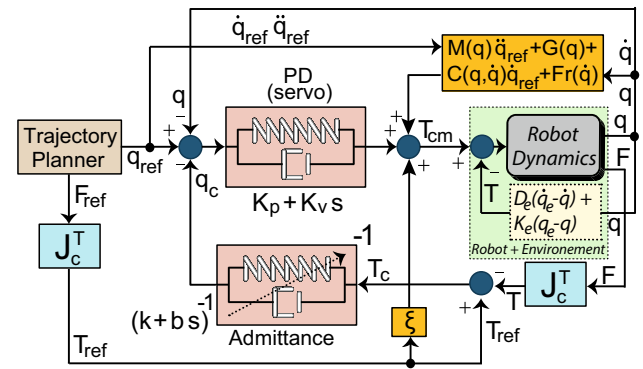


**Fig. 5** An admittance control scheme was implemented for managing the position-force control trade-off. The robot-environment interaction is demonstrated at the joint level

gears. This conventional approach enables us to avoid backlash problem; yet, the high output mechanical impedance impedes the system's ability to cope with force constraints that may arise due environmental interactions, e.g., touchdown impacts. With this limitation in mind, we implemented admittance control to handle physical interactions with the environment.

Figure 5 displays the implemented controller, where $K_p$, $K_v$ and $k$, $b$ are diagonal matrices that store PD control and admittance control gains. $M(q)$, $C(q, \dot{q})$, $G(q)$ and $Fr(\dot{q})$ denote inertia, Coriolis and centrifugal, gravity and joint friction terms, respectively. $F$ and $q$ parameters stand for interaction force at the foot and joint displacement vectors. The robot-environment interaction is demonstrated at the joint level: $q_e$, $D_e$ and $K_e$ stand for environmental displacement, damping, and stiffness, respectively. Note that the measured force reflects on the robot dynamics by means of environmental interaction.

The underscript $_{ref}$ refers to reference values, while measured quantities have no underscript. $J_c$ is the Jacobian matrix defined between the CoM and foot center, $T_{cm}$ is the command torque, $q_c$ is the output of the admittance controller, $T_c$ is the torque that corresponds to measured force error $F$, $T_{ref}$ is the torque that corresponds to reference force $F_{ref}$, $T$ is the actual torque that corresponds to measured contact force $F$, $s$ is the Laplace variable.

The rationale behind the synthesis of such a controller is the lack of torque-controllability at robot joints, since the robot is *stiff-by-nature* due to use of conventional actuators with high mechanical output impedance. However, position control can be attained with the help of high gain position control loop. Moreover, the controller bandwidth can be further extended with the help of dynamics and friction compensation. When the robot is physically interacting with the environment, e.g., sudden contact changes or touch-down impacts, the force error increases. In this case, the admittance controller generates $q_c$ which is the angular displacement that
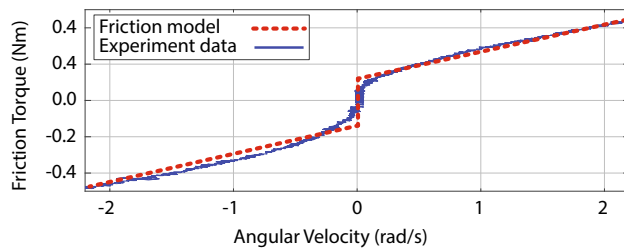
**Fig. 6** Experimental friction data vs. identified friction model

correspond to force error. As a secondary feedback via $q_c$, we construct a force loop which increases the compliance to contend with force errors.

In other words, the joint reference $q_{ref}$ is updated for about $q_c$ radians to diminish force error; $q_{ref} := q_{ref} - q_c$. When there is no force error, e.g., when the robot is in flight phase or standing still, $q_c$ becomes zero and position control is automatically prioritized. This way, the compliance is actively increased only when necessary, in particular, to handle physical interactions.

Joint angle reference $q_{ref}$ is generated by the algorithm disclosed in Sect. 3. Force reference during support phase $F_{ref}$ is calculated using the reference values (Hyon and Cheng 2007). The corresponding joint force references are then calculated using the CoM Jacobian.

$$F_{zref} = m(\ddot{z}_{ref} + g) \tag{17}$$

$$F_{xref} = (x_{ref} - X_{zmpref})\frac{F_{zref}}{z_{ref}} \tag{18}$$

$$T_{ref} = J_c^\mathsf{T} \begin{bmatrix} F_{xref} \\ F_{zref} \end{bmatrix} \tag{19}$$

### 4.1 Friction and dynamics compensation

Using an inverse dynamic algorithm (Mistry et al. 2010), we compute the necessary torques to implement computed torque control; see Fig. 5. Furthermore, friction characteristics of each joint are identified using the disturbance observer-based method reported in Ugurlu et al. (2015). As an example, friction characteristics of hip joint is depicted in Fig. 6 where solid blue and dotted red lines indicate the experimental data and identified model, respectively. Having identified friction characteristics for each joint separately, friction compensation torques are added to the system as illustrated in Fig. 5.

### 4.2 Inequality constraints

The generated torque command is subject to the following actuator limitations.

$$T_- < T_{cmd} < T_+; \ q_- < q_{ref} < q_+; \ \dot{q}_- < \dot{q}_{ref} < \dot{q}_+ \tag{20}$$

In (20), the subscripts $-$ and $+$ denote minimum and maximum actuator limits. We set the trajectory parameters and controller gains in a way that (20) is not violated. Furthermore, when the vertical force is greater than a threshold, the robot is in contact with the ground: $F_z > \epsilon$. In this case the following inequalities must be satisfied to avoid slippage:

$$\sqrt{F_x^2 + F_y^2} < \mu_s F_z; \ M_z < \mu_s F_z \tag{21}$$

where $F_x$, $F_y$ and $F_z$ are contact forces along the x, y, and z axes, $M_z$ is the induced yaw moment, $\mu_s$ is the static friction constant between the foot and the environment. In the case of TTI-Hopper, we employed custom-made rubber pads such that $\mu_s$ is sufficiently large to satisfy this condition.

