2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Modular Control of Limit Cycle Locomotion over Unperceived Rough Terrain

Mostafa Ajallooeian*, Sébastien Gay, Alexandre Tuleu, Alexander Spröwitz and Auke J. Ijspeert

***Abstract*— We present a general approach to design modular controllers for limit cycle locomotion over unperceived rough terrain. The control strategy uses a Central Pattern Generator (CPG) model implemented as coupled nonlinear oscillators as basis. Stumbling correction and leg extension reflexes are implemented as feedbacks for fast corrections, and model-based posture control mechanisms define feedbacks for continuous corrections. The control strategy is validated on a detailed physics-based simulated model of a compliant quadruped robot, the Oncilla robot. We demonstrate dynamic locomotion with a speed of more than 1.5 BodyLength/s over unperceived uneven terrains, steps, and slopes.**

I. INTRODUCTION

Design of legged locomotion controllers has always been a challenge. This is due to the fact that successful legged locomotion consists of many tasks including coordination of multiple degrees of freedom, balance control, dealing with the switching dynamics imposed by the discrete contacts, etc. The problem of locomotion control becomes even more challenging when the target environment is both *irregular* and *unperceived* (through external sensors like laser scanners). Here in this paper we focus on the design of controllers for dynamic locomotion of quadrupeds over unperceived rough terrain of medium difficulty. Therefore we will not extensively address the research about *static* locomotion control (e.g. [1], [2]), the ones which only has been tested on a *flat* terrain (e.g. [3], [4]), or the ones which are tested on *perceived* rough terrain, like the majority of the works done on the LittleDog [5] under the Learning Locomotion program [6].

One of the first successful attempts to dynamic locomotion with quadruped robots was the seminal work of Raibert et al. [7], [8]. Their control approach is based on dividing the locomotion control into three main subtasks: hopping control, speed control by adapting the step length, and posture control via adjusting the joint torques. Though Raibert's control was not extensively tested on unperceived rough terrain back in 80's, it has been extended and successfully used on robots like BigDog [9] for dynamic locomotion over unperceived rough terrain, however the details are not publicly disclosed.

There are also other locomotion control approaches applied to quadrupeds running on unperceived rough terrain. This includes the research done on the Tekken robot [10], [11] where a bio-inspired control approach consisting of pattern generators and reflexes is applied. Another example is the control approach presented by Maufroy et al. [12]. They utilize a Central Pattern Generator (CPG) [13] model enriched with phase modulations based on legs loading/unloading. They tested their approach on uneven terrain in simulation and on the Kotetsu robot facing lateral perturbations and steps.

*All the authors are with the Biorobotics laboratory, École Polytechnique Fédéral de Lausanne (EPFL), CH-1015, Switzerland. Corresponding author: Mostafa Ajallooeian. Email: mostafa.ajallooeian@epfl.ch

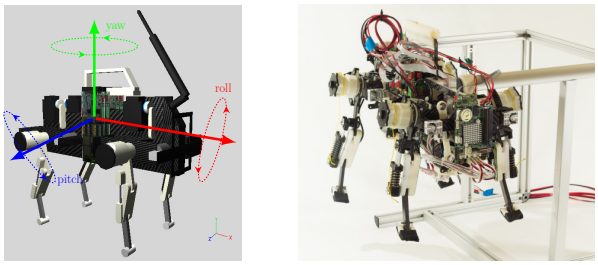


Fig. 1. Oncilla platform. The simulated robot (left) is a detailed model of the hardware robot (right). Legs are passively compliant, and implement pantograph mechanisms.

Moreover, there are the recent locomotion controllers based on the floating-based inverse dynamics control. This includes the control strategy used on the HyQ robot, which is based on inverse dynamics and virtual model control [14], and the operational space control on the StarlETH robot [15]. Both of these control approaches have been tested with robots running on a treadmill with occasional unperceived obstacles, on slopes, and against lateral perturbations.

Our main motivation here is to introduce a simple way to design controllers for quadruped locomotion over rough terrain. More precisely, we want our controller to have the following properties:

1) The controller should be modular and hierarchical. This means that the control should be divided into meaningful modules, and a lower level module should be able to work even in the absence of the higher level ones. Different modules should be tuned on top of each other, and should not be strongly interconnected. This fact will reduce the complexity of finding the right control parameters since they can be set sequentially.
2) Our target robots are comparatively cheap and lightweight robots. So the control approach should depend on as little sensory information as possible, and it should not be computationally heavy.
3) The controller should allow for dynamic and relatively fast gaits (at least more than 1 BL/s) over unperceived rough terrain of medium difficulty (BL: Body Length).

The first property distinguishes our desired control strategy from interconnected controllers like the ones on Tekken [10], [11]. The second property makes our desired control strategy different from approaches which strongly depend on

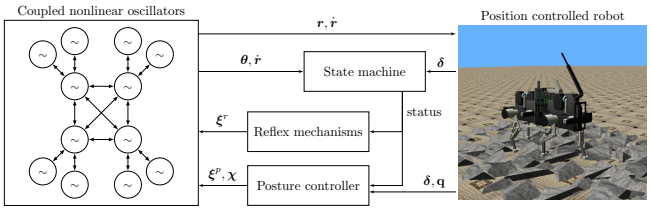


Fig. 2. The control architecture. Coupled nonlinear oscillators implement a CPG model as the basis. Reflex and posture control feedbacks affect the CPG states. A state machine controls the activation of the feedbacks. $\mathbf{r}$ is the CPG radial outputs controlling joint angles, $\boldsymbol{\theta}$ is the CPG phases, $\boldsymbol{\delta}$ is the on/off contacts, $\mathbf{q}$ is the sensed joint angles, $\boldsymbol{\xi}^r$ and $\boldsymbol{\xi}^p$ are reflex and posture control feedbacks, and $\boldsymbol{\chi}$ is the CPG radial offsets.

sensory information or rather heavy computation, e.g. inverse dynamics controllers like in [14], [15], These controllers need torque controlled robots which are equipped with torque and/or full contact sensing which, as of this moment, are not cheap. Finally the third property distinguishes our desired control strategy from control approaches like ZMP [16] ones, whose constraints prevent high speed locomotion.

To design a controller with the mentioned properties, we use Central Pattern Generators (CPG), implemented using coupled nonlinear oscillators, as the low level module for generating the locomotion patterns. We have shown in a recent study [17] that CPGs, even used in open-loop, if properly applied to a passively compliant quadruped, can lead to forward locomotion speeds up to 6.9 BL/s, equal to a froud number $fr = 1.3$. We believe that CPGs are good bases for fast locomotion.

