Modeling Proprioceptive Sensing for Locomotion Control of Hexapod Walking Robot in Robotic Simulator

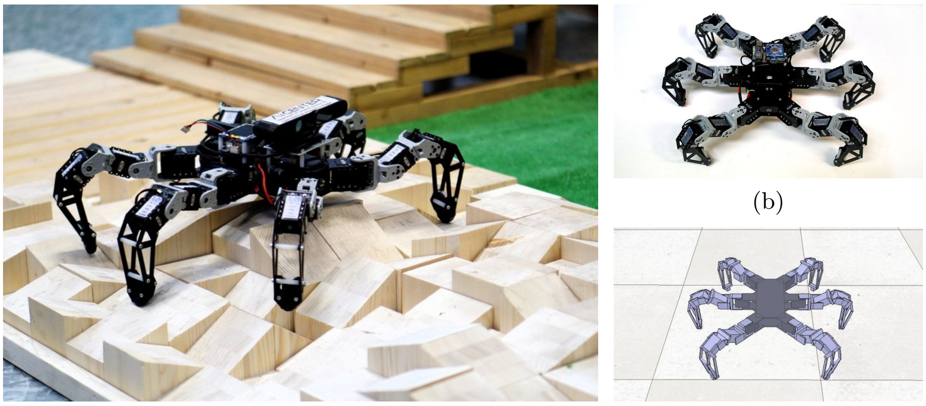
Minh Thao Nguyenová, Petr Čížek$^{[0000-0001-6722-3928]}$, and Jan Faigl$^{[0000-0002-6193-0792]}$

Faculty of Electrical Engineering, Czech Technical University in Prague,
Technicka 2, 166 27, Prague, Czech Republic
`{nguyemi5,cizekpe6,faiglj}@fel.cvut.cz`
`https://comrob.fel.cvut.cz`

Abstract. Proprioceptive sensing encompasses the state of the robot given by its overall posture, forces, and torques acting on its body. It is an important source of information, especially for multi-legged walking robots because it enables efficient locomotion control that adapts to morphological and environmental changes. In this work, we focus on enhancing a simplified model of the multi-legged robot employed in a realistic robotic simulator to provide high-fidelity proprioceptive sensor signals. The proposed model enhancements are based on parameter identification and static and dynamic modeling of the robot. The enhanced model enables the V-REP robotic simulator to be used in real-world deployments of multi-legged robots. The performance of the developed simulation has been verified in the parameter search of dynamic locomotion gait to optimize the locomotion speed according to the limited maximal torques and self-collision free execution.

1 Introduction

Modeling and dynamic simulations are established tools in robotics to enable verification of algorithms and control strategies before their deployment on real robots. Especially robots with complex morphology benefit from simulations to enhance their abilities to navigate the environment. It is the case of multi-legged robots [13] whose individual legs are connected through the trunk and also through the ground which all together form a complex linkage system with dynamic coupling between the individual components [10]. Simplifications and specific assumptions might be introduced in modeling of such systems, e.g., on rigidity of the construction [12], non-slippery footholds [10] or actuator dynamics [15]. However, it is important to verify that the considered simplifications and assumptions do not provide imprecise real-world execution that may result in mission failure or damage of the robot. Therefore, it is crucial to have a high-fidelity model and simulation of the real world behavior of the robot to support a seamless deployment of developed algorithms on real platforms even in



(a)

(b)

(c)

Fig. 1. (a) Hexapod walking robot in a rough terrain where the locomotion gait parametrization found using the developed high-fidelity simulation can prevent the robot from damaging the actuators. (b) The hexapod robot in the default pose and (c) its model in the V-REP robotic simulator.

complex scenarios like robotic football [16] or deployment in soft-terrains [7]. Besides, modeling is also necessary to obtain a collision checker for motion planning techniques and advanced locomotion control strategies, e.g., as in [1, 2].

