CLAWAR 2019: 22nd International Conference on
Climbing and Walking Robots and the Support
Technologies for Mobile Machines,
Kuala Lumpur, Malaysia, 26-28 August 2019.
https://doi.org/10.13180/clawar.2019.26-28.08.15

Reaction to Touch: Evasion and Physical Guidance with the Lightweight Quadruped Robot Serval

Boxing Wang[1,2], Peter Eckert[1], Chunlin Zhou[2], Jun Wu[2,3] and Auke Jan Ijspeert[1]
[1] *Biorobotics Laboratory, EPFL, Switzerland*
[2] *College of Control Science and Engineering, Zhejiang University, China*
[3] *Binhai Industrial Technology Research Institute of Zhejiang University, China*

Physical human-robot interactions are commonly seen in many legged robot application scenarios. Although physical interactions could affect the stability of the robot, robots can be more versatile and adaptive by performing different reactions according to tactile input. In this paper, we propose an intuitive control framework that enables the trotting quadruped robot Serval to use the same, modular control framework, triggering different behaviors according to the perturbation experienced. In order to detect the tactile input, a force sensor is attached to the middle spine joint of Serval. For gentle taps, a central pattern generator model is used to perform open-loop control. As gentle taps induce few disturbances to the stability, the robot is open-loop controlled to move laterally as a reaction. For strong pushes, virtual model and stability margin control are added to bend the spine and adjust supporting legs to keep balance. As no sidesteps are required, the robot still keeps moving on its original route. Experiments are presented to validate the performance of the proposed control framework.

1. Introduction and Motivation

Recent developments in the world of robotic animals show potential to bridge the existing reality gap from lab to real life environments.[1–3] In this endeavor, it becomes apparent that human-robot-interaction is an issue to be addressed to enable a safe coexistence. Most developers in the quadruped realm focus on protecting the robot hardware from being damaged by avoiding physical contact as often as possible, e.g through mapping ”no-go”-zones. Approaches like the ones in robot manipulation or robots for tactile interaction such as physical guidance or compliant contacts are not yet thoroughly explored in the quadruped robot realm.[4]

To further our understanding of human-robot interaction and enable safe coexistence for the robot and its surroundings, we will introduce an intuitive control scheme featuring balance control and physical guidance with our lightweight and compliant quadruped robot Serval.[5] The approach, unlike other push adaption control methods, allows the robot to keep its original route under strong pushes but perform lateral moving under soft taps. Thus, a different touch-reaction pattern is introduced.

The rest of the paper is laid out as follows: related work is introduced in Sect. 2; Mechatronic structure, model analysis and detailed control framwork are introduced in Sect. 3; Experimental results are presented in Sect. 4; Future work is discussed in Sect. 5.

2. Related Work

To coexist in a populated environment, mainly two features have to be present in a robot: (1) The robot has to be able to adapt to external interference. (2) The robot has to react to outside stimuli.

2.1. *Stabilization Strategies for Quadruped Robots*

Among different techniques, stability criteria are often used to plan some few footsteps in the future and modify the position of the center of mass (CoM) to keep balance when disturbances are induced.

Stability criteria that consider robot dynamics, such as zero moment point and capturability, often provide high performance. Zero moment point[6,7] (ZMP) stability criterion combined with preview control enables the robot to recover from pushes by adjusting the position of the center of mass.[8] Capturability[9] is a stability criterion that focuses on using a limited number of steps to stop a legged robot. It has been successfully applied on HyQ[10] and Atlax.[11] However, these criteria often require complex control structure and a high-performance dynamics tracking system, which makes it hard to be implemented on small robots with physical compliance.

Simple stability criteria, such as different types of energy stability margin[12–15] , are based on adjusting the relation between the CoM and the support polygon. Stability margin gives a robot the ability of self-stabilization while not requiring a profound knowledge of the robot dynamics. Virtual model control[16,17] is another fast way to achieve stability. It sets virtual springs and dampers to decide the desired joint torque to stabilize. No dynamical model of the robot is needed. The above methods can be easily tuned and integrated into different types of controllers.

