

Calibration Method to Improve Transfer from Simulation to Quadruped Robots

Gabriel Urbain, Alexander Vandesompele, Francis wyffels, and Joni Dambre

IDLab, Electronics and Information Systems Department, Ghent University imec,
Ghent, Belgium

`gabriel.urbain@ugent.be`

Abstract. Using passive compliance in robotic locomotion has been seen as a cheap and straightforward way of increasing the performance in energy consumption and robustness. However, the control for such systems remains quite challenging when using traditional robotic techniques. The progress in machine learning opens a horizon of new possibilities in this direction but the training methods are generally too long and laborious to be conducted on a real robot platform. On the other hand, learning a control policy in simulation also raises a lot of complication in the transfer. In this paper, we designed a cheap quadruped robot and detail a calibration method to optimize a simulation model in order to facilitate the transfer of parametric motor primitives. We present results validating the transfer of Central Pattern Generators (CPG) learned in simulation to the robot which already give positive insights on the validity of this method.

Keywords: Embodiment, Compliant Quadruped Locomotion, Transfer Learning, CPG

1 Introduction

The progress in *machine learning* within the last decades represents a great opportunity for improving specific performance metrics in the robotics state-of-art [4]. Among possible applications, locomotion constitutes an excellent example as deterministic controllers are not well adapted to unknown terrains and sensitive to external perturbations. Encouraging results involving complex gait control policies using rich sensor information to increase adaptiveness and robustness in an unknown dynamic environment have been introduced in the past years. Among the most influential contributions, *deep reinforcement learning* techniques have shown the most important improvements [12, 17]. However, the works are mainly conducted in simulation, whereas the controllers embedded on state-of-the-art robots are still mostly using sophisticated analytic methods with explicit knowledge of the robot physical parameters, e.g in [2, 16, 18]. Indeed, direct learning on a robot presents several drawbacks, among which the training time –which can be generally sped up in a physics simulation–, the wearing and hysteresis of mechanical parts, but also potential damages when the robot is exploring its own motor control capacities.

Partial training of policies in simulation is a straightforward idea to obtain a good initial state and avoid mechanical damage before transfer to a real robot. This field, known as *transfer learning*, involves big challenges as a lack of accuracy and realism during the simulation can easily lead to a failure in the real environment. However, simply using a more realistic simulator shifts the problem toward a sophisticated implementation of the physics engine and the need of heavier computational resources. Despite efforts in this direction, there is still a huge reality gap with the current simulators. Therefore, transfer in robotics has mainly focused on making controllers robust to changing environments. Among the recent related work, [5] shows how a quadruped locomotion gait, pre-trained in one environment, can learn faster on other surfaces. For higher-level architectures and learning techniques, [6] presents a new neural network architecture for transfer of *reinforcement learning* with various robotic tasks and [13] focuses on locomotion using a hierarchical bio-inspired control architecture combining recurrent networks for motor primitives using proprioceptive feedback together with higher level feed-forward networks also processing visual information.

The *embodiment* theory, and in particular *morphological computation* in its larger sense, i.e. as presented in [8] may be a preliminary solution to theorize the transfer problem. In this framework, body, controller, and their intricate relation are analyzed from a dynamical perspective. Each entity can be modeled as a non-linear filter with computational skills rather than explicitly in a kinematic parameter space. From that point of view, a good simulation works with an accurate representation of the transfer functions rather than a detailed physics implementation. This can be obtained through automated optimization rather than fine-tuning of model parameters. Such an idea has been used for instance with quadruped robots that refine their own body representation by matching real and simulated sensors inputs [3]. With the trends in data-driven learning techniques, an increasing number of studies also consider the simulated physics as a black box with parametric equations, for instance using neural networks [15]. However, this creates a problem with respect to interpretability and the parametric calibration approach presented in this paper using a standard physics engine and a parametric model represents a good compromise.

This paper introduces an automated calibration method for a simulation model that enables optimal transfer of a controller to a mechanical platform. It takes inspiration in *morphological computation*: rather than trying to replicate rigorously the robot physics, we optimize a parametric model to maximize the similarities between simulation and real world of the body sensor-to-actuator transfer function. In the next section, the design and optimization methods are discussed in further detail. The third section shows the results of two experiments: the first concerns the calibration method and the second evaluates the performance of an open-loop controller trained in simulation and transferred on the real robot. Finally, conclusions and perspectives on the next research steps are given in the last section.