The presented work is motivated by the real-world deployment of a hexapod walking robot shown in Fig. 1. We aim to have a realistic robotic simulator with a model of the robot, and its identified parameters would provide high-fidelity simulated measurements. The electrically actuated hexapod robot is built from off-the-shelf components, and each of its six legs has three joints motorized with the Dynamixel AX-12 servomotors that provide position feedback only. Thus, we are looking for a model of the robot that allows us to estimate the proprioceptive measurements, such as joint torque values that are not directly provided by the robotic platform. In particular, we need a simulation of the robot that allows finding a proper parametrization of the dynamic locomotion gait [14] prior the experimental evaluation to ensure safe and reliable operation by disallowing high joint torques, which would otherwise damage the actuators. Searching for such a parametrization during real-world experiments might have fatal consequences for the robotic platform. Hence, the simulation is a safe, fast, efficient, and inexpensive way for the development of the desired locomotion controllers.

There are multiple commercial and open-source simulation tools and libraries that support realistic robotic simulations. In our work, we utilize the Coppelia Robotics V-REP [6] realistic simulator, as it supports the modeling of static and dynamic objects and their interactions using different physics engines. We have created a simulation model of our hexapod walking robot with the emphasis on the identification of the model parameters to make it close to the real robotic platform. The achieved results in the simulation of the robot are supported by several verification experiments. Moreover, the developed simulation model of the robot has been employed for parameter search of the dynamic locomotion gait [14] to optimize the locomotion speed considering restrictions on the maximum joint torques and self-collision free execution.

The paper is organized as follows. Section 2 briefly introduces the utilized robotic simulator. The description of the modeling and parameters identification is in Section 3 and the results on the experimental verification of the model are reported in Section 4. Concluding remarks are dedicated to Section 5.

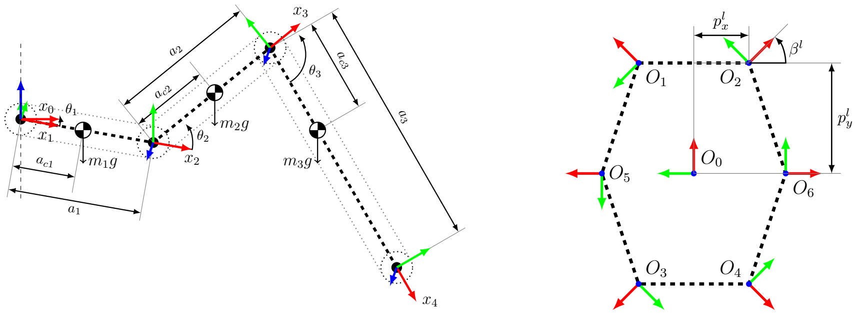
2 V-REP Realistic Robotic Simulator

V-REP is a powerful cross-platform 3D simulator based on distributed control architecture, i.e., control programs (or scripts) can be directly attached to individual scene objects and run simultaneously in threaded or non-threaded fashion [9]. Outside the control, the dependency of dynamic shapes can be set to simulate objects that move dependently on other objects, e.g., links between two joints of a leg. Most of the element properties can be reached and changed in the simulator GUI or via remote API client. Static properties such as proportions can be set for any shape; however, dynamic properties (e.g., mass, inertia, friction) are specific for dynamic objects only. Dynamic objects interact with the environment and other dynamic objects, and the computation of the dynamics can be done by one of four different physics engines, namely the Bullet physics library [5], Open Dynamic Engine (ODE) [11], Vortex Dynamics [3], and Newton Dynamics [8]. For the herein presented work, we use the Bullet physics library [5], which we found to be both the open source and sufficiently accurate [4].

3 Hexapod Modeling in V-REP Simulator

The addressed problem is to model the real hexapod walking robot in a realistic robotic simulation. Specifically, we aim to minimize the error between the proprioceptive data measured on a real robot and in the simulator. Thus, the goal is to estimate the proprioceptive signals that are not directly measured by the real robot, but they are acting in the robot interaction with the environment. Therefore, the requested realistic simulation has to provide the equivalent level of the interaction that allows developing locomotion control strategies using only the simulator before the experimental evaluation with the real robot. The real hexapod robot and the created visual model are shown in Fig. 1.