### 4.3 Stability

The robot dynamics can be expressed as follows:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + Fr(\dot{q}) = T_{cm} + T, \tag{22}$$

where measured torque due to environmental interaction is $T = -D_e(\dot{q}) + K_e(q_e - q)$. Here, it is assumed that the environment is stationary, i.e., $\dot{q}_e = 0$. The command torque $T_{cm}$ is designed as below:

$$T_{cm} = M(q)\ddot{q}_{ref} + C(q,\dot{q})\dot{q}_{ref} + G(q) + Fr(\dot{q}) + \xi T_{ref}$$
$$+ K_v(\dot{q}_{ref} - \dot{q}_c - \dot{q}) + K_p(q_{ref} - q_c - q). \tag{23}$$

In (23), $\xi$ is a diagonal matrix whose elements are defined as $\xi_{[i,i]} = \frac{K_{vi}}{b_i}$; the underscript $i$ indicates the joint number ($i = 1, 2, 3$). The admittance controller output $q_c$ is related to actual torque via $T_c = T_{ref} - T = kq_c + b\dot{q}_c$. In this condition, the Lyapunov candidate is determined as in the following.

$$V = \frac{1}{2}\left(\dot{\tilde{q}}^\mathsf{T} M(q)\dot{\tilde{q}} + \tilde{q}^\mathsf{T} K_p\tilde{q} + q^\mathsf{T}\beta K_e q + \hat{q}^\mathsf{T} P\hat{q}\right), \tag{24}$$

in which $\tilde{q} = q - q_{ref}$, $\dot{\tilde{q}} = \dot{q} - \dot{q}_{ref}$, and $\hat{q} = [q \ q_e]^\mathsf{T}$; $(\cdot)^\mathsf{T}$ indicates matrix transpose. $\beta$ is a diagonal matrix which is defined as $\beta = I + \xi$ and $I$ is a unit matrix. Furthermore, $P$ is defined as the following matrix.

$$P = \begin{bmatrix} 2A & -\beta K_e \\ -K_e^\mathsf{T}\beta^\mathsf{T} & 2C \end{bmatrix} \tag{25}$$

In (25), $A$ is a skew-symmetric matrix and $C$ is a positive semi-definite matrix. Both $A$ and $C$ are free parameters that

we can assign without any restriction. The time derivative of $V$ along the trajectories is yielded as in the following:

$$\dot{V} = -\dot{\tilde{q}}^{\mathsf{T}} K_v \dot{\tilde{q}} - \dot{q}^{\mathsf{T}} \beta D_e \dot{q} - \dot{q}_{ref}^{\mathsf{T}} \beta T - \dot{\tilde{q}}^{\mathsf{T}} \gamma q_c, \qquad (26)$$

where $\gamma$ is a diagonal matrix that defined as $\gamma = K_p - k\xi$. If the controller gains are tuned in a way so as to ensure $\frac{K_{vi}}{b_i} = \frac{K_{pi}}{k_i}$ for each joint, Eq. (26) takes the following form:

$$\dot{V} = -\dot{\tilde{q}}^{\mathsf{T}} K_v \dot{\tilde{q}} - \dot{q}^{\mathsf{T}} \beta D_e \dot{q} - \dot{q}_{ref}^{\mathsf{T}} \beta T. \qquad (27)$$

Due to the nature of running motion, $T$ remains bounded, and $\dot{q}_{ref}$ is bounded by design. Thus, $\dot{q}_{ref}^{\mathsf{T}} \beta T$ is bounded. The derivative of the Lyapunov function is negative definite when the following inequality is satisfied:

$$\lambda_{mn}(K_v) \left\| \dot{\tilde{q}} \right\|^2 + \lambda_{mn}(\beta D_e) \|\dot{q}\|^2 > \lambda_{mx}(\beta) \left\| \dot{q}_{ref} \right\| \|T\| \qquad (28)$$

In (28), $\lambda_{mn}(\cdot)$ and $\lambda_{mx}(\cdot)$ indicate the minimum and maximum eigenvalues of a matrix, respectively. Based on this results, we conclude that the controller is asymptotically stable when there is no contact motion, i.e., $T = 0$, because $V > 0$ and $\dot{V} = -\dot{\tilde{q}}^{\mathsf{T}} K_v \dot{\tilde{q}} < 0$.

The controller is uniformly ultimately stable when there is contact motion as $V$ and $\dot{V}$ take the following forms.

$$V = \frac{1}{2} \left( \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} K_p & 0 \\ 0 & M(q) \end{bmatrix} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} + \begin{bmatrix} q \\ \hat{q} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \beta K_e & 0 \\ 0 & P \end{bmatrix} \begin{bmatrix} q \\ \hat{q} \end{bmatrix} \right) \qquad (29)$$

$$\dot{V} = -\begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} 0 & 0 \\ 0 & K_v \end{bmatrix} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} - \begin{bmatrix} q \\ \dot{q} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} 0 & 0 \\ 0 & \beta D_e \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} - \dot{q}_{ref}^{\mathsf{T}} \beta T \qquad (30)$$

In the light of this analysis, it is observed that both position and velocity states $(q, \dot{q})$ and position and velocity errors $(\tilde{q}, \dot{\tilde{q}})$ are bounded. To improve the stability, $K_v$ gains may be increased so that the convergence rate of the error gets faster and the radius of the bounded error gets smaller.

## 5 Experiment results

In order to test the viability of the proposed locomotion controller, we conducted a series of experiments using TTI-Hopper. These experiments can be listed as follows.

– Experiment #1: TTI-Hopper was placed on a cart. The cart was pushed back and forth within a certain range and the frequency of this movement was synchronized via a metronome; see Fig. 7.
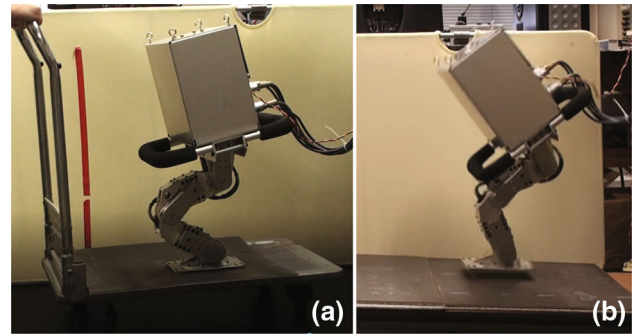


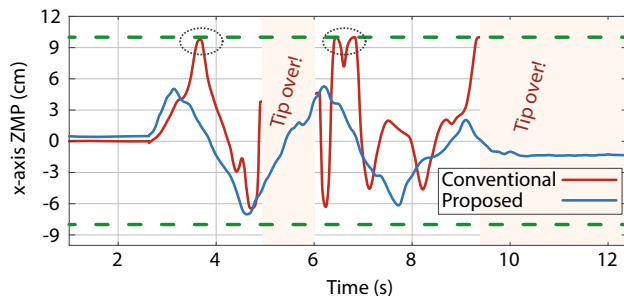**Fig. 7** **a** Balancing on a moving cart experiment. **b** Running on an inclined surface

– Experiment #2: TTI-Hopper performed a forward running motion with 8 consecutive jumps on a flat surface. This experiment was conducted twice; with and without the proposed admittance controller.
– Experiment #3: TTI-Hopper performed a brief forward running motion 10 consecutive jumps on a flat surface. This experiment was conducted twice, with and without the proposed trajectory generation algorithm EZR. In both experiments, the proposed admittance controller was used.
– Experiment #4: TTI-Hopper performed a forward running motion on an inclined surface with a slope of approximately 9 degrees. The running motion included 13 consecutive jumps; see Fig. 7.