2 Methods

2.1 The Robot

Mechanical Design The platform used in this project is an update of the *Tigrillo* robot [21] shown in Figure 1. It implements four under-actuated knee joints, whose kinematic constraints can be manually tuned using detachable springs and dampers to tune the passive compliance properties. A special effort was applied to make it low-cost, versatile and reproducible. The robot weighs 950 g and fits in a box of 30 cm by 18 cm. The track widths are 15 cm in the front and 11 cm between the hind legs and the distance between front and back measures 16 cm, providing a stable balance with any configuration of slow gaits. The legs are directly coupled to four *Dynamixel RX-24F* servomotors selected as a compromise between weight, torque and fast rotation speed.



Fig. 1: On the left, the quadruped *Tigrillo* robot used in this paper for experiments on calibration and transfer of control. On the right, the corresponding parametric simulation model in *Gazebo*.

The use of passive compliance in the knees in place of rigid constraints has experimentally shown a decrease of the optimal cost of transportation but also helps to obtain smoother behaviors in the overall locomotion process [21]. The same leg principles have been also presented in research involving robots like Tekken [9], Puppy [1] or Bobcat [14].

Electrical Design Following the same constraints on reproducibility and cost, the electronics stack is made from three off-the-shelf boards. First, a DC step-up voltage converter supplies the other boards and motors with a 12 V regulated voltage and a stalk current that can rise to 10 A when the legs are pushing together and the motors have to deliver a high torque. Secondly, an *OpenCM* board is used to read the analog sensor values and send the position or velocity commands to the servomotors. The computer board is a *Raspberry Pi 3* running the *Robot Operating System (ROS)*¹ and streaming actuation and sensor signals to a computer over *ROS* topics.

¹ <http://wiki.ros.org/>

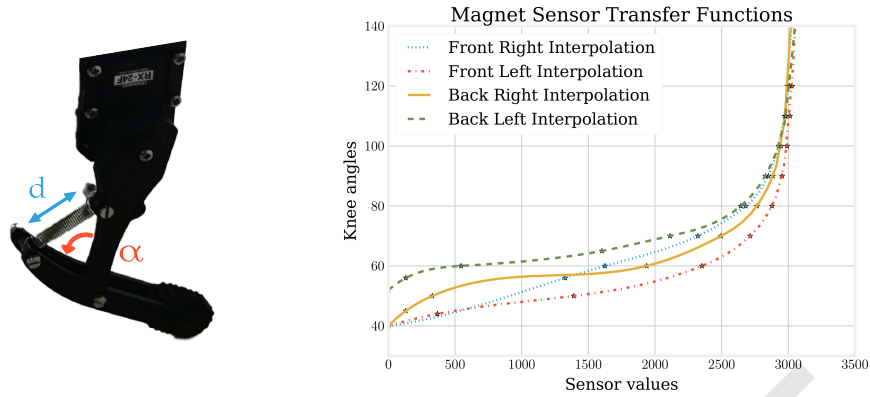


Fig. 2: On the left, a close-up picture of the leg shows the direct coupling with the *Dynamixel* motor and the non-actuated knee joint embedding the angle sensor. On the right, the four sensor calibration functions are represented. Differences in offset and shape of the curves come from the manual fixation of the magnet and the Hall sensor.

Angle Sensors One important aspect regarding the application of the morphological computation theory is to extract nonlinear feedback from the robot compliant morphology [11]. To this end, permanent rare-earth magnets are attached to the lower parts of each leg and analog *Hall sensors* to the higher parts. The setup presents an advantage in cheapness and does not obstruct the joint movement. This sensor is principally made of a semi-conductor triggered by the Hall effect to output a voltage between 0 V and 5 V proportionally to the surrounding magnetic field \mathbf{B} . As a first approximation, this field is decreasing with the cubic value of the distance to the magnet d . This length varies with the square root of the cosine of the knee angle α , according to the generalized Pythagorean theorem. As a consequence, the variation of the sensor voltage is high for small angles and the order of magnitude for the best accuracy is around a hundredth of a degree. It however decreases quickly with the knee angle to reach approximately one degree when the legs are fully extended. For each leg, a conversion table between the sensor value and the measured angle is used to interpolate the transfer function of the sensor presented in Figure 2. All the curves have a shape conform to the expected aspect but the manual fixation of the magnet and the Hall sensor leads to a different offset for each leg.