2.2. *Tactile Human-Robot-Interaction (THRI)*

Instead of going too much into details of the classification of THRI we refer the interested reader to study the review article[4] . To control the behavior of our robot via THRI achieving a higher level of safety and being able to directly manipulate movement trajectories are of rather simple nature. For physical interactions that do not affect the stability, our robot needs only to work in an open-loop mode, changing its state according to these interactions, in a sense like being controlled by a joystick. One way to realize this is to modify an open-loop controller. For this work, a central pattern generator model[18] named dynamical movement primitives (DMP)[19–21] is chosen. DMP is a weak nonlinear system where the shape of its limit cycle can be easily predefined. As a CPG model can change the amplitude and the frequency of its output through only a few parameters, it is suitable for controlling the walking speed and direction of a quadruped robot, which suffices in this study.

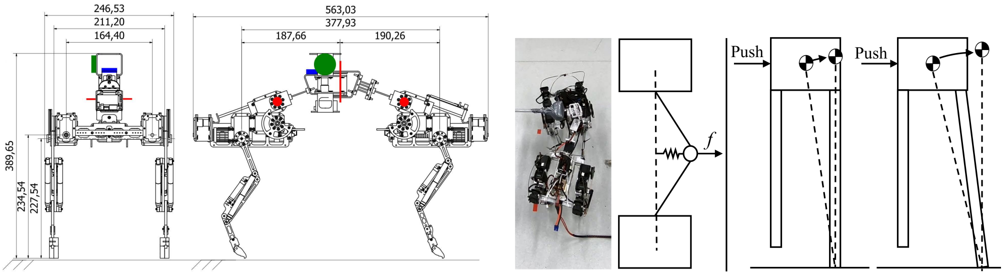
3. Method

3.1. *Mechatronic Structure of Serval*

Serval (Fig. 1a) consists of an active 3-DOF spine, leg units with adduction/abduction (AA) mechanism and a scaled ASLP-version of Cheetah-Cub-AL.[22] All motors are combined with in-series elastics to avoid possible damages. The robot consists of a low-cost IMU (Biscuit) and force sensors (Optoforce OMD-30-SE-100N), which are attached to the spine joint for physical contact detection. Serval can execute different motion skills completely tether-less and has a total weight of about $3.5kg$. The interested reader is pointed to Ref. 5 for further details on rich open loop behavior.

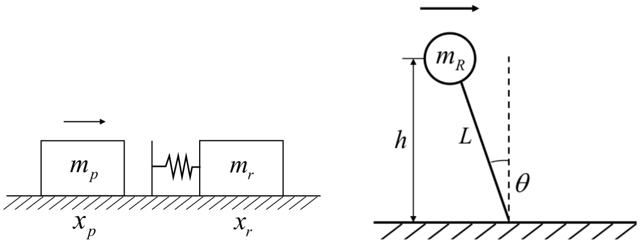
3.2. *Model Analysis*

In order to keep balance under large lateral pushes, a method which combines virtual model with stability margin control is proposed. As illustrate in Fig. 1b, when the attached force sensor detects pushes, first, virtual spring control is applied to the spine to soften the impact; Second, supporting legs will move laterally to prevent sidewards falling.



(a) (b)

Fig. 1. (a) Schematic side view of Serval illustrating the positioning of the spinal joints (red), IMU (blue) and force sensors (green). (b) Physical illustration of the push adaptation method. Left: virtual spring control for the spine. Right: lateral foot-end displacements to prevent sidewards falling.



(a) (b)

Fig. 2. (a) Spring collision model (b) Simplified model of lateral foot-end position adjustment.

Virtual spring control makes no direct contributions for balance control, however, it extends the collision time which is the precondition for other future reactions. For simplicity, we simplify this process into a spring collision model. As shown in Fig. 2a, a push is modeled as a mass block with a initial velocity; The robot, whose spine is controlled by virtual spring law, is modeled as a stationary mass block with a massless spring in front of it.