The modeled robot is an electrically actuated hexapod robot built from off-the-shelf components. The robot consists of the base trunk with revolute joints moving around the vertical axis and a control unit. Each leg is attached to a single joint and is formed by two more linked revolute joints moving around the horizontal axis. Parts of each leg are named coxa, femur, and tibia as it is visualized in Fig. 2a. Each joint is motorized by the Dynamixel AX-12A servomotor with the position feedback only. All 18 servomotors are connected in a daisy chain where all servomotors can be set with a new desired position within 1 ms. The current servomotor position can be obtained from the individual servomotor within 1 ms. Therefore commanding all the servomotors with new desired



(a) Leg schematic (b) Body schematic

Fig. 2. (a) Schematic of the robot leg. Each leg has three parts (links) – coxa, femur, and tibia connected by three joints (θ_1, θ_2, and θ_3). The coxa joint is fixed to the body with a vertical rotation axis while the two other joints are oriented with respect to the horizontal axis. (b) Schematic of the robot body. Each coxa joint corresponds to the respective origin of the coordinate frame O_l.

positions and reading back their current positions take altogether 19 ms which is a crucial property that has to be taken into account when modeling the robot dynamics. In the rest of this section, the robot modeling with the parameter identification is presented that is further followed by the description of the applied dynamic modeling of the actuators.

3.1 Kinematic Model of the Robot Body

The robot model consists of the elementary shapes modeled according to the original robot parts that correspond with the real world proportions and morphology of the real robot. The original proportions are respected with minor geometric simplifications of individual parts to speed up the simulation and avoid unnecessary computations. The modeled dimensions and masses of the individual model shapes have been set according to Table 1 and Table 2 that list the properties of the robot leg and trunk, respectively, see Fig. 2. The total mass of the real robot is distributed evenly in each of the utilized elementary shapes. The inertia matrices of each link have been computed by the simulator under the assumption of evenly distributed mass. The individual shapes are connected to move dependently according to the robot morphology, i.e., the kinematic chains are formed with the hierarchical structure starting from the hexapod body and ending in the tibia link.

3.2 Dynamic Model of Servomotors

The properties of the Dynamixel AX-12A servomotor have been modeled using the V-REP simulator that supports different actuator models such as linear, revolute, and prismatic; each with several adjustable parameters. The basic revolute actuator in the force/torque control mode is used to model the servomotor because it best fits our scenario. The simulator supports different control modes

Table 1. Body parameters

l	1	2	3	4	5	6
β^l [rad]	$3\pi/4$	$\pi/4$	$5\pi/4$	$7\pi/4$	π	0.0
p_x^l [mm]	-60.5	60.5	-60.5	60.5	-100.5	100.5
p_y^l [mm]	120.6	120.6	-120.6	-120.6	0.0	0.0
Body mass (without battery)						1212 g
Mass of the battery						330 g

Table 2. Leg parameters

Link name	i	a_i [mm]	a_{ci} [mm]	m_i[g]
Coxa	1	52	26	22
Femur	2	66	20	72
Tibia	3	138	51	104

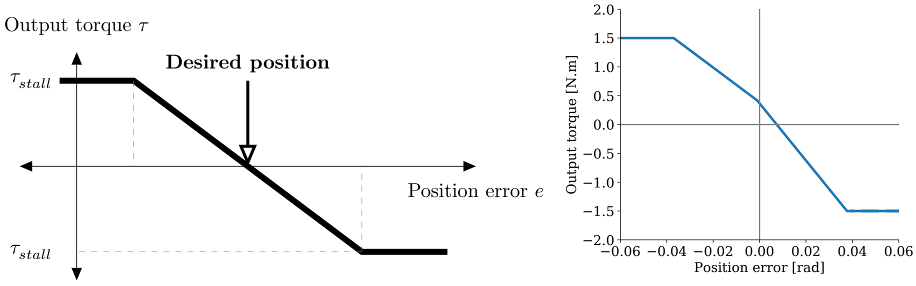
of the actuator including PID position control, spring-damper mode or custom control using user-provided Lua script attached to the actuator, which is beneficial for precise modeling of the actuator dynamics, or custom control rules. In our case, the real actuator is composed of the motor and reduction gear which dynamics can be described by the equation

