

# An Active Compliance Controller for Quadruped Trotting\*

Konstantinos Machairas, *Student Member, IEEE* and Evangelos Papadopoulos, *Senior Member, IEEE*

**Abstract**—In this paper, we present the idea that very simple control schemes can lead to complex running gaits. To show this, a novel control framework is proposed for dynamic quadruped trotting, consisting of a toe trajectory planning part and an active compliance part driving each leg. While keeping the controller's structure simple, the notion of the *system virtual stiffness* is introduced as a single control parameter tuned to drive a quadruped through a complete dynamic trotting scenario, with acceleration from stance, constant speed locomotion, and deceleration. The method can serve as a tool for generating robust running locomotion at a wide range of speeds, or as a basis for other high level controllers. The presented ideas are evaluated through simulation experiments with a 2d dynamic model of four three-segment legs.

## I. INTRODUCTION

Recently, the robotics community in both academia and industry presented impressive results in theory and experiments concerning legged locomotion. Quadruped robots with different actuation systems, such as the MIT Cheetah 2, [1], the HyQ, [2], and the Boston Dynamics Spot, [3], stepped out of the labs and performed complex gaits, high jumps, difficult turns and other challenging tasks. However, despite these major steps forward, there is still a long way until legged robots can successfully perform tasks in real life and replace wheeled, flying or other robots in tasks such as planet exploration, where their superiority is evident in theory. To this end, important issues must be tackled in design, control, and technology, which will ensure stability, robustness and efficiency for legged machines.

Throughout the years, several control methodologies have been proposed and tested in simulated models and in real robots. The first controllers addressed the problem of dynamic legged locomotion by suitably positioning the legs in the aerial phase, and by controlling the hip and knee torques in the stance phase, while passive compliance elements were employed to store energy and to smoothen the interaction with the ground, [4]. Next, controllers that included direct force control schemes together with foot-end trajectory planning algorithms were proposed for robots equipped with force sensors, [5]. Another important contribution to legged systems theory came from the Central Pattern Generators (CPGs) approach, with a series of works presenting neural

circuits able to produce easily rhythmic patterns generating trajectories, [6]. However, tuning these control parameters is not straightforward; so far they have been mostly applied in statically stable gaits, [7]. Furthermore, in an effort to achieve energy efficient locomotion, the stability of the passive dynamics of running robots was examined, [8].

Most of the aforementioned approaches, used passive compliance elements, e.g. spring-damper systems, in order to perform dynamic gaits, in the absence of which, they only managed to perform statically stable gaits. It is thus evident that compliant interaction with the ground is a key parameter in dynamic locomotion, and a control scheme is characterized by how it tackles this issue. However, adding only fixed compliance into the robot's structure is not the best solution, making it hard for it to cope with an adequate range of tasks.

The robotics community addressed this problem using two approaches. The first, and more intuitive one, tried to include adjustable stiffness mechanisms into the robot's structure, [9]. However, despite their advantages, such as the ability to store energy only by using common elements like springs and motors, they resulted in heavy and sophisticated mechanisms that still had difficulty in presenting a reasonable stiffness range. On the other hand, a second and promising approach recently came to surface, with interesting works on active compliance control, presenting the first robots capable of running without passive compliance elements, [2], [10]. Experiments with the MIT Cheetah 2 proved that passive compliance is not necessary for dynamically stable locomotion, while experiments with IIT's HyQ showed that virtual electrical spring-damper systems can replace the mechanical ones, if some conditions are met. Both robots used hierarchical controllers that performed position control in each leg's swing phase and impedance control in its stance phase, using preprogrammed foot-end trajectories for each phase. These works have clearly stated that active compliance will play a major role in legged robots of the future. However, many questions remain unanswered, and thus, much research has to be done on technical and theoretical issues regarding these types of controllers.

In this paper, a control scheme for dynamic quadruped trotting is proposed, characterized by great simplicity compared to most controllers proposed so far. We claim that there is no need to detect stance to aerial phase transition for each leg, or to switch to a special controller for each phase, and on this basis, we devise a unified controller consisting of a toe trajectory planning part and an active compliance part driving each leg. A relationship is found between the robot's forward velocity and a control parameter that we call

\* This research has been financed by the "IKY Fellowships of Excellence for Postgraduate Studies in Greece – Siemens Programme" in the framework of the Hellenic Republic – Siemens Settlement Agreement.

K. Machairas (e-mail: kmach@central.ntua.gr), and E. Papadopoulos (egpapado@central.ntua.gr) are with the Department of Mechanical Engineering, National Technical University of Athens, 15780 Athens, Greece.

system virtual stiffness, corresponding to the virtual stiffness of all actuated joints. In this way, a simple method is provided to generate stable and accelerating trotting at a wide range of forward velocities by properly adjusting this single parameter during the gait. Finally, the proposed framework is evaluated through simulation experiments with a 2d quadruped model performing running trotting at various speeds.