According to the momentum and energy conservation law, it is derived that:

$$m_p\dot{x}_{p0} = m_p\dot{x}_p + m_r\dot{x}_r \tag{1}$$

$$\frac{1}{2}m_p\dot{x}_{p0}^2 = \frac{1}{2}m_p\dot{x}_p^2 + \frac{1}{2}k\Delta x^2 + \frac{1}{2}m_r\dot{x}_r^2 \tag{2}$$

where $\dot{x}_{p0}$ is the initial velocity of the push-represented mass block, $\dot{x}_p$ and $\dot{x}_r$ are the velocity of push-represented and robot-represented block at any time instant. k is the spring stiffness and Δx is the compression of the spring.

After enough time, two blocks will reach their steady states, which are same as the result of a perfect elastic collision:

$$\dot{x}_p = \frac{(m_p - m_r)\dot{x}_{p0}}{m_p + m_r} \tag{3}$$

$$\dot{x}_r = \frac{2m_p\dot{x}_{p0}}{m_p + m_r} \tag{4}$$

Thus, the spring does not change the magnitude of the energy injected into the robot. However, as a temporary energy storage component, it does affect the shortest distance between two blocks:

$$\Delta x|_{max} = \left(\frac{m_p m_r}{k(m_p + m_r)}\right)^{\frac{1}{2}} \cdot \dot{x}_{p0} \tag{5}$$

where $\Delta x|_{max}$ is the maximum compression length of the spring, which is also the shortest distance between two blocks. It is derived under the condition when $\dot{x}_p = \dot{x}_r$. Therefore, the smaller the k is, the longer distance is needed to finish the energy injection, and more time the robot could have to give further reactions.

Adjustment of lateral foot-end position prevents sidewards falling. This adjustment, as illustrated in Fig. 1b, will move the foot-end in the opposite direction of the lateral push. As it is the trot gait used in this study, there is always a supporting leg in the other side of the push. Thus, the model can be simplified as shown in Fig. 2b.

In Fig. 2b, m_R is the mass of the block, L is the length of the non-retractable link, h is the initial height, and θ is the tilt angle. The dynamics is described as

$$m_R L^2 \ddot{\theta} = m_R g L \sin(\theta) \tag{6}$$

and the initial conditions are:

$$\theta_0 = \arccos(L/h) \tag{7}$$

$$\dot{\theta}_0 = v_0 \cdot \cos(\theta_0) \tag{8}$$

where v_0 is the resulted horizontal velocity after the push. Thus, driving the foot-end move in the opposite direction of the lateral push will increase the value of θ_0 and reduce the value of $\dot{\theta}_0$, which in other words increases the energy threshold of the robot to fall and decreases the lateral speed caused by pushes. Both aspects increase stability.

Theoretically, this foot-end adjustment could be used alone. However, without the virtual spring control, the adjustment could cause excessive motor load. With the virtual spring control, according to (5), a time window before the interaction force reaches its maximum could be created.

3.3. *Controller Design*

The modular controller we propose consists of four parts, see Fig. 3: (1) the walking pattern generator, (2) the balance compensation module, (3) the spine control module and (4) the inverse kinematics module. Outputs of the first three modules are desired foot-end positions in Cartesian space. They will be added together and then transfered into the inverse kinematics module to derive final servo positions commands.

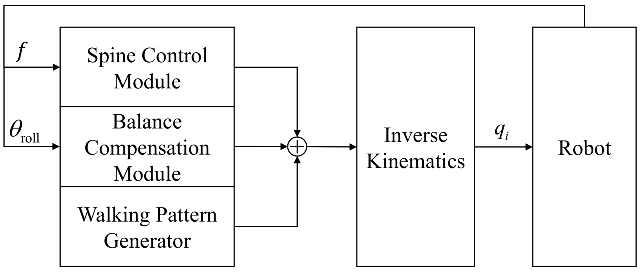


Fig. 3. Controller structure with walking pattern generator, balance compensation, spine control module and inverse kinematics modules