In all experiments, PD gains and admittance controller gains were set to constant values tabulated in Table 2. The controller gains were experimentally set in accordance with the standard rules; stability, tracking performance, and non-excitation of unmodeled nonlinear dynamics. The sampling frequency was 1 kHz. Running locomotion parameters are assigned as follow: stance and flight time windows were 0.2 and 0.18 s, forward displacement was 1.5 cm and foot clearance was 2.5 cm. Inspired by the natural center of pressure variation in running locomotion (Cavanagh 1987), the reference x-axis ZMP was defined using a linear function between the touch-down and lift-off instants. The initial and final values of this reference is $-3$ and $3$ cm, respectively. These values are assigned by leveraging the forward velocity vs. dynamic balance trade-off.

Note that there is no upper limit for consecutive jumps while TTI-Hopper runs forward. It is kept limited for a clearer view. Reader can refer to the multimedia attachment to view further scenes from experimental studies, e.g., running on a flat surface, running on an inclined surface, balancing on a moving cart, balancing on a platform which is abruptly dropped.

**Table 2** PD servo and admittance controller gains

|  | $K_p$ | $K_v$ | $k$ | $b$ |
|---|---|---|---|---|
| Hip | 50 | 6 | 5.5 | 0.66 |
| Knee | 35 | 5 | 3.85 | 0.55 |
| Ankle | 20 | 4.8 | 2.2 | 0.53 |



**Fig. 8** x-Axis ZMP responses when the robot was instructed to keep its balance on a moving cart, experiment #1

## 5.1 Experiment #1: balancing on a moving cart

To demonstrate balancing abilities within the presence of continuous disturbances, TTI-Hopper was placed on a moving cart which was pushed back and forth with a certain range; see Fig. 7a. The cart movement period was approximately 3 seconds and this was attained via a metronome. The robot was instructed to maintain its balance.

In Fig. 8, x-axis ZMP responses are depicted as the main result. The solid blue and red lines stand for ZMP responses for the proposed admittance controller and for the conventional controller (Napoleon et al. 2005), respectively. Support polygon boundaries are indicated via dashed green lines. In case the conventional controller was used, the measured ZMP often reached the support polygon boundaries. That led to situations where the robot could not properly maintain its balance and tip over, see shaded pink areas. In contrary, the proposed admittance controller enabled the robot maintain its balance despite the applied continuous disturbance. In conclusion, it is observed that the active compliance property that is introduced by the proposed controller is superior to the conventional ZMP feedback in terms disturbance attenuation.

## 5.2 Experiment #2: one-legged running on a flat surface

The forward running capability of TTI-Hopper was tested in this experiment, in which the robot was instructed to perform 8 consecutive forward jumps. Similar to the case in experiment #1, the experiment was conducted twice: (a) with the proposed admittance controller, (b) with the conventional

controller (Napoleon et al. 2005). The proposed trajectory generation algorithm was used in both cases.

The results are presented in Fig. 9, in which the solid blue and red lines indicate variations from the proposed and conventional controllers, respectively. Furthermore, the robot could not maintain its balance and tipped over after the 3rd touch-down instant; therefore, the corresponding data are not displayed after that moment.

Vertical GRF variations can be observed in Fig. 9a, in which zero vertical GRF levels are indicative of successful flight phases. In this figure, the reference force is indicated via a dashed green line. In case the proposed controller was used, TTI-Hopper was able to perform 8 consecutive forward jumps. At the touch down-instants, GRF peaked up to 200 N. In the case of conventional controller, the GRF peaks seemed less; however, this was due to improper foot landing as one or two sensors could not have a contact with ground.

The CoM height variation was computed using on board sensors and projectile dynamics and it is plotted in Fig. 9b. This result suggests that the CoM height was kept around 27.2 cm with ±0.1 cm variation throughout the entire whole locomotion period.

ZMP measurement is depicted in Fig. 9c, where the upper and lower boundaries were shown using dashed green lines. The data clearly shows that the ZMP reaches to the upper boundary following the 3rd touch-down instant in case the conventional controller was used. When the proposed controller was used, the ZMP varied between 0 and 3.1 cm and always stayed within the boundaries, adequately demonstrating the fact that the robot maintained its balance during running. Note that the ZMP is undefined when the robot is in the air, thus, such portions of data are indicated with yellow rectangles.

Another useful property of the proposed controller is the ability to increase the compliance so as to handle ground impacts. When the robot is not able to do so, the impact is transmitted to the upper body and it may lead to undesired torso movements. This could be observed in Fig. 9d; the robot torso showed large movements when the conventional controller was used. Torso rate was measured to be up to 17 deg/s. When the proposed controller was used, the robot exhibited a compliant behavior, and as a result, the torso movement was well contained. It varied between 0 and 3 deg/s.