2.2 Calibration of the Simulation Model

The approach followed in the calibration method consists in tuning a set of simulation parameters in order to optimally match the simulated body's response to the real robot's response as observed by the sensors, when an actuation signal

is applied. The simulation is performed in the *Neurorobotics Platform (NRP)* [7] using *Gazebo* with the *ODE* physics engine. We select the set of parameters θ both for their importance in the locomotion behavior but also because they are harder to measure or model accurately:

$$\theta = \{k_f, k_h, \mu_f, \mu_h, d_f, d_h, m_{\text{dist}}, \}. \quad (1)$$

In this notation, the index f refers to the front legs of the robot and h refers to the hind legs. The parameters k are the spring constants in the knee joints, expressed in N/m. These parameters are chosen because the complexity of the spring models in the physics engine is rather limited and can lead to insufficient results, as discussed further in the Results section. The parameters d are the contact depth coefficients expressed in m and represent how much two rigid bodies can overlap during simulation to compute the friction forces. They directly interfere with the static friction coefficients μ and a good manual tuning generally requires empirical comparison with the real robot. Finally, the total mass of the robot is fixed and determined by weighting the robot but the distribution ratio between the front and the back is represented with m_{dist} . Many other parameters like the damping values in the knee joints, the minimal value of knee angles, the kinematic friction coefficients of the feet and the motor characteristics parameters have been evaluated in this research but none of them has shown significant improvements with respect to the uncalibrated model. Therefore, they are not discussed further in this paper. Figure 3 shows the architecture of the

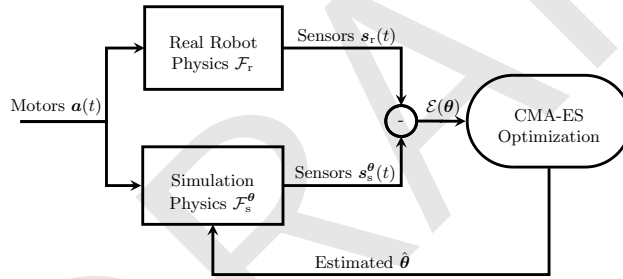


Fig. 3: The calibration is represented on this diagram. Sensor values recorded on the robot are used to optimize the unknown morphology parameters of the simulation model with *CMA-ES*.

optimization process used for calibration. The robot is actuated in open-loop with four sinusoidal signals $\mathbf{a}(t)$, with the same amplitude and phase for all legs such that its body will alternate between standing-up and sitting-down. The sensor values of the robot $\mathbf{s}_r(t)$ are recorded to estimate the robot transfer function:

$$\mathbf{S}_r(p) = \mathcal{F}_r(\mathbf{A}(p)), \quad (2)$$

where $\mathbf{A}(p)$ and $\mathbf{S}_r(p)$ are the Laplace transforms of the robot motor and sensor signals $\mathbf{a}(t)$ and $\mathbf{s}_r(t)$.

An optimization is performed with *covariance matrix adaptation evolutionary strategy (CMA-ES)*, as formulated in [10]. It has the advantage to converge rapidly in a landscape with several local minima and requires only few initialization parameters. The algorithm generates simulation models with different sets of parameters and estimates the one that minimizes the error:

$$\hat{\theta} = \arg \min_{\theta} \mathcal{E}(\theta). \quad (3)$$

The error function $\mathcal{E}(\theta)$ is chosen to represent the difference between two temporal signals but shall also allow invariance against a possible slight phase shift. This invariance can be obtained by computing the *Mean Absolute Error (MAE)* for different configurations where the robot sensors signal $\mathbf{s}_r(t)$ and the simulation sensors signal $\mathbf{s}_s(t)$ are shifted within a time window, and finally taking the minimum error:

$$\mathcal{E}(\theta) = \min_{k \in [-M, M]} \frac{1}{N} \sum_{i=k}^{N+k} \text{abs}(\mathbf{s}_{r,i} - \mathbf{s}_{s,i}^{\theta}), \quad (4)$$

where N is the number of samples measured for the two signals during a period T , and M is the boundary of the shifting window. The errors for each sensors are simply added to obtain the final total error value.