The walking pattern generator produces the basic foot-end trajectory for trotting on flat grounds, which defines stride length, step height and stance phase ratio. This pattern generator is based on the DMP as mentioned earlier. Its limit cycle can be arbitrarily predefined and reshaped by changing scale factors in real time. Equations of DMP are:

$$\tau\dot{\phi} = 1 \tag{9}$$

$$\tau\dot{\boldsymbol{w}} = \alpha[\beta(r_s - \boldsymbol{r}) - \boldsymbol{w}] + \boldsymbol{K}_a\boldsymbol{f}(\phi) \tag{10}$$

$$\tau\dot{\boldsymbol{r}} = \boldsymbol{w} \tag{11}$$

where $\boldsymbol{w} = [w_1, w_2, ..., w_i]^T$ and $\boldsymbol{r} = [r_1, r_2, ..., r_i]^T$ are state variables. $\boldsymbol{r}$ is the final multi-dimensional output. $\tau = T/(2\pi)$ is a temporal scaling factor, and T is the desired trajectory period. r_s is the baseline of the oscillation, which is set to 0. α and β are positive constants. $\boldsymbol{f}(\phi)$ is the vector named forcing term, which is used to drive $\boldsymbol{r}$ to generate desired output trajectory. $\boldsymbol{K}_a = \mathrm{diag}(k_{a1}, k_{a2}, ..., k_{ai})$ is the scaling factor for tuning the trajectory amplitude in real time. For a predefined output trajectory r_d, the corresponding $f(\phi)$ can be calculated as follows:

$$f_d(\phi) = \tau^2\frac{\mathrm{d}^2 r_d(\phi)}{\mathrm{d}\phi^2} + \alpha\frac{\mathrm{d}r_d(\phi)}{\mathrm{d}\phi} + \alpha\beta(r_d - r_s) \tag{12}$$

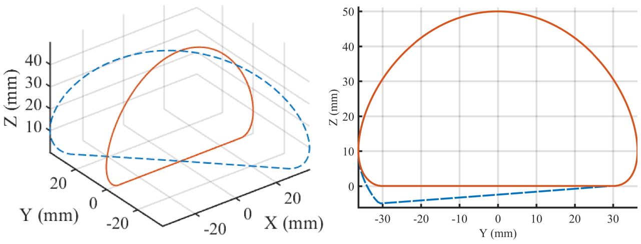


Fig. 4. Predefined foot-end trajectory in the walking pattern generator. Left: Predefined trajectory with and without zero lateral amplitude. Right: Stance phase compensation.

In this study, we use DMP to generate 3-dimensional foot-end trajectories. As shown in Fig. 4 left, the predefined trajectories are constructed by an arc for swing phase and a straight line for stance phase. In normal trotting, k_{ai} that corresponds to the lateral amplitude is set to 0 to achieve a purely forward moving. When requested (from the spine control module) to execute a lateral movement, this parameter will be changed to a non-zero value, resulting lateral speed.

In order to keep the robot balanced during undisturbed, level trotting, only trajectory offsets, and stance phase compensations are needed. Stance phase compensation (as shown by the dashed line in Fig. 4 right) means adding a slope to the trajectory in the stance phase for hind legs. The affected hind legs, as a result, stretch more before entering the swing phase, reducing the impacts when corresponding swing legs enter stance phase and offsetting the possible body pitch angle deviation. Both trajectory offsets and stance phase compensation are defined empirically with few trials.

The balance compensation module plays a secondary role to keep balance. During experiments, it was observed that the open-loop pattern generator was sensitive to deviations from the perfectly flat ground and small hardware modifications. Thus, during real-time running, this module is used to increase the robustness by outputting small lateral foot-end displacements according to measured roll angle:

$$\dot{p}_{BC} = k_{BC} \cdot \theta_{roll} \tag{13}$$

where p_{BC} is the lateral displacement output by the module, k_{BC} is a constant value and θ_{roll} is the roll angle of the robot. Note that the output of this module will only be applied to the supporting legs and is constrained to small values (less than 15 mm in our experiments) to prevent unwanted lateral drift.