As previously explained, the joint angle reference trajectories are updated via the secondary feedback signal $q_c$, so as to contend with contact forces. Indeed, this property is more prominent during touch-down and the subsequent stance phases. In Fig. 10, reference hip angle trajectory ($q_{ref}$, dotted green), measured hip angle ($q$, solid red) and the updated hip angle trajectory ($q_{ref} - q_c$, solid blue) are plotted. Indeed, when there is no contact force, i.e., when the robot is in the flight phase, $q_c$ converges to zero, therefore, $q$ tracks $q_{ref}$. When the robot is in contact with the ground and force error
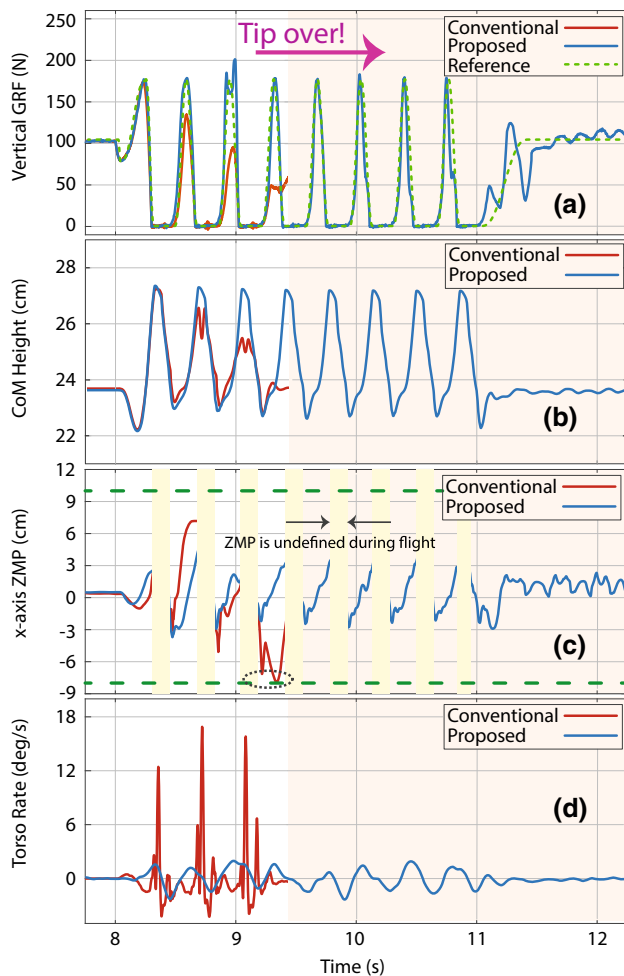
**Fig. 9** Experiment #2 results: **a** GRF variations, **b** x-axis ZMP, **c** torso angle rates
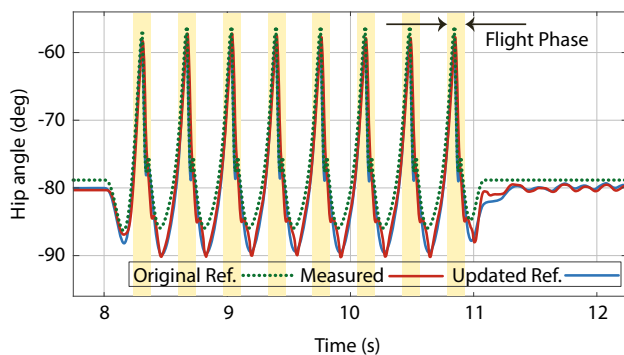


**Fig. 10** Hip angle trajectories, experiment #2. The admittance controller updates the reference trajectory in a way so as to contend with contact force errors. The updated reference is $q_{ref} := q_{ref} - q_c$ (Color figure online)

increases due to touch-down, the $q_c$ signal updates the original reference ($q_{ref} := q_{ref} - q_c$) so that the robot can handle
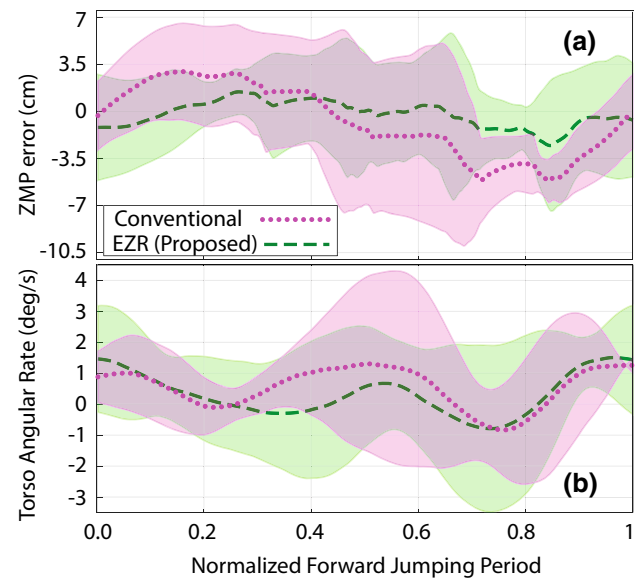


**Fig. 11** Experiment #3 results. Mean $\pm$ SD for ZMP error and torso angle rate for 10 consecutive jumps

the contact in compliant manner to achieve balancing running locomotion. This time, $q$ tracks the updated reference

### 5.3 Experiment #3: one-legged running with and without the proposed trajectory generator: EZR

To observe the effect of the proposed trajectory generator EZR, we ran one-legged running experiments on a flat surface in which the robot continuously jumped forward for 10 times. The experiment was conducted twice: (i) using the proposed trajectory generator EZR, and (ii) using a conventional ZMP-based trajectory generation (Kagami et al. 2002). In both cases, the proposed admittance controller was employed.

In both cases, the robot successfully completed the running motion. Therefore, we conclude that the proposed admittance controller is the essential element in ensuring stable and continuous running motion. To investigate further, mean $\pm$ SD (standard deviation) values of ZMP error and torso angle rate were plotted for 10 consecutive forward jumps; see Fig. 11. In this figure, purple and green areas indicate variations for the conventional and proposed EZR methods, respectively. Although the mean values varied similarly, the variation of ZMP error and torso angle rate was more constrained when the proposed trajectory generator EZR was used. We argue that this is due to the fact that EZR is able to characterize rotational inertia and the associated angular momentum during trajectory generation, an important capability in postural balancing. Therefore, it led to more dynamically consistent trajectories.
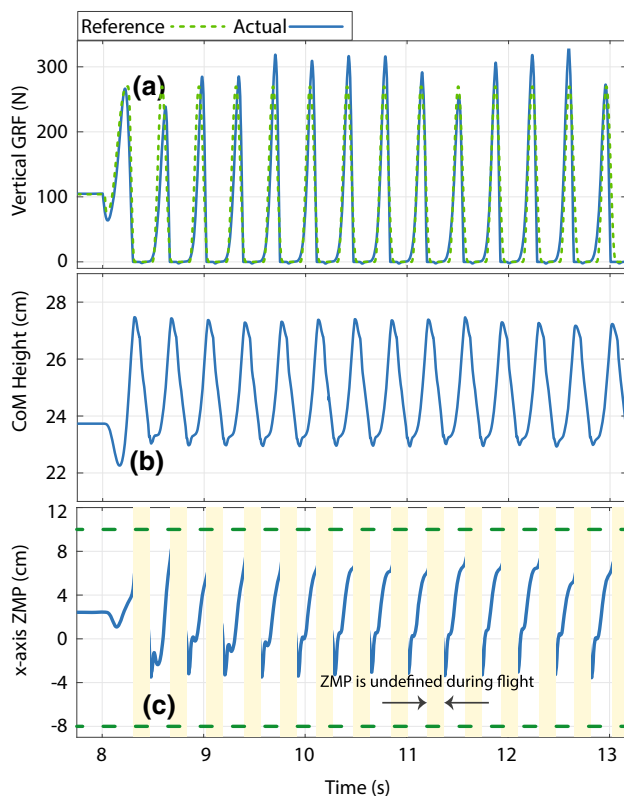
**Fig. 12** GRF response and x-axis ZMP variations, experiment #4

### 5.4 Experiment #4: one-legged running on an inclined surface

To challenge the forward running capability of TTI-Hopper, it was placed on an inclined surface with a slope of 9 degrees; see Fig. 7b. Torso angle was bent forward so as to ensure the static balance before it started running. All the other running parameters were kept the same as in the case of experiment #2. The robot performed 13 consecutive forward jumps. The results are displayed in Fig. 12.