## II. ROBOT MODEL AND SYSTEM DYNAMICS

### A. Robot Model

We introduce a planar model of a quadruped robot with structure inspired from biology, and mostly by the Cheetah, the fastest and one of the most agile animals on earth. As shown in Fig. 1, the model consists of a body of mass  $m_b$ , and moment of inertia (MoI)  $I_b$  about its center of mass (CoM), and four three-segment legs, with each segment being of mass  $m_{leg,i}$ , MoI  $I_{leg,i}$  about its CoM, and length  $l_{leg,i}$  where  $leg \in \{FR, HR, FL, HL\}$  and  $i \in \{1, 2, 3\}$ , with  $F$  standing for fore,  $H$  for hind,  $R$  for right, and  $L$  for left. The segments of a leg are numbered starting from the ones hinged to the body using the subscript  $i$ . Each segment's CoM is located in the middle of its length. The hip to hip distance is denoted by  $2d$ . The hip and knee joints are driven by actuators modeled as ideal torque sources  $\tau_{leg,j}$ , with the subscript  $leg$  same as defined above and  $j \in \{1, 2\}$ . Finally, a torsional spring of stiffness  $k_a$  is located at each ankle joint. The damping coefficients related with each joint are denoted by  $b_{leg,i}$ , with subscripts same as defined above.

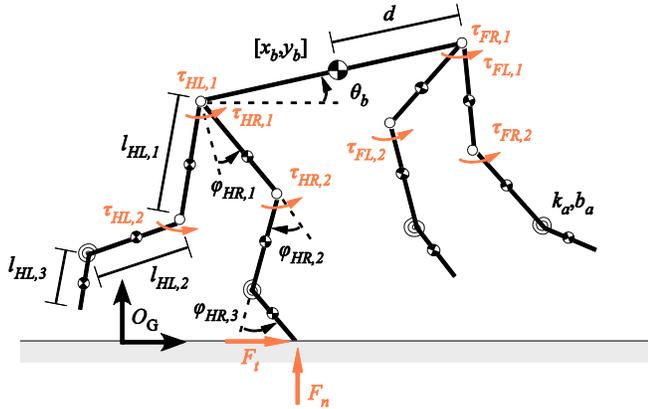


Figure 1. The dynamic model of a biomimetic quadruped robot.

### B. Ground Forces Modeling

We assume a point contact each time a toe impacts the ground. To avoid the unrealistic assumptions of the discrete contact models usually used in legged systems analysis, we use a continuous compliant Hunt-Crossley model, as described in [11]. The normal contact force is thus given by:

$$F_n = k_g \delta_y^n + b_g \delta_y^n \dot{\delta}_y \quad (1)$$

where  $\delta_y$  and  $\dot{\delta}_y$  are the local indentation and its rate respectively,  $n=1.5$  in the case of Hertzian non-adhesive contact,  $k_g$  is the stiffness coefficient depending on the ma-

terials coming in contact, and  $b_g$  is the damping coefficient calculated as a function of stiffness, [12], given by:

$$b_g = 1.5c_a k_g \quad (2)$$

Here,  $c_a$  is considered equal to 0.2 without affecting the generality of the conclusions.

Friction is also added to the ground model in order for the results to be more realistic. A common exponential model that was first introduced in [13] is used to describe the friction-velocity curve when slippage occurs, and a proportional to velocity model to describe the static friction. Finally, a velocity threshold  $u_e$  is used as a criterion for the transition between stiction and slippage. An advantage of this approach is the separation of the rigid body dynamics from the contact dynamics, since the ground forces can be computed only from position and velocity variables. In this way, the frictional forces can be computed simply as:

$$F_t = \begin{cases} -\text{sgn}(\dot{x}_{ft}) \cdot \mu_s \cdot F_n \cdot |\dot{x}_{ft} / u_e|, & |\dot{x}_{ft}| \leq u_e \\ -\text{sgn}(\dot{x}_{ft}) \cdot F_n \cdot (\mu_c + (\mu_s - \mu_c) \cdot e^{-|\dot{x}_{ft}/u_e|}), & |\dot{x}_{ft}| > u_e \end{cases} \quad (3)$$

where  $\mu_c$  is the kinetic friction coefficient,  $\mu_s$  is the static friction coefficient,  $\dot{x}_{ft}$  is the tangential velocity of the foot,  $u_s$  is the Stribeck velocity, and  $F_n$  the normal force as defined above. With this model,  $F_t$  is constrained to lie in the friction cone, and if there is a force in this range that can prevent slippage, then this is the value of  $F_t$ ; otherwise,  $F_t$  lies on the nearest range boundary, and slippage occurs.

### C. Equations of Motion

We parameterize the space of generalized coordinates by the absolute pitch angle of the body  $\theta_b \in \mathbb{S}^1$ , the position vector of the body CoM  $\mathbf{p}_b \in \mathbb{R}^2$  w.r.t. the inertial reference frame  $O_G$ , and the relative angles of the legs' links  $\varphi_{leg,i}$ , as shown in Fig. 1. Using the Lagrangian formalism, the equations of motion (EoM) are written as:

$$\mathbf{M}(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B} \cdot \boldsymbol{\tau} + \mathbf{J}^T \cdot \mathbf{F} \quad (4)$$

where  $\mathbf{q}$  is the vector containing the generalized coordinates,  $\mathbf{M}$  the mass matrix,  $\mathbf{C}$  the matrix containing gravity, centrifugal and Coriolis terms,  $\mathbf{B}$  the input matrix, and  $\mathbf{J}$  the Jacobian that maps the external forces from the ground  $\mathbf{F}$  into the generalized coordinate space  $\mathbf{q}$ . We use these EoM for analysis and in our dynamics simulator.