The spine control module controls the spine deflection in response to a push. If a force sensor detects a push, a lateral displacement for the midpoint of the spine will be generated according to the virtual model control. Specifically, a virtual spring that we assume to attach to the midpoint of the spine is expressed in the following control law:

$$p_{spine} = k_{spine} \cdot f \tag{14}$$

where p_{spine} is the lateral displacement for the midpoint of the spine (as shown in Fig. 1b left), k_{spine} is a constant value and f is the force value measured by the force sensor (for now only the value along lateral direction is considered). In our experiments, k_{spine} was set to 8 mm/N, and p_{spine} was limited to $[0,\ 50]$ mm.

As analyzed in subsection 3.2, when a push is detected, the lateral foot-end position adjustment is applied immediately to keep balance:

$$p_s = k_s \cdot p_{spine} \tag{15}$$

where p_s is the output foot-end displacement. k_s is a small constant value, which usually varies from 0.3-0.5, depending on the amplitude of the push. For soft pushes (light slapping), k_s is set to 0.3, while for hard pushes (light kicks), k_s is set to 0.45.

4. Experiments and Results *

4.1. *Punching test during trotting*

This test is designed to test the push resistance during a straight trot. A spring-loaded piston is used to perturb the robot with a controllable impulse. Comparison between the results with and without the activated spine control module are presented. The overall motion in Fig. 5 is complemented by the corresponding data record Fig. 6.

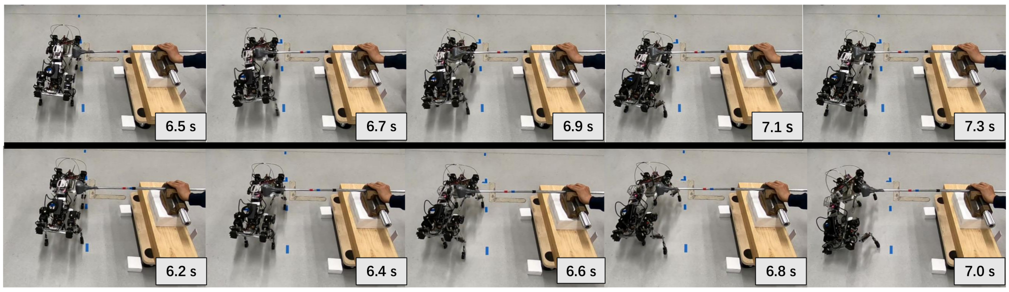


Fig. 5. Overview of the punching test. Above: video record with active spine control module (success of perturbation compensation). Below: video record with inactive spine control (failure through fall).

In Fig. 6, balance was kept within one trotting period through compensation using the spine control module. The force data reached its peak value earlier than the roll angle did, and the feasibility of the parallel structure employed in the proposed controller was verified.

4.2. *THRI: Obstacle avoidance through physical guidance*

In this test (Fig. 7), we investigated a different way induce lateral movement. For strong pushes ($F > 1N$), the proposed controller will activate the spine control module. Serval will bend its spine and adjust its foot locations to absorb the push, and finally stay in the straight trotting line. For small pushes (intended for physical guidance, $F < 1N$), where the balance will not be affected, the spine control module will directly send a lateral moving signal to the walking pattern generator in order to perform an open-loop lateral moving.

*The experimental video could be found at https://www.youtube.com/watch?v=BqK0tQkY_Zw

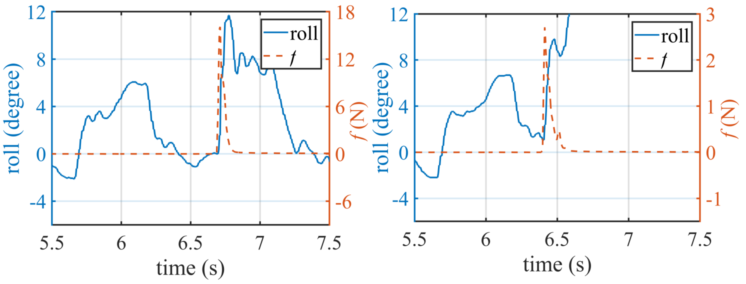


Fig. 6. Roll angle and force data record of the punching tests. Left: record with active spine control module (success of perturbation compensation). right: record with inactive spine control module (failure through fall with lower force).