We add reflex feedbacks to the oscillators to compensate for situations where a rapid correction is needed. We also add model-based posture control feedbacks to continuously adjust body rotations while traveling over rough terrain. As a result, we introduce a systematic way of designing feedback signals for Central Pattern Generator controllers as well.

We systematically test our control strategy on a simulated quadruped locomoting over unperceived rough terrain. This simulated quadruped is a detailed model of the Oncilla robot (Figure 1) which will be used for a full validation in near future. This paper is an extension of our previous study [18] on a stiff torque controlled simulated quadruped.

II. CONTROL METHODOLOGY

The modular controller introduced here uses a computational Central Pattern Generator (CPG) model as the core. CPGs have proven to be useful for limit cycle locomotion and has been widely used on different robots [10]–[12], [19], [20]. An open-loop CPG might suffice for flat terrain locomotion, but sensory feedback is needed to compensate for perturbations. We implement reflexes for fast corrections, and model-based posture control for continuous corrections, and both of these feedbacks affect the CPG states. An overall schema of this modular control strategy is depicted in Figure 2.

These modules will be detailed in the following sections. We need to mention that each leg of the robot has three *actuated* degrees of freedom (DOF), first the joint responsible for leg abduction/adduction (lateral hip joint), second for leg protraction/retraction (sagittal hip joint), and last for leg extension/flexion (sagittal knee joint). These joints will be addressed with AA, PR, and FE subscripts. The movements of the ankle joints are coupled to the knee joints (pantograph mechanism), and they are not directly actuated.

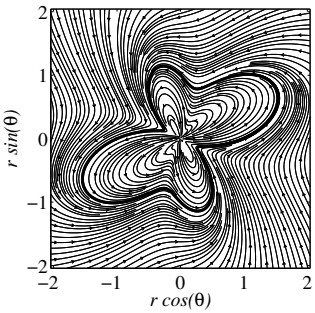


Fig. 3. An one-dimensional example phase portrait of a morphed amplitude controlled oscillator. The desired limit cycle is defined by $f(\theta) = \sin(3\theta)\tanh(\cos(\theta)) + 1$, $\gamma = 10$, and $\mu = 1$. The radial limit cycle is globally asymptotically stable, and the oscillator converges to the limit cycle from any state. Please note that the illustrated phase portrait is only for positive r values.

In total, there are $N = 12$ controllable DOF in the robot and $L = 4$ legs. Individual joints will be addressed with i and j subscripts, and legs will be addressed with l subscripts. So x_{PR_l} means the x state of the PR joint of the lth leg.

A. Central Pattern Generators

We use morphed oscillators to implement a Central Pattern Generator model. Morphed oscillators are nonlinear oscillators which can exhibit desired arbitrary limit cycle shapes defined as functions of phase. We employ coupled morphed amplitude controlled oscillators, defined as:

$$\dot{\theta}_i = \Omega_i \tag{1}$$

$$\dot{r}_i = \mu\Omega_i f_i'(\theta_i) + \gamma\left(\mu f_i(\theta_i) + \chi_i - r_i\right) + \xi^r + \xi^p \tag{2}$$

$$\Omega_i = \omega + \sum_{j=1}^{N} c_{ij}\sin(\theta_j - \theta_i - \phi_{ij}) \tag{3}$$

where θ_i, Ω_i and r_i respectively are the phase, the coupling dynamics, and the radial output of the ith oscillator. μ is the radius of the amplitude controlled oscillator, γ is the convergence rate, ω is the locomotion frequency multiplied by 2π, and c_{ij} and ϕ_{ij} are the coupling strength and phase difference between the ith and jth oscillators. $f_i(\theta)/\mu$ defines the shape of the limit cycle of the ith oscillator and $f_i'(\theta) = \partial f_i(\theta)/\partial\theta$. $f_i : \theta \mapsto r$ can be any arbitrary C^1-differentiable function of phase.

r_i is the joint angle reference for the ith DOF, χ_i is an additional feedback offset added to the reference, and ξ^r and ξ^p are reflex and posture control angular velocity feedbacks (sections II-B and II-C).

The radial limit cycles of these oscillators are globally asymptotically stable (phase is indifferent). With non-negative c_{ik} values and consistent phase differences, these oscillators always converge to the desired phase differences and the desired limit cycle, even facing (finite-time) perturbations (see Appendix I for a brief proof). This fact eases the process of feedback integration and ensures stability. An example phase portrait is depicted in Figure 3.

B. Reflexes

Reflexes are crucial in cases where fast corrections are needed. There are two kind of reflexes that we address:

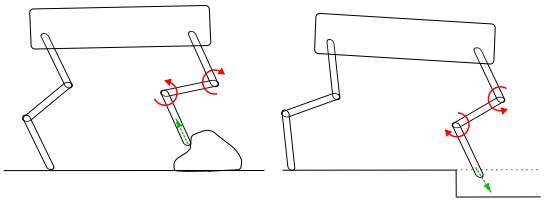


Fig. 4. Reflexes. Left) Leg hits an obstacle in the swing phase. A stumbling correction reflex for extra knee flexion is activated. Right) A missing contact situation. Knee extension reflex increases the leg length to quickly acquire ground contact. Note that the legs follow a pantograph mechanism, so movements of knee and ankle joints are coupled.

1) Stumbling correction reflex: As the study by Forssberg et al. [21] on cats shows, an extra and fast leg flexion reflex is evoked when a limb hits an obstacle in the swing phase. We formulate this reflex as an impulse feedback to quickly flex the knee (Figure 4-left): $\xi_r \xleftarrow{set} k_r$.

2) Leg extension reflex: The study by Daley et al. [22] shows that if a guinea fowl misses a contact at the beginning of the stance phase, then the leg is extended or at least kept extended until a contact is sensed, and they discuss that such a reaction stabilizes the locomotion. This reflex can be implemented by extra extension of the knee joint when the expected contact is missing (Figure 4-right): $\xi_r \xleftarrow{set} -k_r$.

The above reflexes can be simply implemented by setting a constant instantaneous activation $\pm k_r$. The timing of these reflexes are important, and the state machine in Figure 5 illustrates when each feedback is activated. The discussed reflex impulses should be active for a short time, and we use a simple first order filter to implement a fading memory:

$$\dot{\xi}_r = -\beta_r \xi_r \tag{4}$$

where β_r is set such that the feedback is forgotten (by a ratio of 99.99%) in less than 10% of the stride duration.