$$J\ddot{q} + B\dot{q} + F(\dot{q}) + R\tau = K\,V, \tag{1}$$

where q is the rotor position angle before reduction, J is the rotor inertia, B is the rotor damping, F is the sum of static, dynamic and viscous friction, R is the gearbox ratio, τ is the servomotor torque, K is the back electromotive force, and finally V is the motor voltage. Although a precise model of the servomotor can be obtained by the identification of the parameters, the most influencing parts of the model are the parameters of the motion controller and frictions, which can be directly modeled in V-REP as a PID controller and link frictions, respectively. Therefore, there is no need for writing a custom actuator control script as the same behavior can be achieved by a proper parametrization of the existing solutions already provided by the simulator.

In particular, the PID controller has been parametrized according to the real servomotor as $P = 1, I = 0, D = 0$ and the inner friction of the joint has been simulated by setting the friction and angular damping in the material properties for the Bullet ver. 2.83 engine for the links attached to each joint.

The fact that the motor provides only the position feedback and uses the P-type controller allows us to estimate the static torque according to the servomotor documentation. In the static case, the position error e between the desired and current positions of the actuator is proportional to the joint torque τ according to Fig. 3a. The stall torque $\tau_{stall} = 1.5$ Nm is reported by the actuator manufacturer, and it is used as a hard limit for the torque values in our dynamic locomotion experiment.



(a) (b)

Fig. 3. (a) The relation of the torque τ and the position error e of the Dynamixel AX-12A servomotor. The value of τ is limited by a stall torque value τ_{stall}. (b) The relation of the torque and the position error for the simulated actuator.

The real behavior under the considered simplifications has been verified by the following experiment. The femur link of a single leg has been made static in the simulation, and the tibia link has been set with a large mass. Then, the actuator has been commanded to swing between the boundary positions which does not make either of links move, but verify the studied behavior. The result is visualized in Fig. 3b that complies with Fig. 3a. A small offset of the torque values is most likely caused by the experimental setup, where it is not possible to mitigate the effects of the gravity acting on the dynamic link. Further experiments verifying the precision of the developed static and dynamic simulation are detailed together with our experimental scenario on the parametrization of the locomotion gait in Section 4.

3.3 Notes on the V-REP Simulation

In the synchronous operation mode, the V-REP simulator operates by advancing the simulation time with the constant time step triggered by the user. Significant attention has to be paid to the proper selection of the simulation step to cope with the real world operation and sensor communication. Specifically, the Dynamixel AX-12A servomotor allows setting a new position to all actuators once every 1 ms whereas a single read takes 1 ms as well. Therefore, we have selected 1 ms as a base simulation step with the same restrictions on the communication. Thus, bulk reading has to be done exactly in the same order in the simulation as on the real robot with the underlying calls for the simulation step.

4 Experimental Results on Verifying the Realistic Simulator

In this section, three experimental scenarios are described that have been designed to verify the realistic behavior of the developed simulation. Each report on the achieved experimental results contains both real experiments and simulated results, and their comparison. The selected verification scenarios are (i) the static torque analysis experiment that further verifies the torque-position error relation; (ii) the dynamic leg movement experiment that verifies the overall performance of the dynamic simulation; and (iii) an experimental deployment in searching for a parametrization of the locomotion controller that supports the overall deployability of the developed model in a real-world robotic task.

4.1 Torque Analysis Experiment

In this experiment, three legs of the hexapod robot are lifted to create a support polygon in the form of a triangle as it is shown in Fig. 4. Subsequently, the angle of the tibia joint on the supporting middle leg goes from $-\pi/4$ to 0, and thus gradually transfers a larger part of the robot body weight to this leg which increases the torque τ on the leg joints. The experiment has been executed and analyzed in a quasi-statically setup for which the motion is sufficiently slow to consider each robot state to be static. The experiment has been done three times with different weights of the robot trunk. Namely without the battery pack, with a single battery pack, and with two battery packs, see Table 1 for the corresponding weights. The mass of the trunk has been adjusted in the simulation accordingly. The ground truth for the torque values τ_{real} has been calculated as the torque given by the distance of the leg endpoint to the joint axis multiplied by the weight applied on the legs endpoint measured by the table scale. The simulated torque τ_{sim} has been read directly from the simulator.