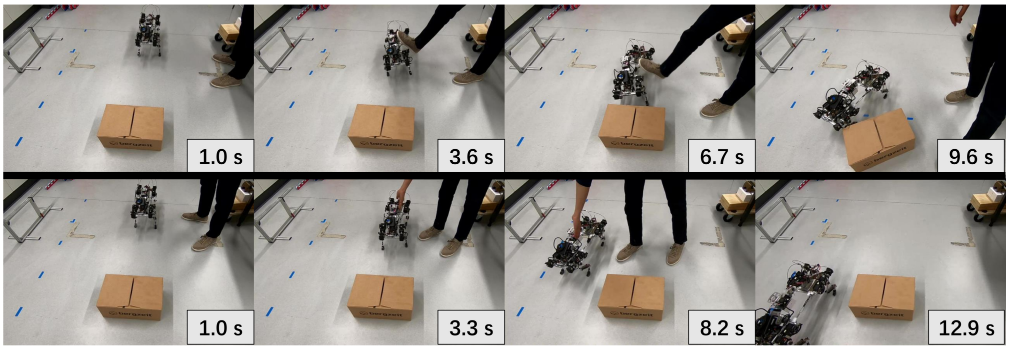


Fig. 7. Overview of the THRI test. Above: Kicks cannot change the walking course of the robot. Below: Small taps will activate the walking pattern generator to perform an open-loop lateral moving.

Fig. 7 illustrates the video record of the THRI test. The obstacle placed in the robot's path would only be avoided if soft taps were applied to the force sensor. Kicks activated our strategy that prioritizes robot stability resulting in little deviation from the path, ultimately leading to the collision with the obstacle.

5. Discussion and Future Work

In this article, we proposed an intuitive control framework that takes advantage of one spinal joint to let the robot stay in its original route under strong pushes but perform open-loop lateral moving under gentle taps. Within the proposed control framework, final foot-end trajectories are generated by combining commands from different modules with different functions. However, for now, we can only verify the effectiveness of this decoupling through experiments. In the future, we will work on the theoretical proof. Additional, we aim to equip Serval with a complete sensory network, moving from point contact of one lateral force sensor to a skin-like sensor network. Depending on the region touched, the robot could perform different actions, ranging (but not exclusive) from sitting down/standing up to changing gait, speed and direction.

6. Acknowledgements

We want to thank all who supported this work with ideas, constructive criticism, and hands-on experience. We acknowledge funding sources from Qingdao National Laboratory for Marine Science and Technology (2017WHZZB0302), Zhejiang Provincial & National Natural Science Foundation of China (Y18F030012 & 61836015), the State Key Laboratory of Industrial Control Technology, Zhejiang University, China (ICT1807), and EPFL, Switzerland.