Ground reaction force response and reference are plotted in Fig. 12a. As may be observed, zero GRF levels indicate successful foot clearances. Moreover, GRF peaks showed a variance between 250-320 N, causing a force error. This is due to the challenging nature of the experiment, as the environment information was not taken into account. It was the active compliance property of the proposed controller to ensure balanced running locomotion. Therefore, the uncertainty that was introduced to the system in this experiment caused such a variation.

The CoM height is plotted in Fig. 12b, in a similar fashion to that of Fig. 9b. The CoM height was observed to vary around 27.2 cm with ±0.1 cm variation throughout the entire whole locomotion period.

ZMP measurement is depicted in Fig. 12c, where dashed green lines indicate the upper and lower limits of the error.

ZMP response is undefined when there is no contact, therefore, the data is during the flight phase is not plotted; see yellow rectangular areas. The ZMP always remained within the support polygon and this proves that the robot was able to perform running on an inclined surface while maintaining its dynamic balance. Yet, the ZMP seemed to be closer to the upper bound towards. This is due to the fact that the torso was bent forward to keep the static balance on the inclined surface.

## 6 Concluding remarks

In this paper, we proposed an admittance controller and a trajectory generator to achieve stable running locomotion using an untethered one-legged robot with conventional actuators. Thanks to the active compliance provided by the proposed controller, the robot is able to handle contact forces, especially at the moment of touch-down. Trajectory generator EZR is able to characterize angular momentum, therefore, it provides more dynamically consistent CoM trajectories when compared to conventional trajectory planners.

The proposed controller manages the position-force trade-off by means of admittance gains; the position tracking is compromised when there is only force error due to contact. This property enabled us to successfully demonstrate balancing under various disturbances and running locomotion on flat and inclined surfaces. Furthermore, the stability of the balance controller was mathematically investigated in the sense of Lyapunov and the inter-relation between the PD and admittance controller gains were demonstrated.

The proposed methodology was tested using an untethered one-legged robot, powered via conventional actuators. The whole algorithm was executed under 1 kHz sampling without any computational overload, it is computationally inexpensive. Therefore, the proposed controller may be applied as a mid-level trajectory controller for a general class of legged robots that aim to handle contacts compliantly.

The studies in model predictive control and trajectory optimization showed that the legged locomotion control can be interpreted by means of an optimization problem such that trajectory parameters and control gains can be optimally assigned (Neunert et al. 2016). Using a similar approach, it is possible to synthesize locomotion controllers by considering the actuator limitations and friction cone (Orsolino et al. 2020). With the help of the state-of-the-art computers, these algorithms can be executed in real-time in the order of 2.5–10 ms. In our next work, we will develop an optimization algorithm which can be included within a high-level controller on top of the proposed method, so that controller gains and trajectory parameters can be tuned for improved performance.

Another improvement can be made concerning the Cartesian-space tracking control. Although joint-space tracking is attained well, a Cartesian-space controller may be necessary to ensure the targer jumping height and length constraints. To this end, state estimators can be integrated and their output may be assessed within a high-level controller to ensure Cartesian trajectory regulation (Flayols et al. 2017; Bloesch et al. 2012; Hartley et al. 2020).

## Appendix: Derivation of the Lyapunov function

As previously explained, the admittance controller was expressed as $T_{ref} - T = kq_c + b\dot{q}_c$. This expression can be rearranged as follows:

$$\dot{q}_c = \mu(T_{ref} - T) - \nu q_c \tag{31}$$

$$K_v \dot{q}_c + K_p q_c = \xi T_{ref} - \xi T - (k\xi - K_p)q_c \tag{32}$$

In (31), $\mu$ and $\nu$ are diagonal matrices that are defined as $\mu_{[i,i]} = \frac{1}{b_i}$ and $\nu_{[i,i]} = \frac{k_i}{b_i}$. Using the torque command signal given in (23), the dynamics of the system may be reduced to the following form:

$$M(q)\ddot{\tilde{q}} = -C(q,\dot{q})\dot{\tilde{q}} - K_v \dot{\tilde{q}} - K_p \tilde{q} - K_v \dot{q}_c - K_p q_c + \xi T_{ref} + T, \tag{33}$$

where $\tilde{q} = q - q_{ref}$. Combining (32) with (33) yields the following:

$$M(q)\ddot{\tilde{q}} = -C(q,\dot{q})\dot{\tilde{q}} - K_v \dot{\tilde{q}} - K_p \tilde{q} + \beta T - \gamma q_c. \tag{34}$$

In (34), $\beta = I + \xi$ and $\gamma = K_p - k\xi$. Recall that $T = -D_e\dot{q} - K_e(q_e - q)$, due to environmental interaction. If we utilize this information, (34) can take the final form stated below.

$$M(q)\ddot{\tilde{q}} = -C(q,\dot{q})\dot{\tilde{q}} - K_v \dot{\tilde{q}} - K_p \tilde{q} - \beta D_e \dot{q} - \beta K_e(q_e - q) - \gamma q_c. \tag{35}$$

First, we use the following Lyapunov function candidate:

$$V_1 = \frac{1}{2}\dot{\tilde{q}}^\mathsf{T} M(q)\dot{\tilde{q}} + \frac{1}{2}\tilde{q}^\mathsf{T} K_p \tilde{q} \tag{36}$$

The time derivative of (36) can be expressed as below:

$$\dot{V}_1 = -\dot{\tilde{q}}^\mathsf{T} K_v \dot{\tilde{q}} - \dot{\tilde{q}}^\mathsf{T}\beta D_e \dot{q} - \dot{\tilde{q}}^\mathsf{T}\beta K_e(q - q_e) - \dot{\tilde{q}}^\mathsf{T}\gamma q_c \tag{37}$$

$$= -\dot{\tilde{q}}^\mathsf{T} K_v \dot{\tilde{q}} - \dot{q}^\mathsf{T}\beta D_e \dot{q} - \dot{q}^\mathsf{T}\beta K_e(q - q_e) - \dot{\tilde{q}}^\mathsf{T}\gamma q_c + \dot{q}_{ref}^\mathsf{T}\beta D_e \dot{q} + \dot{q}_{ref}^\mathsf{T}\beta K_e(q - q_e) \tag{38}$$

$$= -\dot{\tilde{q}}^\mathsf{T} K_v \dot{\tilde{q}} - \dot{q}^\mathsf{T}\beta D_e \dot{q} - \dot{q}^\mathsf{T}\beta K_e q + \dot{q}^\mathsf{T}\beta K_e q_e - \dot{\tilde{q}}^\mathsf{T}\gamma q_c + \dot{q}_{ref}^\mathsf{T}\beta T. \tag{39}$$