C. Posture Control Feedbacks

Posture control mechanisms are needed as soon as the robot locomotes on inclined or irregular surfaces, where body rotations and leg postures should be continuously adjusted. We implement three posture control feedbacks: 1) ξ^{att} for attitude control; 2) ξ^{dir} for direction control; and 3) angle of attack control directly affecting χ_i states.

Attitude and direction control use the same mechanism, but we keep them separate since attitude control is more important compared to direction control and we want to be able to have bigger gains for the attitude control. The posture control feedback signal ξ^p is the sum of the attitude and direction control feedbacks ξ^{att} and ξ^{dir}.

1) Attitude control: In [18] we used Virtual Model Control (VMC) [23] to convert posture control virtual forces to joint torques. VMC uses the Jacobian transpose method [24] to generate torques representing the desired virtual forces. If one wants to generate virtual velocities in the task space (instead of virtual forces), then similarly Jacobian inverse can be used to calculate the joint angular velocities which represent those task space virtual velocities[1]. This later method does not need a torque controlled machine.

[1]This method is also commonly used for iterative / velocity-based inverse kinematics. In the context of quadruped locomotion, see e.g. [25].

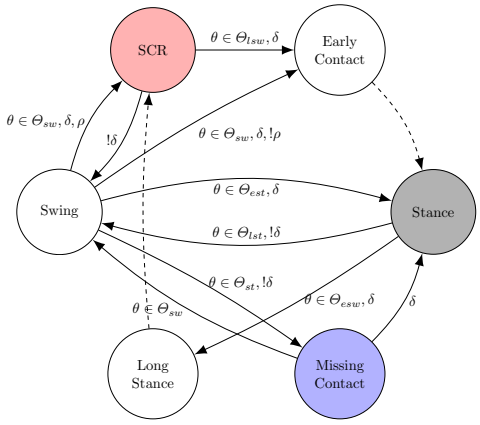


Fig. 5. The state machine used for feedback timing (for one leg). The state machine uses the on/off contact status δ and the leg's sagittal hip phase $\theta = \theta_{PR_l}$ to activate feedbacks. The stumbling correction reflex (SCR) is activated in the *SCR* state (when a swing leg hits an obstacle), and the leg extension reflex is activated in the *Missing Contact* state. The posture control feedbacks are active in the *Stance* phase when the leg is in contact. For this figure, Θ_* is the phase span each state is active in, extracted from a flat terrain run. $*$ can be early swing (esw), swing (sw), late swing (lsw), early stance (est), stance (st), late stance (lst). ρ determines whether a leg is protracting, which is 1 if $\dot{r}_{PR_l} > 0$. The dashed arrows indicate immediate transitions.

To have a general idea, Figure 6 illustrates how task space virtual velocities can be generated to adjust the posture. If Figure 6-left is the present state of the robot, and the (arbitrary) desired body position and orientation are the ones in Figure 6-middle, then virtual velocities in the Figure 6-right (red arrows) can be generated to adjust the posture while keeping the feet at the place they are (without slippage).

Performing attitude control consists of three tasks: estimating ground inclination, adjusting body rotation, and adjusting body position. The sensory information to do all these tasks is the binary (on/off) contact status of each leg, δ, the joint angles read by encoders, and the rotation matrix indicating robot's orientation w.r.t. world coordinates, $\mathbf{R}_{ryp}$, given by an absolute orientation sensor. Rotation matrix $\mathbf{R}_{ryp}$ can be described by roll, yaw and pitch angles (Figure 1) which will be addressed with $\measuredangle_r$, $\measuredangle_y$ and $\measuredangle_p$ respectively. We also use the symbol $\mathcal{R}(.,.,.)$ as the function to reconstruct a rotation matrix from roll-yaw-pitch angles.

We first calculate the yaw-less rotation matrix $\mathbf{R}_{rp} = \mathcal{R}(\measuredangle_r, 0, \measuredangle_p)$, and use it to estimate the ground's pitch (inclination) angle α:

$$\begin{aligned} \mathbf{\Delta p} &= \mathbf{R}_{rp}(\mathbf{p}_{fore} - \mathbf{p}_{hind}) \\ \alpha &= \tan^{-1}(\mathbf{\Delta p}_y / \mathbf{\Delta p}_x) \end{aligned} \tag{5}$$

where $\mathbf{p}_{fore}$ and $\mathbf{p}_{hind}$ are the Cartesian positions of one fore and hind contact legs w.r.t. the frame attached to the robot's trunk. Knowing the ground inclination, we try to keep the body parallel to the ground, and compensate for all the body roll. So the rotation matrix to be adjusted is:

$$\mathbf{R}_{adj} = \mathcal{R}\left(\measuredangle_r, 0, \measuredangle_p - \alpha\right) \tag{6}$$

Additionally, we want the vertical projection of the neck/tail point to be in between the fore/hind feet, to prevent

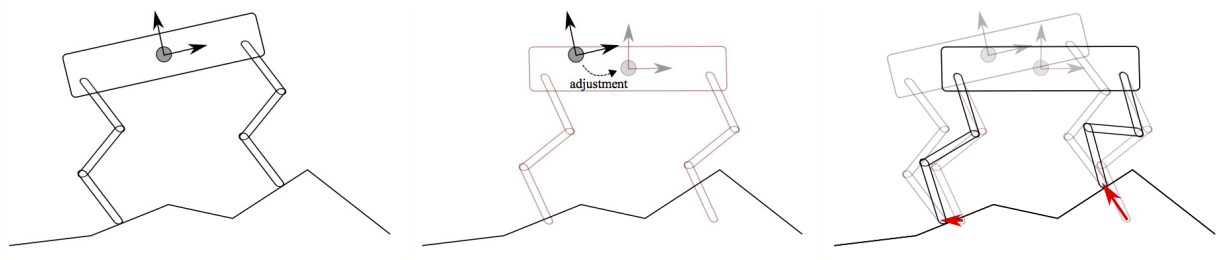


Fig. 6. Using task space virtual velocities to adjust the posture. Left) The initial posture. Middle) An arbitrary desired adjustment of robot's position and orientation. Right) The robot can transit to the desired posture while keeping to feet at the position they are (preventing slippage). This defines the virtual velocities in the task space (red arrows) which bring the robot from the initial posture to the desired one.

a laterally skewed posture. So the position adjustments are:

$$\mathbf{p}_{l,adj} = \frac{1}{2}(\mathbf{p}_l + \mathbf{p}_{contra\{l\}}) \ , \ l = 1..4 \tag{7}$$

where $\mathbf{p}_l$ and $\mathbf{p}_{contra\{l\}}$ respectively are the Cartesian position of the lth foot and its contralateral foot w.r.t. the frame attached to the robot's trunk.