References

1. “Boston Dynamics — Boston Dynamics.” https://www.bostondynamics.com/bigdog.
2. M. Hutter, C. Gehring, D. Jud, *et al.*, “ANYmal - a highly mobile and dynamic quadrupedal robot,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 38–44, IEEE, oct 2016.
3. A. T. Sprowitz, A. Tuleu, M. Ajaoolleian, *et al.*, “Oncilla robot: a versatile open-source quadruped research robot with compliant pantograph legs,” *Frontiers in Robotics and AI*, vol. 5, p. 67, 2018.
4. B. D. Argall and A. G. Billard, “A survey of Tactile HumanRobot Interactions,” *Robotics and Autonomous Systems*, vol. 58, pp. 1159–1176, oct 2010.
5. P. Eckert, A. E. M. Schmerbauch, T. Horvat, *et al.*, “Towards rich motion skills with the lightweight quadruped robot serval - a design, control and experimental study,” in *International Conference on Simulation of Adaptive Behavior*, (Frankfurt, Germany), pp. 41–55, 2018.
6. S. Kajita, H. Hirukawa, K. Harada, *et al.*, *Introduction to Humanoid Robotics*. Springer Berlin Heidelberg, 2014.
7. M. Vukobratović and B. Borovac, “Zero-moment point – thirty five years of its life,” *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2004.
8. J. Urata, K. Nshiwaki, Y. Nakanishi, *et al.*, “Online decision of foot placement using singular lq preview regulation,” in *IEEE-RAS 11th International Conference on Humanoid Robots (Humanoids)*, (Bled, Slovenia), pp. 13–18, 2011.
9. T. Koolen, T. D. Boer, J. Rebula, *et al.*, “Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models,” *International Journal of Robotics Research*, vol. 31, no. 9, pp. 1094–1113, 2012.
10. V. Barasuol, J. Buchli, C. Senimi, *et al.*, “A reactive controller framework for quadrupedal locomotion on challenging terrain,” in *Proceedings of IEEE International Conference on Robotics and Automation*, (Karlsruhe, Germany), pp. 2554–2561, 2013.
11. G. Wiedebach, S. Bertrand, T. Wu, *et al.*, “Walking on partial footholds including line contacts with the humanoid robot atlas,” in *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, (Cancun, Mexico), pp. 1312–1319, 2016.
12. E. Garcia and P. González de Santos, “A new dynamic energy stability margin for walking machines,” in *Proceedings of International Conference on Advanced Robotics*, (Coimbra, Portugal), pp. 1–6, 2003.
13. B. S. Lin and S. M. Song, “Dynamic modeling, stability and energy efficiency of a quadrupedal walking machine,” *Journal of Robotic Systems*, vol. 18, no. 11, pp. 657–670, 2001.
14. E. G. Papadopoulos and D. A. Rey, “A new measure of tipover stability margin for mobile manipulators,” in *Proceedings of IEEE International Conference on Robotics and Automation*, (Minneapolis, USA), pp. 3111–3116, 1996.
15. D. A. Messuri, *Optimization of the locomotion of a legged vehicle with respect to maneuverability*. PhD thesis, The Ohio State University, USA, 1985.
16. M. Ajallooeian, S. Pouya, A. Sproewitz, *et al.*, “Central pattern generators augmented with virtual model control for quadruped rough terrain locomotion,” in *Proceedings of IEEE International Conference on Robotics and Automation*, (Karlsruhe, Germany), pp. 3321–3328, 2013.
17. G. Zhang, X. Rong, C. Hui, *et al.*, “Torso motion control and toe trajectory generation of a trotting quadruped robot based on virtual model control,” *Advanced Robotics*, vol. 30, no. 4, pp. 284–297, 2016.
18. A. J. Ijspeert, “Central pattern generators for locomotion control in animals and robots: a review,” *Neural Networks*, vol. 4, no. 21, pp. 642–653, 2008.
19. A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, *et al.*, “Dynamical movement primitives: Learning attractor models for motor behaviors,” *Neural Computation*, vol. 2, no. 25, pp. 328–373, 2013.
20. C. Zhou, B. Wang, Q. Zhu, *et al.*, “An online gait generator for quadruped walking using motor primitives,” *International Journal of Advanced Robotic Systems*, vol. 6, no. 13, pp. 1–12, 2016.
21. M. Ajallooeian, J. van den Kieboom, A. Mukovskiy, M. A. Giese, and A. J. Ijspeert, “A general family of morphed nonlinear phase oscillators with arbitrary limit cycle shape,” *Physica D*, no. 263, pp. 41–56, 2013.
22. P. Eckert, “Cheetah-Cub-AL — BioRob.” https://biorob.epfl.ch/page-131896-en.html.