Finally, it is assumed that the parameter set that is obtained in the optimization corresponds to a good approximation of the robot transfer function:

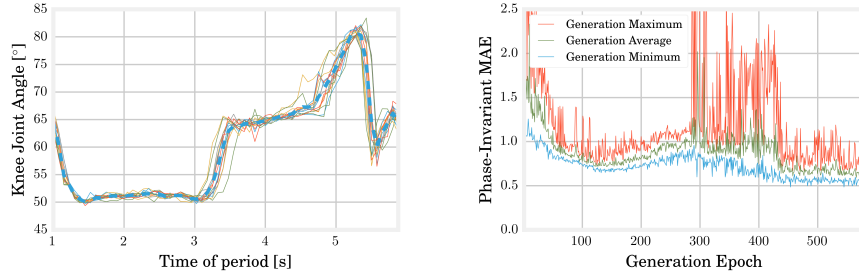
$$\mathcal{F}_s^{\hat{\theta}} \approx \mathcal{F}_r. \quad (5)$$

In order to decrease the noise on the sensors, which is caused by different factors like some electro-magnetic perturbations or the undesirable residual vibration coming from the motors, the signal is segmented using the zero-crossing of the periodic actuation signal as a threshold. All the sensors segments are then projected on a one-period domain with a fixed number of points to be easily averaged and compared (see Figure 4a).

3 Results

3.1 Calibration of the Simulation Model

To apply the calibration method in practice, the robot is actuated by a 0.3 Hz sine wave during one minute and the real motor signals and the sensor signals are recorded during the last 50 seconds, to provide a reference for the optimization. After this step, a *CMA-ES* algorithm is run on a computer. Different initialization values were tested but a low variance was observed in the results. For each algorithmic epoch, a population of 10 individuals is generated: each of them has a different morphology sampled in the distribution of θ provided by *CMA-ES* and is simulated with the same actuation pattern as the real robot. The optimization evolution presented on Figure 4b shows a saturation of the *MAE* error to 0.5 after more than 500 generations.



(a) **Front Left Sensor Period Average.** The sensor signals recorded on the robot are segmented periodically to obtain a smoother average for scoring the optimization.

(b) **CMA-ES Evolution.** The evolution of the *CMA-ES* algorithm through the successive generations converges a first time around the 100th generation then move again to find a better minimum at the 500th generation.

Fig. 4: An overview of the optimization method evolution.

To validate qualitatively this optimum, the *Neurorobotic Platform (NRP)* is used to control together the simulation model and the real robot. A variety of actuation sinusoids with different amplitudes are applied on the motors. A visual comparison indicates a good correlation between the simulation and the real observations². Interestingly, failures like stumbling or falling are also observed simultaneously in both worlds which is very useful as the role of pre-training in simulation is to exclude actuation patterns that lead to instability of the physical robot.

To explain these results, a single sine wave of 2 Hz –the same as during the optimization– is given to the motors. The average sensor signals over one period are plotted for the real robot, the optimum morphology in simulation and the initial default morphology in simulation (see Figure 5a). The latter model is made with our best knowledge of the robot mechanical parameters and simulator parameters and serves as a baseline. A comparison of the parameters before and after optimization is also given in Table 1. It is quite obvious on the figure that the signals from the optimized morphology are closer to the observation on the robot. This has certainly two major causes. First, the default parameters are unable to reproduce correctly the friction forces between the feet and the ground. This effect depends both on the friction coefficient μ but also the contact depth between two rigid bodies in simulation, which has no concrete meaning in the physical world. During testing, when the robot lifts its body off the ground, the *Center of Gravity (CoG)* move forward above the supporting point of the front feet. The feet suddenly slide on the ground and the knee angle value drops before to increase again when the robot moves back to the

² <https://youtu.be/CqpkC630fJA>

ground and the *CoG* is shifted backwards. This phenomenon is observed on both the real robot and the optimized model but not with the default parameters. Another reason concerns the spring stiffness, which seems too low for the default morphology although it has been fixed by actual measurement on the robot. This is induced by the limitation of the simulator which only allows to model the physical spring via an equivalent torsional spring linear model in the joint. For low spring values, the model is quite accurate but for higher values, the non-linearities induce a saturation on the real robot which cannot be simulated consistently. The optimization copes with this problem by converging on a larger stiffness in the hind legs as displayed in Table 1.