Finally, if $\mathbf{R}_{adj}$ orientation adjustments, and $\mathbf{p}_{l,adj}$ position adjustments should be made, the task space virtual velocities performing this adjustments are:

$$\mathbf{v}_l = (\mathbf{I} - \mathbf{R}_{adj})\mathbf{p}_l + \mathbf{I}(\mathbf{p}_{l,adj} - \mathbf{0}) \ , \ l = 1..4 \tag{8}$$

and the required joint space velocity feedbacks are:

$$\begin{bmatrix} \xi^{att}_{AA_l} \\ \xi^{att}_{PR_l} \\ \xi^{att}_{FE_l} \end{bmatrix} = -k_{att} \ \mathbf{J}_l^{-1}\mathbf{v}_l \ \delta_l \ , \ l = 1..4 \tag{9}$$

where $\mathbf{J}_l$ is the 3×3 Jacobian of the forward kinematics of lth foot Cartesian position w.r.t. the world coordinates, k_{att} is the attitude control gain, δ_l is the on/off contact status of the lth leg, and ξ^{att}_* are attitude control angular velocity feedbacks which are added to the CPG (as a part of ξ^p_*).

2) Direction control: Direction control is done in the same way as attitude control. Assuming that the desired yaw (heading) angle is $\measuredangle^{des}_y$, then the rotation matrix to correct the locomotion direction is:

$$\mathbf{R}_{dir} = \mathcal{R}\left(0, \measuredangle^{des}_y - \measuredangle_y, 0\right) \tag{10}$$

and no position adjustments are needed for turning. Finally Equations 8-9 (replacing $\mathbf{R}_{adj} \leftarrow \mathbf{R}_{dir}$, $\mathbf{p}_{l,adj} \leftarrow \mathbf{0}$, $k_{att} \leftarrow k_{dir}$ and $\xi^{att}_* \leftarrow \xi^{dir}_*$) are used to calculate ξ^{dir}_* terms.

3) Angle of attack: We know from both the Raibert's control [8], and the studies on the Spring Loaded Inverted Pendulum (SLIP) [26], that the angle of attack can be chosen to accelerate or decelerate the body. A more vertical angle of attack will speed up the locomotion, while a more flat angle of attack will causes a break [27]. We use this fact to change the angle of attack while locomoting on slopes, which needs adding (for upwards slope) or removing (for downwards slope) energy to/from the system. Since the PR joint (sagittal hip joint) controls the angle of attack, we linearly couple its oscillation offset to the ground inclination:

$$\chi_{PR_i} = k_\chi \alpha \tag{11}$$

where k_χ is the angle of attack control gain, which should be around 1 to have a rather vertical leg posture w.r.t. world coordinates.

III. EXPERIMENTS

A. The Simulation Platform

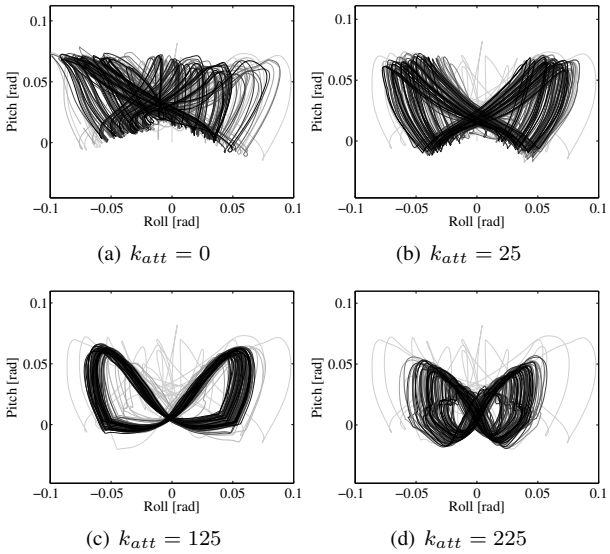
We experiment with a simulated robot which is a detailed model of the Oncilla robot [28], see Figure 1. The forward dynamics physics simulation is done using the Webots commercial software (with customized physics plugins to be as close as possible to the robot), and interfaced using the AMARSi Software Architecture [29]. Both physics simulation and control loop are working at 500Hz (2ms timestep).

The (simulated) robot is a lightweight quadruped with passively compliant legs. The robot weighs 3.9Kg, the standing hip height is about 180mm, the distance between the shoulder/hip axes is 215mm ipsilaterally , and 128mm contralaterally. Each leg follows a three segmented pantograph mechanism, keeping the first and third segments parallel. All actuation is done proximally, so the legs are low-inertia. AA and PR joints are controlled on their motor axes, and the FE joint is controlled using a cable-clutch mechanism, actuated near the shoulder/hip point. Because of the parallel mechanism, the range of motion does not allow for singular configurations (e.g. a fully stretched leg).

All the results which are reported in the following are for the simulated robot locomoting with a trot gait with a forward speed of about 0.4m/s, more than 1.7BL/s (BL: Body Length, ipsilateral shoulder to hip distance). At the time of writing, the hardware robot experiments are initiated, and the hardware robot locomotes with the CPG module on flat terrain with a forward speed similar to the simulated one. An absolute rotation sensor (MicroStrain 3DM-GX3-35) is being mounted on the robot, and rough terrain locomotion control will be validated on the robot in near future. A video of the hardware robot running with an open-loop CPG on flat terrain is included in the `accompanied_video`.

B. Parameter tuning

As we discussed during the introduction, we aim for a control architecture where the modules can be tuned on top of each other. Here we will show how this goal is obtained. All the gains are initialized with zero values, and then they are set sequentially.



(a) $k_{att} = 0$ (b) $k_{att} = 25$

(c) $k_{att} = 125$ (d) $k_{att} = 225$

Fig. 7. Roll and pitch variations (RPV) for different attitude control gains. A good value of for k_{att} stabilizes the RPV and make it more periodic, however excessive increase of this gain can have a counter effect.