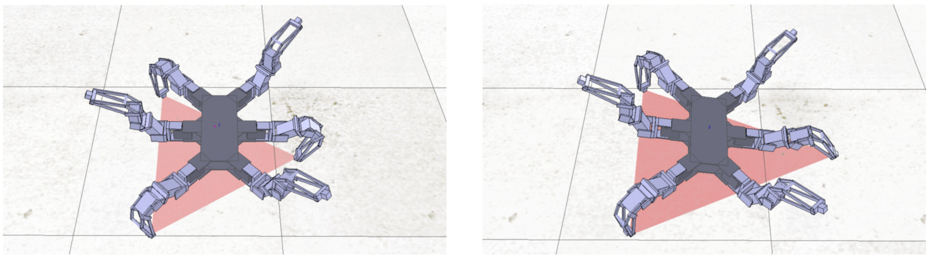
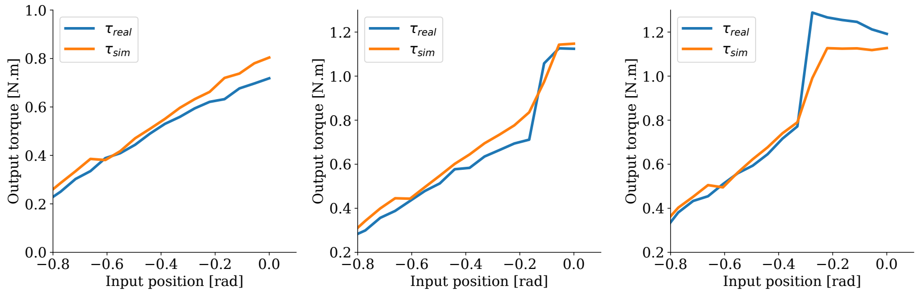

(a) (b)

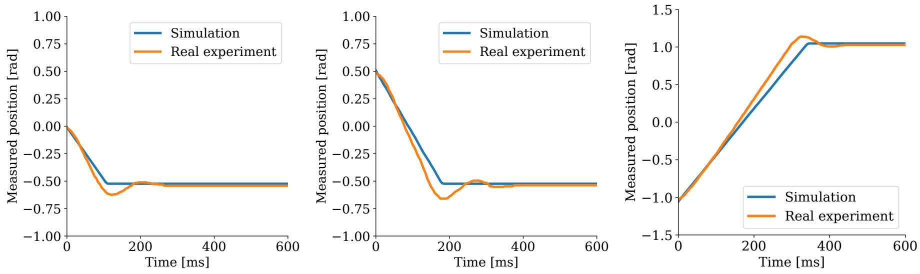
Fig. 4. The used setup in the torque analysis experiment. (a) Starting position and (b) end position.

The achieved results are visualized in Fig. 5a for the unloaded robot. Fig. 5b and Fig. 5c capture torques in the same experiment with the robot loaded with one and two battery packs, respectively. In the trials with the increased body weight, the servomotors went over the stall torque limit and turned off, which can be identified as a plateau in the plots, which is visible for both the simulation and real-world experiments. The experimental results indicate that the simulation can provide sufficiently precise estimations of the joint torque values.



(a) (b) (c)

Fig. 5. Real torque values τ_{real} and the simulated torque τ_{sim} for the tibia joint and (a) unloaded robot; (b) the robot body loaded with additional 330 g; and (c) the robot body loaded with 660 g.