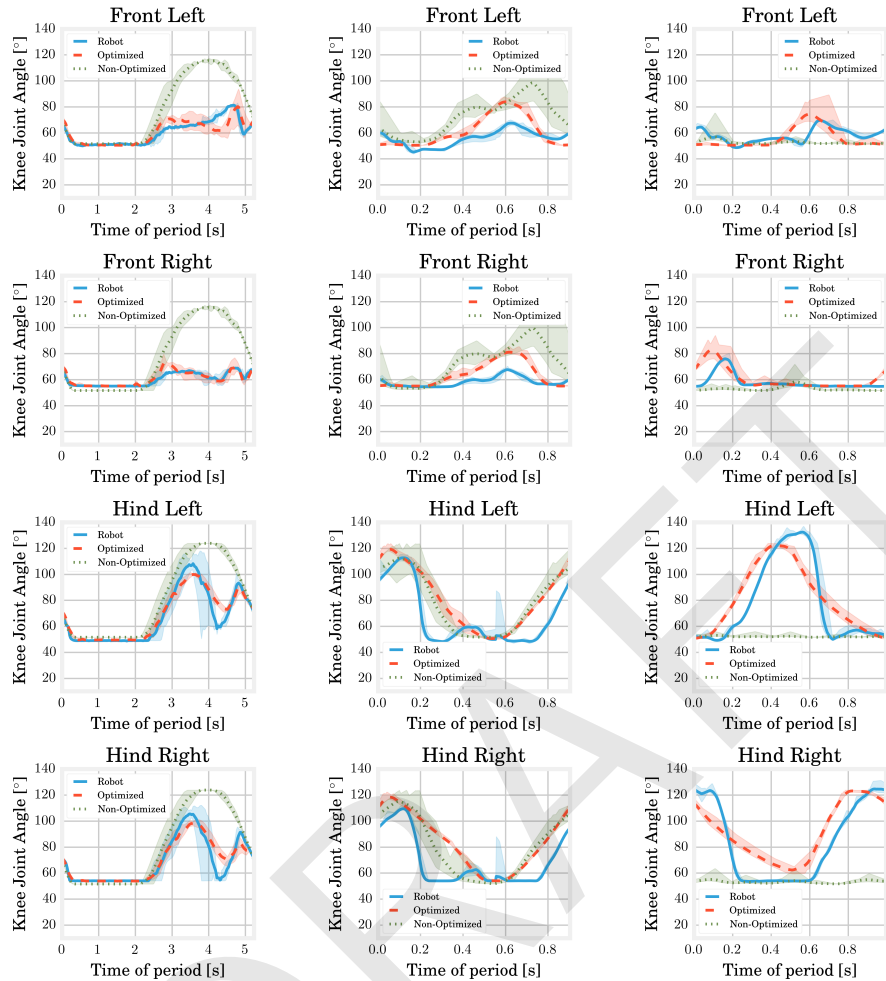
Morphology Parameters	Non-Optimized	Optimized
Hind Contact Depth Coefficient (mm)	0.5	7.8
Hind Friction Coefficient μ	0.1	0.000819
Hind Spring Stiffness (N/m)	181.6	440.629
Hind Mass (kg)	0.238	0.276
Front Contact Depth Coefficient (mm)	0.5	3.8
Front Friction Coefficient μ	0.1	0.283
Front Spring Stiffness (N/m)	181.6	181.4691
Front Mass (kg)	0.712	0.674
Mean Absolute Error	3.231	0.483

Table 1: This table shows the calibrated morphology parameters before and after optimization. If the mass distribution and the front spring stiffness does not change significantly, the hind knee joints become much stiffer to counteract the spring saturation on the real robot and the friction with the ground decreases in the front and increases in the back to render correctly the general movement.

3.2 Validation with Open-loop Gaits

In order to corroborate the preliminary observations on the calibration method described in the last section, different open-loop controllers are trained in simulation and we observe how the transfer performs on the real robot.

The controller is modeled by four coordinated *CPGs* with the equations introduced in [19]. Constraints on the frequency and the phases between the different legs are added to obtain motor primitives for walking and bounding gaits at 1 Hz. The other *CPG* parameters are obtained with a *CMA-ES* optimization with the optimized simulation model and where the robot speed is used as a score. This optimization has been successfully conducted in previous research [20] and helps to explore the motor space to find the most stable gaits to locomote along



(a) **Optimization.** The signal is closer to the real robot after optimization. This comes from stiffer springs in the hind legs and a different friction that allows more realistic trajectories.