The first module to tune is the CPG module. Many locomotion controllers, including CPG models, inverse dynamics controller, etc need the joint angle references to be provided. These joint angle profiles can be hand-tuned or set using optimization techniques. We hand-tune the PR and FE joint angle profiles (starting from sine profiles, and modifying for about 20 to 30 trials), which in turn defines the f functions defining the limit cycle shape of the oscillators ($f = 0$ for AA joints). f functions are modeled using piecewise cubic Hermite polynomials [30] with four knot points in one period. Stride duration is set to 0.4s, which gives the locomotion frequency of 2.5Hz. CPG's convergence rate γ is set such that typical perturbations are damped in less than 10% of the stride duration. This defines $\gamma = 50$. For all the oscillators $c_{ij} = 5$, and the phase differences ϕ_{ij} define a trot gait. The phase difference between PR and FE joints is set to be $\pi/3$.

After tuning the CPG module, we tune the attitude control gain k_{att}. As we have shown in our previous study [18], attitude control regularizes the role-pitch variations (RPV) and stabilizes them. So we set the k_{att} such that it stabilizes the RPV (Figure 7). Normally a step-by-step increase of k_{att} makes RPV more periodic, but excessive increase of it can disturb the locomotion. We have illustrated the RPV for four values in Figure 7. As it is shown, there a good value around $k_{att} = 125$.

We then set the angle of attack offset k_χ such that the robot could go down a 20% slope. We start with a default value of 1, and then slightly increase it to obtain the desired performance. This gives a value of $k_\chi = 1.25$.

After that we set the reflex gain k_r such that the robot could overcome an obstacle, and a step-down with height equal to 20% of the leg-length. This leads to a reflex gain of $k_r = 50$. We finally set the direction control gain k_{dir} such that the robot could turn with a minimum turning rate of 45 deg/s, which gives $k_{dir} = 25$.

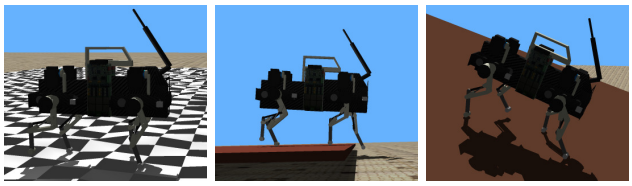


Fig. 8. Unperceived rough terrain scenarios. Left) Randomized uneven terrain. Middle) Step. Right) Downwards slope.

C. Rough Terrain Locomotion

Three different scenarios were used to evaluate the proposed control strategy (Figure 8):

- Randomized uneven terrain, 12% of the leg-length height variations (max local slope $= \pm 20\%$);
- 20% downward slopes;
- Step down, 20% of the leg-length height.

Each of the above scenarios is repeated 25 times from different initial conditions (robot is placed in different initial positions w.r.t. the rough terrain). Each experiment is ran for 20 seconds from which the first $5 - 8$ seconds is used for initialization (unperturbed). The same gains as described in section III-B are used for all the scenarios, and we do not change or re-tune the gains for different scenarios. The controller does not have any kind of prior information about the environment and the perturbation scenario.

The overall results of the rough terrain locomotion scenarios are shown in Figure 9, and provided in the `accompanied_video`. A CPG-only control was partially successful on the **randomized uneven terrain**. As from our previous study on a stiff quadruped [18], we were expecting the open-loop control to perform badly, however, a 56% success rate was obtained. This partial success is due to the compliance, which prevents minor stumblings by passive deflection of the legs, and moderately self-stabilizes the roll and pitch oscillations. This is similar to what is reported in [17]. Nevertheless, the posture control mechanisms are needed for a better performance. As Figure 9 shows, a 96% success rate is obtained by applying the closed-loop control.

The CPG-only control was mostly unsuccessful in the **step** scenario and only 20% of the trials were successfully passed. In contrast, the CPG control with reflex and posture control feedbacks successfully passed the trials. The leg extension reflex is very important for this scenario, as it compensates for the missing contact at the step down. The posture control mechanism comes into play after the step where the body oscillations, induced by the perturbation caused by the step, should be stabilized.

None of the **slope** experiments were successfully passed using a CPG-only control. Again, both reflex and posture control mechanisms are crucial for success in this scenario as they prevent stumbling, compensate for missing contacts, and keep the body roll and pitch oscillations contained. We additionally tested our control method against 36.5% (20 degrees) downward slopes, which are quite difficult as unperceived rough terrain. We realized that a fine tuning of the reflex gains is needed for this case ($k_r = 120$ for the extension reflex and $k_r = 50$ for the stumbling correction

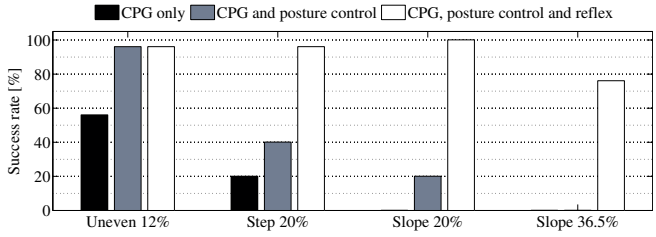


Fig. 9. Performance of the *CPG only*, *CPG + posture control* and *CPG + reflex + posture control* controllers on unperceived rough terrain. The *CPG only* control is partially successful on the uneven terrain because of the compliance, and the compliance fulfills the role of a weak reflex mechanism. A much better performance can be obtained by adding the posture control module. Only the complete control (*CPG + posture control + reflex*) is successful in all of the scenarios. We additionally test with an extra scenario, downward 36.5% slopes, and the robot was successful in 19 out of 25 trials. In all the scenarios, a consistent increase of the performance is observed by adding the posture control and reflex modules.

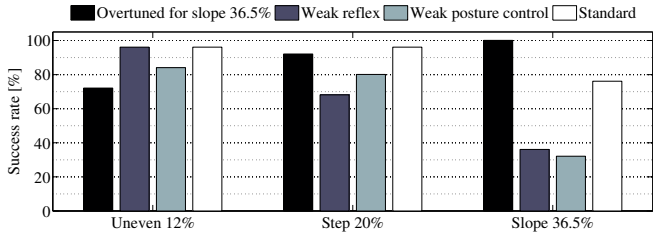


Fig. 10. Variations of the control parameters. Black) The control can be overtuned to perform well on 36.5% slopes, but overtuning will affect the performance in the other scenarios. Purple) A weaker reflex ($k_r = 25$) leads to lower performance in step and slope environments. Azure) Weaker posture control gain ($k_{att} = 60$) affects the whole performance.

reflex), and then the robot can pass this scenario successfully (Figure 10, black columns). This means that, having the prior knowledge about the environment, the reflex gain can be coupled to the slope inclination. This can be a direction for further research.