As the next step, we introduce the following Lyapunov function:

$$V_2 = V_1 + \frac{1}{2}q^\mathsf{T}\beta K_e q. \tag{40}$$

We can differentiate (40) as below.

$$\dot{V}_2 = \dot{q}_{ref}^\mathsf{T}\beta T - \dot{\tilde{q}}^\mathsf{T} K_v \dot{\tilde{q}} - \dot{q}^\mathsf{T}\beta D_e \dot{q} + \dot{q}^\mathsf{T}\beta K_e q_e - \dot{\tilde{q}}^\mathsf{T}\gamma q_c \tag{41}$$

Finally, we introduce the Lyapunov function given in (24), which can be expressed in terms of $V_2$.

$$V = V_2 + \frac{1}{2}\hat{q}^\mathsf{T} P\hat{q} \tag{42}$$

As stated previously, $\hat{q} = [q\ q_e]^\mathsf{T}$ and $P$ was given in (25), which included a skew-symmetric matrix $A$ and a positive semi-definite matrix $C$, i.e., $C \geq 0$. Its time derivative is computed as follows.

$$\dot{V} = \dot{V}_2 + \frac{d}{dt}\left(\frac{1}{2}\hat{q}^\mathsf{T} P\hat{q}\right) \tag{43}$$

Before computing the derivation of (27) based on (43), we begin by proving the positive definiteness of $\frac{1}{2}\hat{q}^\mathsf{T} P\hat{q}$.

$$\frac{1}{2}\hat{q}^\mathsf{T} P\hat{q} = \frac{1}{2}\begin{bmatrix} q \\ q_e \end{bmatrix}^\mathsf{T}\begin{bmatrix} 2A & -\beta K_e \\ -K_e^\mathsf{T}\beta^\mathsf{T} & 2C \end{bmatrix}\begin{bmatrix} q \\ q_e \end{bmatrix} \tag{44}$$

$$= q_e^\mathsf{T} C q_e - q^\mathsf{T}\beta K_e q_e \tag{45}$$

$C$ is a free design parameter such that the inequality $q_e^\mathsf{T} C q_e \geq q^\mathsf{T}\beta K_e q_e$ holds true. Therefore, the Lyapunov function $V$ given in (24) is positive definite.

As the next step, we derive (27) by showing that $\frac{d}{dt}\left(\frac{1}{2}\hat{q}^\mathsf{T} P\hat{q}\right) = -\dot{q}^\mathsf{T}\beta K_e q_e$. To do so, the first time derivative of (45) is yielded.

$$\frac{d}{dt}\left(\frac{1}{2}\hat{q}^\mathsf{T} P\hat{q}\right) = \dot{q}_e^\mathsf{T} C q_e + q_e^\mathsf{T} C\dot{q}_e - \dot{q}^\mathsf{T}\beta K_e q_e - q^\mathsf{T}\beta K_e \dot{q}_e \tag{46}$$

Since the environment is stationary, i.e., $\dot{q}_e = 0$, $\frac{d}{dt}\left(\frac{1}{2}\hat{q}^\mathsf{T} P\hat{q}\right) = -\dot{q}^\mathsf{T}\beta K_e q_e$ holds true. Therefore, Lyapunov function $V$ and its time derivative are obtained as stated in (24) and (26). When the controller gains are assigned as $\frac{K_{vi}}{b_i} = \frac{K_{pi}}{k_i}$ for each joint, (26) is reduced to (27).

# References

Ahmadi, M., & Buehler, M. (2006). Controlled passive dynamic running experiments with the ARL-monopod II. *IEEE Transactions on Robotics*, 22(5), 974–986.

Balafoutis, C. A., & Patel, R. V. (1991). Efficient computation of manipulator inertia matrices and the direct dynamics problem. *IEEE Transactions on Systems, Man, and Cybernetics*, 19, 1313–1320.

Batts, Z., Kim, J., & Yamane, K. (2016). Untethered one-legged hopping in 3d using linear elastic actuator in parallel (leap). In: *Proc. intl. symposium on experimental robotics, Tokyo, Japan* (pp. 103–112).

Bellicoso, C. D., Jenelten, C., Gehring, F., & Hutter, M. (2018). Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots. *IEEE Robotics and Automation Letters*, 3(3), 335–346.

Bloesch, M., Hutter, M., Hoepflinger, M., Leutenegger, S., Gehring, C., Remy, C.D., & Siegwart, R. (2012). State estimation for legged robots—Consistent fusion of leg kinematics and. In *Robotics: science and systems VIII, robotics: science and systems foundation*.

Boaventura, T., Semini, C., Buchli, J., Frigero, M., Focchi, M., & Caldwell, D. G. (2012). Dynamic torque control of a hydraulic quadruped robot. In *Proc. IEEE intl. conf. on robotics and automation (ICRA), St. Paul* (pp. 1889–1894).

Bowling, A. (2011). Impact forces and agility in legged robot locomotion. *Journal of Vibration and Control*, 17(3), 674–685.

Buchli, J., Kalakrishnan, M., Mistry, M., Pastor, P., & Schaal, S. (2009). Compliant quadruped locomotion over rough terrain. In *Proc. IEEE intl. conf. on intelligent robots and systems (IROS), St. Louis* (pp. 814–820).

Buschmann, T., Lohmeier, S., & Ulbrich, H. (2009). Biped walking control based on hybrid position/force control. In *Proc. IEEE intl. conf. on intelligent robots and systems (IROS), St. Louis* (pp. 3019–3024).

Caron, S., Kheddar, A., & Tempier, O. (2019). Stair climbing stabilization of the HRP-4 humanoid robot using whole-body admittance control. In *2019 International conference on robotics and automation (ICRA)*. IEEE.

Cavagna, G. A., Franzetti, P., Heglung, N. C., & Willems, P. (1988). The determinants of the step frequency in running, trotting and hopping in man other vertebrates. *Journal of Physiology*, 399(1), 81–92.

Cavanagh, P. R. (1987). The biomechanics of lower extremity action in distance running. *Foot & Ankle*, 7(4), 197–217.

Cho, B.-K., Kim, J.-H., & Oh, J.-H. (2011). Online balance controllers for running in the humanoid robot, hubo2. *Advanced Robotics*, 25(10), 1209–1225.