(b) **Bounding gait.** The optimized model is closer to the real robot during the bounding gait. However, it still fails to reproduce correctly the angle variation speed.

(c) **Walking gait.** For the walking gait, the optimized signal gives good results despite a phase shift due to communication delays. However, the non-optimized model directly falls with this gait.

Fig. 5: Sensor signals of the real robot (blue), the optimized simulation model (red) and the default model (green) during optimization, bounding gait and walking gait.

a straight line. The resulting controller does not make use of sensor inputs and only the actuation and the robot mechanical properties are discussed hereafter. Some videos can be also found online³ to get a better understanding of the locomotion behavior.

Bounding gait To come up with a bounding gait, a constraint is added on the *CPGs* phases such that the hind legs are going through stance and swing phases synchronously and the front ones as well. This gait is inherently very stable as it does not involve any movement perpendicular to the walking direction. Figure 5b compares the value of the knee angle on a the real robot as well as the simulation default and optimized models. The optimization introduces better value ranges for the legs but also much more homogeneous results as highlighted by the larger variance of the non-optimized signal. However, it seems that the simulation fail to render the fast transition measured on the robot for both models. In future work, the optimization process should involve higher frequencies to solve this issue.

Walking gait These gaits are not especially stable as no efforts were made to optimize the robot’s balance in the lateral axis and the robot does not allow feet retraction during locomotion. However, some correct gaits are obtained by optimizing the *CPG* parameters whilst setting the same phase for the legs diagonally opposed. In leads to good performance on the real robot, and the knees angles outlined by the blue curve on Figure 5c are correctly simulated with the optimized robot model. A phase shift of approximately 100ms can be explained by the delay introduced by the communication line and the motor inertia. No results could be collected on the non-optimized model for this specific gait, as the robot directly felt on the ground in the simulator. This is not surprising as the controller was optimized using the calibrated model but indicates a good correlation between optimized model and real robot that does not exist with the default model.

4 Conclusions and Future Work

This paper investigated a method to calibrate the simulation model of a cheap passive compliant quadruped robot called *Tigrillo*, in order to transfer efficiently motor primitives learned in simulation.

To derive a usable model of the mechanical platform, it was not chosen to tune them manually but rather to use morphological computation framework to calibrate a parametric simulation model with optimization techniques. From the first observations, this method resulted in a simulation model that can achieve better performance. It also reduces the amount of knowledge and hypothesis that have to be given in the design process. This method has enabled the transfer of parametric *CPGs* for walking and bounding gaits trained in simulation to the

³ <https://youtu.be/zCHRWxfoOMU>

real platform while keeping a realistic behavior compared to the observations in the simulated environments.

In further work, this approach should be generalized for a larger range of actuation frequencies to get a more significant optimization score. Also, the transfer could be characterized against other relevant metrics like robot trajectory, pitch or foothold pattern. Finally, the training and transfer of closed-loop motor primitives should also be investigated on the *Tigrillo* platform to enable research on higher-level models.

Acknowledgments

This research has received funding from the European Unions Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No. 720270 (Human Brain Project SGA1).