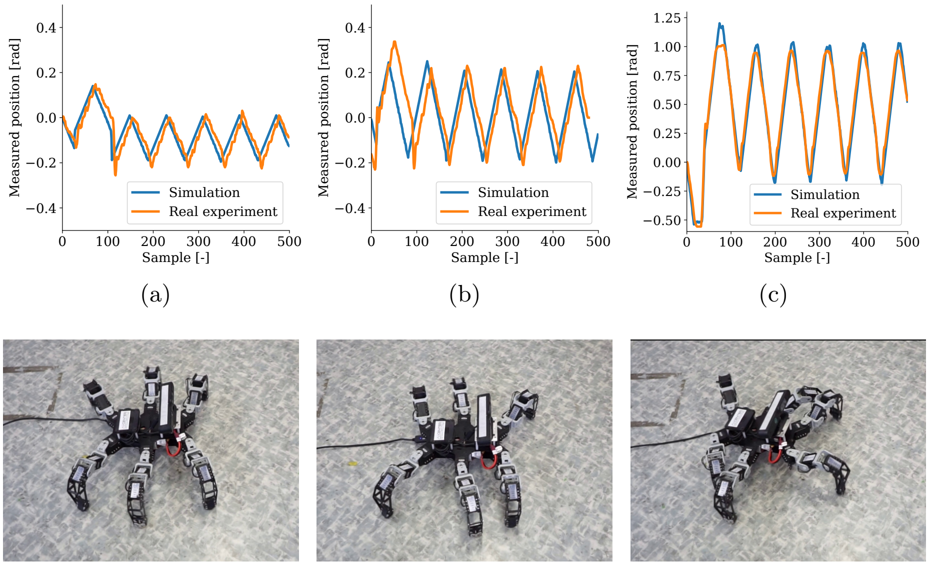
(a) (b) (c)

Fig. 6. The position setting on (a) coxa, (b) femur, and (c) tibia joint.

4.2 Dynamic Leg Movement Experiment

In the case of quasi-static movement, the relationship between the joint position error and the torque visualized in Fig. 3a holds. However, in the case of dynamic movement, further factors start to influence the torque at the individual joints, e.g., velocities and moments of inertia. Therefore, we designed an additional experiment for analyzing the dynamic movement of a single leg as follows.

The robot has been enabled a free movement of the legs in both simulator and real experiments. A single actuator has been set with a new position followed by the immediate consecutive position readings from that particular actuator. As it takes longer than 1 ms to reach the desired position, we have obtained a swing profile of the actuator. Plots in Fig. 6 represent a comparison between the behavior of the model and real joint. The results indicate that the overall shape of the motion is correct (for the used joint speed) and the results also indicate a correct setting of the P-type controller. However, the simulation does not cover the dynamic overshoots probably due to the setting of high friction in the individual links. Unfortunately, we have not been able to overcome this issue.



(a) (b) (c)

(d) (e) (f)

Fig. 7. The dynamic locomotion scenario. The trajectory of the coxa joint in (a) 2 times speeded-up locomotion, (b) 5 times speeded-up locomotion and (c) 6 times speeded-up locomotion for which self-collisions are already visible in the collected data. (d–f) the respective snapshots of the robot in the real experiment.

4.3 Dynamic Locomotion Experiment

Finally, the usability of the derived model has been verified in the dynamic locomotion control experiment. The model is employed in the problem of parameter search of the dynamic locomotion controller [14] which is a variant of the central pattern generator (CPG) based on an artificial neural network that produces a rhythmic pattern used as the input signal to the actuators.

A pre-learned CPG network has been utilized that is parametrized by the stride length of the gait cycle which directly influences the robot forward velocity. A greedy search has been performed to find a particular stride length value that maximizes the forward velocity of the robot but also satisfies the given restrictions on the maximum joint torques and self-collision free execution. For each examined parameter value, the locomotion has been performed for 10 s in the V-REP simulator. During the simulation, self-collisions have been monitored together with the torque values τ_{sim}. The found value provides five times speedup of the locomotion in comparison to the default setting with locomotion speed $v = 0.05 \mathrm{m\,s^{-1}}$ given by [14] for which the robot parts do not collide, and the torque does not exceed the safe value for continuous robot operation of $0.75\tau_{stall}$.