To show the importance of the reflex and posture control modules, we ran the control with different reflex and posture control gains. Figure 10 shows the performance of the control with reduced posture control and reflex gains. A lower reflex gain ($k_r = 25$) lowers the performance in case of the steps and the steeper slopes, and a reduced posture control gain ($k_{att} = 60$) affects the overall performance.

D. Control Signals

Figure 11 illustrates the evolution of the control signals over time for locomotion on the randomized uneven terrain[2]. The illustrations are for three stride cycles of a hind knee (FE) joint. Posture control feedbacks continuously adjust the joint angle reference, while reflexes are short term and for fast corrections. The CPG state r_i converges back to the coded limit cycle f_i in each swing phase (white background), and the effect of the feedbacks are damped since there is no ground contact, hence the control system resynchronizes.

[2]For Figures 11-13, left and right y-axes correspond to the solid and dashed lines respectively. For example, the top subplot in Figure 11 contains two trajectories, closed-loop reference r_i with black solid lines, and open-loop reference f_i with red dashed lines. The y-axis quantities are different for each subplot, and correspond to the ones in Equation 2.

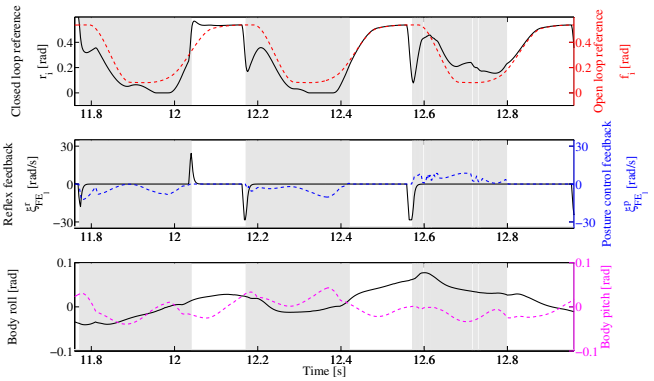


Fig. 11. Control signals for an example run on uneven terrain. The signals are for the control of a hind knee (FE) joint. Posture control feedback continuously adjusts the control reference. Stumbling correction reflex is activated just after $t = 12$s and the leg extension reflex is activated two times before $t = 12.2$s and $t = 12.6$s. Please note that positive values for the FE joint relate to flexions (shortening of the leg length).

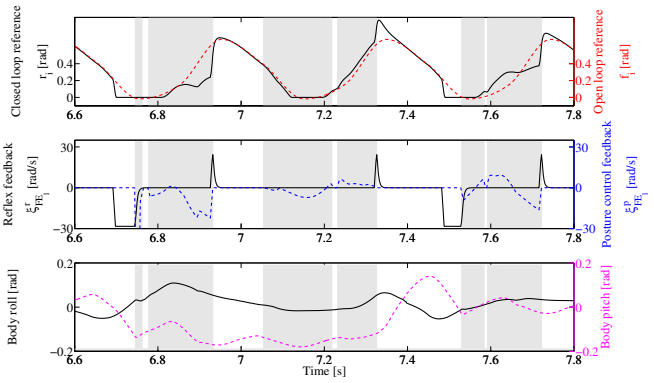


Fig. 12. Control signals for the moment that the robot goes over a downwards step (fore knee). The step occurs around $t = 6.7$s, and causes a missing contact state, and a leg extension reflex is activated until a contact is sensed. At that moment, since the robot is pitched, the posture controller is strongly activated to correct the body posture. Since the robot is pitched the fore leg drags on the ground in the beginning of the two next swing phases, and stumbling correction reflexes are activated. The hind leg comes down the step around $t = 7.4$ (see the correction in the pitch angle), and causes a small impact which slightly lifts the front of the robot, and another leg extension reflex is activated in the fore knee to acquire ground contact. The reflex and posture control feedbacks are damped in the beginning of each swing phase (white background), and the system resynchronizes.

Figure 12 illustrates example control signals at the moment of a step down, for a fore knee (FE) joint. Again, the posture control feedbacks are continuously adjusting the joint angles reference, while the reflexes are quick and short term. Please refer to the caption of Figure 12 for details.

Figure 13 corresponds to locomotion on a downwards 36.5% slope. The signals are for the sagittal hip (PR) joint of a fore leg (since there are no reflexes implemented for the PR joint, ξ^r is given for the FE joint of the same leg). As the figure shows, the body rotations are stabilized, and the activation of the feedbacks are repetitive over the cycles. The effect of the angle of attack feedback χ_{PR_l} is also visible in the offset added to the r_i reference.

E. Extension: Vision Feedback

Note: This extension is contrary to the main topic of this paper (environment being unperceived), and is only given

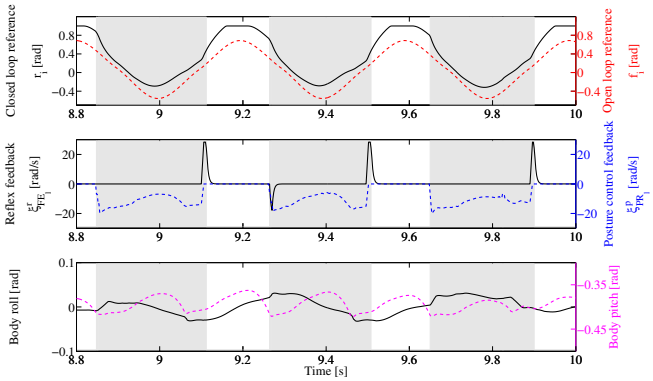


Fig. 13. The control signals for a saggital fore hip (PR) in a 36.5% slope scenario. the angle of attack control adds an extra offset χ_{PR_l} to the r_i joint angle reference (there is a soft joint limit at $r_i = 1$). The activation of reflex and posture control feedbacks is quite repetitive, which means that the robot is in a new limit cycle behavior adjusted for the slope. The same gains as in section III-B are used.

as a proof-of-concept. This extension is not used for the previously reported results. □

In the slope scenario, the locomotion direction is always along the slope. We tested our control when going on the slopes with a heading (yaw) angle different from zero. The control can tolerate heading angles up to ± 15 degrees, and could successfully travel on the slopes, but is unsuccessful for heading angles bigger than that.