Diedrich, F. J., & Warren, W. H, Jr. (1995). Why change gaits? Dynamics of the walk-run transition. *Journal of Experimental Psychology*, 21(1), 183–202.

Ferris, D. P., Louie, M., & Farley, C. T. (1998). Running in the real world: Adjusting leg stiffness for different surfaces. *Proceedings of the Royal Society of London*, 265, 989–993.

Flayols, T., Prete, A. Del., Wensing, P., Mifsud, A., Benallegue, M., & Stasse, O. (2017). Experimental evaluation of simple estimators for humanoid robots. In *2017 IEEE-RAS 17th international conference on humanoid robotics (Humanoids)*. IEEE.

Fujimoto, Y., & Kawamura, A. (1998). Robust biped walking with active interaction control between foot and ground. In *Proc. IEEE intl. conf. on robotics and automation (ICRA), Leuven, Belgium* (pp. 2030–2035).

Geyer, H., Seyfarth, A., & Blickhan, R. (2006). Compliant leg behaviour explains basic dynamics of walking and running. *Proceedings of the Royal Society B*, 273(1603), 2861–2867.

Gollhofer, D., Schmidtbleicher, A., & Dietz, V. (1984). Regulation of muscle stiffness in human locomotion. *International Journal of Sports Medicine*, 15(1), 19–22.

Harada, K., Kajita, S., Kaneko, K., & Hirukawa, H. (2004). An analytical method on real-time gait planning for a humanoid robot. *International Conference on Humanoid Robots*, 3(1), 1–19.

Hartley, R., Ghaffari, M., Eustice, R. M., & Grizzle, J. W. (2020). Contact-aided invariant extended Kalman filtering for robot state estimation. *The International Journal of Robotics Research*, 39(4), 402–430.

Hurst, J. W., & Rizzi, A. A. (2008). Series compliance for an efficient running gait. *IEEE Robotics & Automation Magazine*, 15(3), 42–51.

Hyon, S.-H., & Cheng, G., 2007 Disturbance rejection for biped humanoids. In *Proc. IEEE intl. conf. on robotics and automation (ICRA), Rome* (pp. 2668–2675).

Jordan, K., & Newell, K. M. (2008). The structure of variability in human walking and running is speed-dependent. *Exercise and Sport Sciences Review*, 36(4), 200–204.

Kagami, S., Kitagawa, T., Nishiwaki, K., Sugihara, T., Inaba, M., & Inoue, H. (2002). A fast dynamically equilibrated walking trajectory generation method of humanoid robot. *Autonomous Robots*, 12(1), 71–82.

Kajita, S., Kaneko, K., Morisawa, M., Nakaoka, S., & Hirukawa, H., 2007 Zmp-based biped running enhanced by toe springs. In *Proc. IEEE intl. conf. on robotics and automation (ICRA), Rome* (pp. 3963–3969).

Kajita, S., Morisawa, M., Miura, K., Nakaoka, S., Harada, K., Kaneko, K., Kanehiro, F., & Yokoi, K., (2010). Biped walking stabilization based on linear inverted pendulum tracking. In *2010 IEEE/RSJ international conference on intelligent robots and systems*. IEEE.

Kajita, S., Nagasaki, T., Kaneko, K., & Hirukawa, H. (2007). Zmp-based biped running control. *IEEE Robotics & Automation Magazine*, 14(2), 63–72.

Kamioka, T., Kaneko, H., Kuroda, M., Tanaka,C., Shirokura, S., Takeda, M., & Yoshiike, T., (2017). Dynamic gait transition between walking, running and hopping for push recovery. In *IEEE/RAS intl conf. on humanoid robots* (pp. 1–8).

Kamioka, T., Kaneko, H., Kuroda, M., Tanaka, C., Shirokura, S., Takeda, M., & Yoshiike, T. (2017). Dynamic gait transition between walking, running and hopping for push recovery. In *2017 IEEE-RAS 17th international conference on humanoid robotics (humanoids)*. IEEE.

Kim, Y.-D., Lee, B.-J., Ryu, J.-H., & Kim, J.-H. (2007). Landing force control for humanoid robot by time-domain passivity approach. *IEEE Transactions on Robotics*, 23(6), 1294–1301.

Kormushev, P., Ugurlu, B.,Colasanto, L., Tsagarakis, N. G., & Caldwell, D. G. (2012) The anatomy of a fall: automated real-time analysis of raw force sensor data from bipedal walking robots and humans. In *IEEE/RSJ intl conf. on intelligent robots and systems* (pp. 3706–3713).

Kung, S. M., Fink, P. W., Legg, S. J., Ali, A., & Shultz, S. P. (2018). What factors determine the preferred gait transition speed in humans? a review of the triggering mechanisms. *Human Movement Science*, 57(1), 1–12.

Lebosse, C., Renaud, P., Bayle, B., & de Mathelin, M. (2011). Modeling and evaluation of low-cost force sensors. *IEEE Transactions on Robotics*, 27(4), 815–822.

Mistry, M., Buchli, J., & Schaal, S. (2010). Inverse dynamics control of floating base systems using orthogonal decomposition. In *Proc. IEEE intl. conf. on robotics and automation (ICRA), Anchorage*, (pp. 3406–3412).

Napoleon, N., Izu, H., Nakaura, S., & Sampei, M. (2005). An analysis of zmp control problem of humanoid robot with compliances in sole of the foot. In *The 16th IFAC world congress* (pp. 337–342).

Nelson, G., Saunders, N., Neville, A., Swilling, B., Bondaryk, J., Billings, D., et al. (2012). A humanoid robot for testing chemical protective clothing. *Journal of the Robotics Society of Japan*, *30*(4), 372–377.

Neunert, M., de Crousaz, C., Furrer, F., Kamel, M., Farshidian, F., Siegwart, R., & Buchli, J. (2016). Fast nonlinear model predictive control for unified trajectory optimization and tracking. In *Proc. IEEE intl. conf. on robotics and automation (ICRA), Stockholm* (pp. 1398–1404).

Olson, E. (2010). A passive solution to the sensor synchronization problem. In *2010 IEEE/RSJ international conference on intelligent robots and systems*. IEEE.

Orin, D. E., Goswami, A., & Lee, S.-H. (2013). Centroidal dynamics of a humanoid robot. *Autonomous Robots*, *35*(2), 161–176.

Orsolino, R., Focchi, M., Caron, S., Raiola, G., Barasuol, V., Caldwell, D. G., et al. (2020). Feasible region: An actuation-aware extension of the support region. *IEEE Transactions on Robotics*, *36*(4), 1239–1255.