References

1. Aschenbeck, K.S., Kern, N.I., Bachmann, R.J., Quinn, R.D.: Design of a quadruped robot driven by air muscles. In: Biomedical Robotics and Biomechanics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on. pp. 875–880. IEEE (2006)
2. Barasuol, V., Buchli, J., Semini, C., Frigerio, M., de Pieri, E.R., Caldwell, D.G.: A reactive controller framework for quadrupedal locomotion on challenging terrain. In: 2013 IEEE International Conference on Robotics and Automation, ICRA. pp. 2554–2561 (2013)
3. Bongard, J., Zykov, V., Lipson, H.: Resilient machines through continuous self-modeling. *Science* 314(5802), 1118–1121 (2006)
4. Connell, J.H., Mahadevan, S.: Robot learning, vol. 233. Springer Science & Business Media (2012)
5. Degraeve, J., Burm, M., Kindermans, P., Dambre, J., wyffels, F.: Transfer learning of gaits on a quadrupedal robot. *Adaptive Behaviour* 23(2), 69–82 (2015)
6. Devin, C., Gupta, A., Darrell, T., Abbeel, P., Levine, S.: Learning modular neural network policies for multi-task and multi-robot transfer. In: 2017 IEEE International Conference on Robotics and Automation, ICRA. pp. 2169–2176 (2017)
7. Falotico, E., Vannucci, L., Ambrosano, A., Albanese, U., Ulbrich, S., Tieck, J.C.V., Hinkel, G., Kaiser, J., Peric, I., Denninger, O., Cauli, N., Kirtay, M., Rönnau, A., Klinker, G., Arnim, A.V., Guyot, L., Peppicelli, D., Martínez-Cañada, P., Ros, E., Maier, P., Weber, S., Hubert, M., Plecher, D.A., Röhrbein, F., Deser, S., Roitberg, A., van der Smagt, P., Dillmann, R., Levi, P., Laschi, C., Knoll, A.C., Gewaltig, M.: Connecting artificial brains to robots in a comprehensive simulation framework: The neurorobotics platform. *Frontiers in Neurorobotics* 2017 (2017)
8. Fuchslin, R.M., Dzyakanchuk, A., Flumini, D., Hauser, H., Hunt, K.J., Luchsinger, R.H., Reller, B., Scheidegger, S., Walker, R.: Morphological computation and morphological control: Steps toward a formal theory and applications. *Artificial Life* 19(1), 9–34 (2013)
9. Fukuoka, Y., Kimura, H., Hada, Y., Takase, K.: Adaptive dynamic walking of a quadruped robot tekken on irregular terrain using a neural system model. In: 2003

- IEEE International Conference on Robotics and Automation, ICRA. vol. 2, pp. 2037–2042. IEEE (2003)
10. Hansen, N.: The CMA Evolution Strategy: A Comparing Review, pp. 75–102. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
 11. Hauser, H., Ijspeert, A.J., Fuchslin, R.M., Pfeifer, R., Maass, W.: The role of feedback in morphological computation with compliant bodies. *Biological Cybernetics* 106(10), 595–613 (2012)
 12. Heess, N., TB, D., Sriram, S., Lemmon, J., Merel, J., Wayne, G., Tassa, Y., Erez, T., Wang, Z., Eslami, S.M.A., Riedmiller, M.A., Silver, D.: Emergence of locomotion behaviours in rich environments. *CoRR* abs/1707.02286 (2017), <http://arxiv.org/abs/1707.02286>
 13. Heess, N., Wayne, G., Tassa, Y., Lillicrap, T.P., Riedmiller, M.A., Silver, D.: Learning and transfer of modulated locomotor controllers. *CoRR* abs/1610.05182 (2016), <http://arxiv.org/abs/1610.05182>
 14. Khoramshahi, M., Sprowitz, A., Tuleu, A., Ahmadabadi, M.N., Ijspeert, A.J.: Benefits of an active spine supported bounding locomotion with a small compliant quadruped robot. In: 2013 IEEE International Conference on Robotics and Automation, ICRA. pp. 3329–3334. IEEE (2013)
 15. Martius, G., Lampert, C.H.: Extrapolation and learning equations. *CoRR* abs/1610.02995 (2016), <http://arxiv.org/abs/1610.02995>
 16. Park, H., Wensing, P.M., Kim, S.: High-speed bounding with the MIT cheetah 2: Control design and experiments. *The International Journal of Robotics Research* 36(2), 167–192 (2017)
 17. Peng, X.B., Berseth, G., Yin, K., van de Panne, M.: Deeploco: dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Trans. Graph.* 36(4), 41:1–41:13 (2017)
 18. Raibert, M., Blankespoor, K., Nelson, G., Playter, R.: Bigdog, the rough-terrain quadruped robot. *IFAC Proceedings Volumes* 41(2), 10822 – 10825 (2008)
 19. Righetti, L., Ijspeert, A.J.: Pattern generators with sensory feedback for the control of quadruped locomotion. In: 2008 IEEE International Conference on Robotics and Automation, ICRA. pp. 819–824 (2008)
 20. Urbain, G., Degraeve, J., Carette, B., Dambre, J., wyffels, F.: Morphological properties of mass-spring networks for optimal locomotion learning. *Frontiers in Neurobotics 2017* (2017)
 21. Willems, B., Degraeve, J., Dambre, J., wyffels, F.: Quadruped robots benefit from compliant leg designs. Presented at the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (2017)