The comparison of leg trajectories for the coxa joint with two times speedup, five times speedup, and six times speedup, where self-collisions are already visible in both the simulated and real data, are visualized in Fig. 7. Besides, when the speedup is higher than the estimated parameter, the individual servomotors on the robotic platform start to overheat. Therefore, the performed experiment supports the usability of the developed simulation and its deployability in planning and optimization of the locomotion control for the hexapod walking robot.

5 Conclusion

This paper reports on the development of the realistic simulation model of the hexapod walking robot in a robotic simulator. The developed model provides a high-fidelity simulation of the robot dynamics and demonstrates to be useful in the problem of optimizing parameters of the locomotion control gait to maximize the robot velocity and avoid high torques and self-collisions, which might damage the actuators and the robot platform itself. The developed simulation provides us with a safe, fast, high-fidelity, and inexpensive tool for optimization and development of locomotion controllers. In the future, we would like to automate the search for the hexapod model parametrization that can be especially beneficial in long-term autonomy missions, when the robot morphology may change. We also aim to consider transfer learning techniques to support the usability of the simulation results among different robotic platforms.

Acknowledgement

This work has been supported by the Czech Science Foundation (GAČR) under research Project No. 18-18858S.

References

1. Belter, D., Labecki, P., Skrzypczynski, P.: Adaptive motion planning for autonomous rough terrain traversal with a walking robot. Journal of Field Robotics **33**(3), 337–370 (2016)
2. Čížek, P., Masri, D., Faigl, J.: Foothold placement planning with a hexapod crawling robot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4096–4101 (2017)
3. Collective of authors: Vortex Simulation Software, `https://www.cm-labs.com/`, cited on 2018-07-30
4. Erez, T., Tassa, Y., Todorov, E.: Simulation tools for model-based robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 4397–4404 (2015)
5. Erwin Coumans et al.: Bullet Physics Library, `http://bulletphysics.org/wordpress/`, cited on 2018-07-30
6. Freese, M., Singh, S., Ozaki, F., Matsuhira, N.: Virtual robot experimentation platform V-REP: a versatile 3D robot simulator. In: International Conference on Simulation, Modeling, and Programming for Autonomous Robots. pp. 51–62. Springer (2010)

7. Gao, H., Jin, M., Ding, L., Liu, Y., Li, W., Yu, X., Deng, Z., Liu, Z.: A real-time, high fidelity dynamic simulation platform for hexapod robots on soft terrain. Simulation Modelling Practice and Theory **68**, 125–145 (2016)
8. J. Jerez, A. Suero et al.: Newton Dynamics, `http://newtondynamics.com/forum/newton.php`, cited on 2018-07-30
9. Rohmer, E., Singh, S.P., Freese, M.: V-REP: A versatile and scalable robot simulation framework. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1321–1326 (2013)
10. Roy, S.S., Pratihar, D.K.: Dynamic modeling, stability and energy consumption analysis of a realistic six-legged walking robot. Robotics and Computer-Integrated Manufacturing **29**(2), 400–416 (2013)
11. Russel L. Smith et al.: Open Dynamics Engine, `http://www.ode.org/`, cited on 2018-07-30
12. Soyguder, S., Alli, H.: Kinematic and dynamic analysis of a hexapod walking running-bounding-gaits robot and control actions. Computers & Electrical Engineering **38**(2), 444–458 (2012)
13. Spis, M., Matecki, A., Maik, P., Kurzawa, Adamand Kopicki, M., Belter, D.: Optimized and reconfigurable environment for simulation of legged robots. In: Automation 2017. pp. 290–299. Springer International Publishing (2017)
14. Szadkowski, R.J., Čížek, P., Faigl, J.: Learning central pattern generator network with back-propagation algorithm. In: Proceedings ITAT 2018: Information Technologies - Applications and Theory. pp. 116–123 (2018)
15. Wensing, P.M., Wang, A., Seok, S., Otten, D., Lang, J., Kim, S.: 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 (2017)
16. Zaratti, M., Fratarcangeli, M., Iocchi, L.: A 3D simulator of multiple legged robots based on USARSim. In: RoboCup 2006: Robot Soccer World Cup X. pp. 13–24 (2007)