Ott, C., Roa, M. A., & Hirzinger, G. (2011). Posture and balance control for biped robots based on contact force optimization. In *Proc. IEEE intl. conf. on humanoid robotics, Bled* (pp. 26–32).

Pratt, J., Carff, J., Drakunov, S., & Goswami, A. (2006). Capture point: A step toward humanoid push recovery. In *Proc. IEEE intl conf. on humanoid robots (humanoids), Genova*.

Raibert, M. H., Brown, H. B., & Chepponis, M. (1984). Experiments in balance with a 3D one-legged hopping machine. *The International Journal of Robotics Research*, *3*(2), 75–92.

Sakka, S., & Yokoi, K. (2005). Humanoid vertical jumping based on force feedback and inertial forces optimization. In *Proc. IEEE intl. conf. on robotics and automation (ICRA), Barcelona* (pp. 3752–3757).

Sanyal, A., & Goswami, A. (2013). Dynamics and balance control of the reaction mass pendulum (RMP): A 3D multibody pendulum with variable body inertia. *ASME Journal of Dynamics Systems, Measurement and Control*, *136*(2), 1–10.

Tajima, R., Honda, D., & Suga, K. (2009). Zmp-based biped running enhanced by toe springs. In *Proc. IEEE intl. conf. on robotics and automation (ICRA), Kobe* (pp. 1571–1576).

Takenaka, T., Matsumoto, T., & Yoshiike, T. (2009). Real time motion generation and control for biped robot-3rd report: Dynamics error compensation. In *Proc. IEEE intl. conf. on intelligent robots and systems (IROS), St. Louis* (pp. 1594–1600).

Tsagarakis, N. G., Caldwell, D. G., Negrello, F., Choi, W., Baccelliere, L., Loc, V. G., et al. (2017). Walk-man: A high-performance humanoid platform for realistic environments. *Journal of Field Robotics*, *34*(7), 1225–1259.

Ugurlu, B., & Kawamura, A. (2012). Bipedal trajectory generation based on combining inertial forces and intrinsic angular momentum rate changes: Eulerian zmp resolution. *IEEE Transactions on Robotics*, *28*(6), 1406–1415.

Ugurlu, B., Doppmann, C., Hamaya, M., Forni, P., Teramae, T., Noda, T., et al. (2016). Variable ankle stiffness improves balance control: Experiments on a bipedal exoskeleton. *I IEEE/ASME Transactions on Mechatronics*, *21*(1), 79–87.

Ugurlu, B., Kawasaki, T., Kawanishi, M., & Narikiyo, T. (2012). Continuous and dynamically equilibrated one-legged running experiments: Motion generation and indirect force feedback control. In *Proc. IEEE intl. conf. on intelligent robots and systems (IROS), Algarve* (pp. 1846–1852).

Ugurlu, B., Nishimura, M., Hyodo, K., Kawanishi, M., & Narikiyo, T. (2015). Proof of concept for robot-aided upper limb rehabilitation using disturbance observers. *IEEE Transactions on Human-Machine Systems*, *45*(1), 110–118.

Ugurlu, B., Saglia, J. A., Tsagarakis, N. G., Morfey, S., & Caldwell, D. G. (2014). Bipedal hopping pattern generation for passively compliant humanoids: Exploiting the resonance. *IEEE Transactions on Industrial Electronics*, *61*(10), 5431–5443.

Vukobratovic, M., & Borovac, B. (2004). Zero moment point—thirty five years of its life. *International Journal of Humanoid Robotics*, *1*(1), 157–173.

Wensing, P. M., & Orin, D. (2015). Improved computation of the humanoid centroidal dynamics and application in dynamic whole-body control. *International Journal of Humanoid Robotics*, *13*(1), 1–23.

Wensing, P. M., Wang, A., Seok, S., Otten, D., Lang, J., & Kim, S. (2017). Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots. *IEEE Transactions on Robotics*, *33*(3), 509–522.

Wensing, P. M., & Orin, D. E. (2013). High-speed humanoid running through control with a 3D-SLIP model. In *2013 IEEE/RSJ international conference on intelligent robots and systems*. IEEE.

Zhu, H., Sharma, M., Pfeiffer, K., Mezzavilla, M., Shen, J., Rangan, S., & Righetti, L. (2020). Enabling remote whole-body control with 5g edge computing. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE.

**Barkan Ugurlu** received the Ph.D. degree in Electrical and computer Engineering from Yokohama National University, Yokohama, Japan, in 2010, with the Monbukagakusho scholarship granted by the Ministry of Education, Culture, Sports, and Technology of Japan. From May 2010 to March 2013, he was a Post-Doctoral Researcher, with the Istituto Italiano di Tecnologia, Genova, Italy, and Toyota Technological Institute, Nagoya, Japan. Between March 2013 and February 2015, he was a Research Scientist with the Computational Neuroscience Laboratories, Advanced Telecommunications Research Institute International (ATR), Kyoto, Japan. He currently holds an Assistant Professor position with the Dept. of Mechanical Engineering, Ozyegin University, Istanbul, Turkey. His research interests include biological sensorimotor control and motor recovery, active orthoses and exoskeletons, robot-aided rehabilitation, humanoid/quadruped locomotion control, and multibody dynamics, force, and compliance control.



**Emre Sariyildiz** received his first Ph.D. degree in Integrated Design Engineering from Keio University, Tokyo, Japan, in September 2014 and second Ph.D. degree in Control and Automation Engineering from Istanbul Technical University, Istanbul, Turkey, in February 2016. He was a research fellow in the department of Biomedical Engineering and Singapore Institute for Neurotechnology (SINAPSE) at National University of Singapore, Singapore, between 2014 and 2017. Since

April 2017, he has been Lecturer in the School of Mechanical, Materials, Mechatronic and Biomedical Engineering at University of Wollongong, NSW, Australia. His main research interests are control theory, robotics, mechatronics and motion control.



**Takao Kawasaki** received his Master's degree in Control Engineering from Toyota Technological Institute, Nagoya, Japan, in 2012. Throughout his graduate studies he focused on the development and locomotion control of legged robots. His main research interest is the implementation of engineering methods in real-world tasks.



**Tatsuo Narikiyo** received his Ph.D. degree in Control Engineering from Nagoya University, Nagoya, Japan, in 1984. He was a research scientist in the Japanese Governmental Industrial Research Institute, Kyushu, Japan, from 1983 to 1990. From April 1990 he was an assistant professor and from April 1998 to March 2018 he was a professor in Toyota Technological Institute, Nagoya, Japan. Currently he is a Distinguished Professor at the same Institute. His main research interest is control system synthesis for nonlinear mechanical systems.