For this reason, we extended our control with a simple vision feedback to detect the heading angle towards the slope before going on it. This extension uses a camera mounted above the shoulders of the robot. As Figure 14 illustrates, the line of horizon can be detected utilizing canny edge detection and probabilistic Hough transform [31]. The orientation of this line is correlated with the robot's heading w.r.t the slope direction. If the calculated horizon orientation is h, then the desired locomotion direction (Eq. 10) can be corrected:

$$\measuredangle_y^{des} = k_{vis} h \tag{12}$$

where k_{vis} (set to 10) is the correlation gain, determining how fast the direction should be corrected. Applying this vision feedback, the robot corrects the direction before going on a slope. The extension here is a simple implementation, and is presented as a proof-of-concept how exteroception can be added to the control. More complex setups like stereo vision should be used for a real environment.

IV. DISCUSSION

We presented a modular control approach to locomotion based on modules that are meaningful, and that can be hierarchically put on top of each other. The control approach is fit for unperceived rough terrain locomotion with cheap and lightweight quadruped platforms. Sensing of the joint angles (encoders), body rotations (absolute rotations sensor), and on/off contact data (bumpers) are the sufficient ingredients of the proposed control method.

A Central Pattern Generator (CPG) implemented as coupled nonlinear oscillators is used as the core, which can encode the desired arbitrary limit cycle shape ensuring its

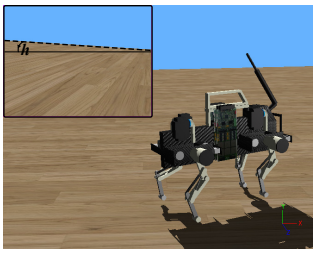


Fig. 14. Detecting the horizon angle h for heading (yaw) correction before a downwards slope. If the robot is not straight towards the slope then, in the filed of view, the distance to the slope is different for the left and right pixels of the camera image. This causes a rotated line of the horizon, which can be detected using the Hough transform. The horizon angle h can then be correlated to the locomotion direction.

global asymptotic stability. This means that adding (finite-time) feedbacks to the CPG will not cause any instability and the system will go back to the desired limit cycle as soon the feedbacks are not active (in the swing phase).

Reflex modules are added to the CPG for fast correction including the stumbling correction reflex and leg extension reflex. Additionally, model based posture control mechanisms are added to adjust body position and orientation continuously over time. The obtained control architecture allows for moderately fast (more than 1.7BL/s forward speed) dynamic locomotion over unperceived rough terrain of medium difficulty.

Compared to the CPG approaches in [10], [12], the proposed control architecture is simple and hierarchical as the modules are not strongly interconnected and can be tuned on top of each other. Also we experiment with more difficult rough terrain scenarios and more systematically, however only in simulation so far. Hardware robot experiments are now underway to validate our results. Nevertheless, the purpose of this paper is only to introduce a modular control methodology apt for unperceived rough terrain locomotion.

The proposed control approach does not depend on inverse dynamic control (like in [14], [15]) that makes the low-level control gains smaller (less stiff control), but is instead lighter and simpler. Our methodology does not depend on careful sensing of the ground reaction forces, a need for torque sensors, or a requirement to know about the mass properties of the robot like the inertia tensors.

Future extensions of the introduced control strategy include: 1) exploiting body acceleration information for lateral foot placement; 2) implementing a phase resetting mechanism (the state machine in Figure 5 already has the activation state for this, but the feedback is not yet implemented); and 3) Exploiting compliance for energy efficiency. We have observed that in a certain range of the attitude control gains, the leg springs can go into a resonance-like behavior. We will explore this effect further in the future.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community's 7th Framework Programme FP7/2007-2013-Challenge 2-Cognitive Systems, Interaction, Robotics-grant agreement No. 248311 (AMARSi). We thank Albert Mukovskiy for helping us with the stability analyses.

APPENDIX I: STABILITY

The coupled morphed oscillators in Equations 1-3 form a hierarchical system. The phase dynamics drives the radial

dynamics, but is not affected by it. This means that the stability of the phase dynamics can be analyzed separately. The coupling dynamics have the potential function:

$$U(\boldsymbol{\theta}) = -\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}\cos(\theta_j - \theta_i - \phi_{ij}) \qquad (13)$$

which has the minima at $\forall i,j : \theta_i = \theta_j - \phi_{ij} + 2k\pi, \forall k \in \mathbb{Z}$. Since $\frac{dU}{dt} = -\sum_{j=1}^{N}(\frac{\partial U}{\partial \theta_j})^2$, then $U(\boldsymbol{\theta})$ plays a role of Lyapunov's function (with $c_{ij} > 0$), proving the asymptotic stability of the coupling. Now if the phase differences are consistent, the system will not be perturbed and remains synchronized. When the oscillators are coupled and not perturbed, they converge to the desired phase differences in the long term. Consequently, the phase dynamics become: $\dot{\theta}_i = \omega + \epsilon, \epsilon \to 0$, and the dimensions become decoupled. So the radial stability of the whole system can be proved by addressing the stability of each dimension. To analyze the asymptotic stability of the radial dynamics of one dimension, lower and upper phase-dependent bounds are defined as:

$$B_{L|U}(\theta) = \mu f(\theta) + \chi + \kappa;\ \kappa \lessgtr 0 \qquad (14)$$

These bounds define closed regions in an orientable 2-manifold of $\theta_i \times r_i : [0, 2\pi) \times \mathbb{R}$ which the dynamics flows (strictly) enter them and never leave them, for all $\kappa \in \mathbb{R} - 0$ ($\kappa = 0$ is the limit cycle itself). Utilizing the Poincaré-Bendixson theorem [32], this proves the asymptotic stability of the radial limit cycle. Detailed stability analyses is out of the scope of this paper, and is currently in-press [33].

REFERENCES

[1] J. Buchli, M. Kalakrishnan, M. Mistry, P. Pastor, and S. Schaal, "Compliant quadruped locomotion over rough terrain," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, oct. 2009, pp. 814 –820.

[2] M. Mistry, J. Buchli, and S. Schaal, "Inverse dynamics control of floating base systems using orthogonal decomposition," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, may 2010, pp. 3406 –3412.

[3] S. Rutishauser, A. Spröwitz, L. Righetti, and A. J. Ijspeert, "Passive compliant quadruped robot using central pattern generators for locomotion control," in *2008 IEEE International Conference on Biomedical Robotics and Biomechatronics*, 2008.

[4] J. Estremera and K. J. Waldron, "Thrust control, stabilization and energetics of a quadruped running robot," *The International Journal of Robotics Research*, vol. 27, no. 10, pp. 1135–1151, 2008.

[5] M. P. Murphy, A. Saunders, C. Moreira, A. A. Rizzi, and M. Raibert, "The littledog robot," *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 145–149, 2011.

[6] J. Pippine, D. Hackett, and A. Watson, "An overview of the defense advanced research projects agencys learning locomotion program," *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 141–144, 2011.

[7] M. H. Raibert, "Trotting, pacing and bounding by a quadruped robot," *Journal of Biomechanics*, vol. 23, no. Supplement 1, pp. 79–81, 1990.

[8] M. Raibert and J. Hodgins, "Animation of dynamic legged locomotion," in *ACM SIGGRAPH Computer Graphics*, vol. 25, no. 4. ACM, 1991, pp. 349–358.

[9] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, "BigDog, the Rough–Terrain quadruped robot," in *Proceedings of the 17th IFAC World Congress*, COEX, South Korea, 2008, pp. 10 823—10 825.

[10] Y. Fukuoka, H. Kimura, and A. H. Cohen, "Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts," *The International Journal of Robotics Research*, vol. 22, no. 3-4, pp. 187 –202, 2003.

[11] H. Kimura, Y. Fukuoka, and A. Cohen, "Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts," *The International Journal of Robotics Research*, vol. 26, no. 5, pp. 475–490, 2007.

[12] C. Maufroy, H. Kimura, and K. Takase, "Integration of posture and rhythmic motion controls in quadrupedal dynamic walking using phase modulations based on leg loading/unloading," *Autonomous Robots*, vol. 28, no. 3, pp. 331–353, 2010.

[13] A. Ijspeert, "Central pattern generators for locomotion control in animals and robots: A review," *Neural Networks*, vol. 21, no. 4, pp. 642 – 653, 2008.

[14] T. Boaventura, C. Semini, J. Buchli, M. Frigerio, M. Focchi, and D. G. Caldwell, "Dynamic torque control of a hydraulic quadruped robot," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1889–1894.

[15] M. Hutter, M. Hoepflinger, C. Gehring, M. Bloesch, C. D. Remy, and R. Siegwart, "Hybrid operational space control for compliant legged systems," in *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2012.

[16] M. Vukobratović and B. Borovac, "Zero-moment pointthirty five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 01, pp. 157–173, 2004.

[17] A. Sproewitz, A. Tuleu, M. Vespignani, M. Ajallooeian, E. Badri, and A. J. Ijspeert, "Towards dynamic trot gait locomotion design, control, and experiments with a compliant quadruped robot," *International Journal of Robotics Research*, To appear in.

[18] M. Ajallooeian, S. Pouya, A. Sproewitz, and A. Ijspeert, "Central pattern generators augmented with virtual model control for quadruped rough terrain locomotion," in *IEEE International Conference on Robotics and Automation (ICRA 2013)*, 2013.

[19] K. Tsujita, H. Toui, and K. Tsuchiya, "Dynamic turning control of a quadruped robot using nonlinear oscillators," in *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 1, sept.-2 oct. 2004, pp. 969 – 974 vol.1.

[20] L. Righetti and A. J. Ijspeert, "Pattern generators with sensory feedback for the control of quadruped locomotion," in *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA 2008)*, 2008, pp. 819–824.

[21] H. Forssberg, "Stumbling corrective reaction: a phase-dependent compensatory reaction during locomotion," *Journal of Neurophysiology*, vol. 42, no. 4, pp. 936–953, 1979.

[22] M. A. Daley and A. A. Biewener, "Running over rough terrain reveals limb control for intrinsic stability," *Proceedings of the National Academy of Sciences*, vol. 103, no. 42, pp. 15 681–15 686, Oct. 2006.

[23] J. Pratt, C.-M. Chew, A. Torres, P. Dilworth, and G. Pratt, "Virtual model control: An intuitive approach for bipedal locomotion," *The International Journal of Robotics Research*, vol. 20, no. 2, pp. 129–143, 2001.

[24] R. P. Paul, *Robot Manipulators: Mathematics, Programming, and Control*, 1st ed. Cambridge, MA, USA: MIT Press, 1982.

[25] M. Mistry, J. Nakanishi, and S. Schaal, "Task space control with prioritization for balance and locomotion," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE, 2007, pp. 331–338.

[26] P. Holmes, R. J. Full, D. Koditschek, and J. Guckenheimer, "The dynamics of legged locomotion: Models, analyses, and challenges," *Siam Review*, vol. 48, no. 2, pp. 207–304, 2006.

[27] J. Hodgins and M. Raibert, "Adjusting step length for rough terrain locomotion," *Robotics and Automation, IEEE Transactions on*, vol. 7, no. 3, pp. 289–298, 1991.

[28] A. Sproewitz, L. Kuechler, A. Tuleu, M. Ajallooeian, M. DHaene, R. Moeckel, and A. Ijspeert, "Oncilla robot, a light-weight bio-inspired quadruped robot for fast locomotion in rough terrain," in *Symposium on Adaptive Motion of Animals and Machines*, 2011, pp. 63–64.

[29] A. Soltoggio and J. J. Steil, "How Rich Motor Skills Empower Robots at Last: Insights and Progress of the AMARSi Project," *Künstliche Intelligenz*, 2012.

[30] F. N. Fritsch and R. E. Carlson, "Monotone piecewise cubic interpolation," *SIAM Journal on Numerical Analysis*, vol. 17, no. 2, pp. pp. 238–246, 1980.

[31] N. Kiryati, Y. Eldar, and A. M. Bruckstein, "A probabilistic hough transform," *Pattern recognition*, vol. 24, no. 4, pp. 303–316, 1991.

[32] J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. New York, USA: Springer-Verlag, 1983.

[33] M. Ajallooeian, J. van den Kieboom, A. Mukovskiy, M. Giese, and A. Ijspeert, "A general family of morphed nonlinear phase oscillators with arbitrary limit cycle shape," *Physica D: Nonlinear Phenomena*, 2013.