## III. CONTROLLER DESIGN

In this section, we propose an active compliance control framework for dynamic trotting. Trotting is a two-beat gait, in which diagonal legs move together and hit the ground at the same time. It is widely used in animals and is one of the first gaits a quadruped robot should be able to perform successfully. A significant metric here, the duty factor  $D \in (0, 1]$  of a leg, is defined as the ratio between the stance duration and the stride period. If all the legs have a duty factor of less than 0.5, then there is certainly an aerial phase during the stride, during which no leg touches the ground; this is called a *flying* or a *running trot*.

The control framework presented herein consists of a higher and a lower level controller. The high-level part is responsible for tuning the parameters of the low-level controller, which is responsible for driving the legs along elliptical trajectories, using an active compliance control scheme. The major differences between this work and related approaches, [10], [14], lie primarily on the simplicity of the reference trajectories and the active compliance scheme, and secondarily on the use of the virtual stiffness of the actuated joints – or *system virtual stiffness* – as a single control parameter tuned to drive a quadruped through stable trotting gaits, spanning a wide range of forward velocities.

#### A. Low-level Controller

The low-level controller consists of three parts: (a) a trajectory planning part employing elliptical primitives in each leg's workspace, (b) an inverse kinematics part to calculate the desired joint angles, and (c) a joint-level active compliance part driving each leg, while at the same time handling the interaction with the ground.

##### Trajectory planning

The idea here is that each toe should move along a nearly elliptical trajectory in order for the robot to move forward. To this end, this first part of the controller generates a sequence of points along an elliptical primitive in the workspace of each leg, which will be used as a reference input to the leg motion controller. This elliptical path is given w.r.t. a frame fixed to each leg's hip and parallel to the inertial reference frame  $O_G$ , as in [14]. The primitive is next used as a virtual trajectory in an active compliance control scheme that drives each leg, while at the same time interacting compliantly with the ground. We borrow the term virtual from impedance control, since this path is not to be followed in a strict manner, but it is only the reference input to the motion controller that will be presented next.

Although similar trajectory planning methods have been presented already, employing Bezier curves, [10], and semi-elliptical trajectories [14], we claim that the simpler primitives presented herein are sufficient to achieve dynamic locomotion of similar complexity. In this section, we will avoid using subscripts for each leg's variables, aiming at a better presentation of the equations. The parametric equations of an elliptical trajectory w.r.t. the hip-fixed frame  $O_H$ , with the ellipse center position given by the vector  $\mathbf{p}_c = [x_c, y_c]^T$ , and the ellipse semi-axes denoted by the parameters  $a$  and  $b$ , as shown in Fig. 2, are:

$$\begin{aligned} x_{ir} &= x_c + a \cdot \cos(\omega_r t + \varphi) \\ y_{ir} &= y_c + b \cdot \sin(\omega_r t + \varphi) \end{aligned} \quad (5)$$

where  $\mathbf{p}_{ir} = [x_{ir}, y_{ir}]^T$  is the vector containing the coordinates of the elliptical path w.r.t. the frame  $O_H$ ,  $\omega_r$  is the angular velocity of the motion along the elliptical path, and  $\varphi$  is a phase variable used for the synchronization of the legs. For a trotting gait, we set  $\varphi = 0$  for the FR and the HL leg, and  $\varphi = \pi$  for the FL and the HR leg. See Fig. 2 for a better understanding of the notation.

#### Inverse Kinematics

Since the control action is finally implemented in the joint space, an inverse kinematics algorithm is used to reconstruct the time sequence of the joint angles corresponding to the previously described sequence in the leg's workspace. This part of the controller takes as input the desired motion in terms of toe position vectors w.r.t. the corresponding hip-fixed frame, and, gives the desired joint angles as output. To this end, we derive the inverse kinematics of a virtual two-segment leg shown in Fig. 2; the second and the third segment, which are connected with a spring, compose a virtual segment to simplify the three-segment inverse kinematics.

The toe coordinates are the ones given by the inverse kinematics only when the ankle spring is in its equilibrium, or when the ankle spring is very stiff. However, this issue does not affect the functionality of the controller, since the precise tracking of the elliptic trajectory is not a strict goal. Referring to Fig. 2, while still avoiding adding notation for different legs, the joint angles for each leg are given by:

$$\begin{aligned} \varphi_{1,des} &= \pi / 2 + \theta_{inv,1} - \theta_b \\ \varphi_{2,des} &= \theta_{inv,2} \end{aligned} \quad (6)$$

where  $\varphi_{1,des}$  and  $\varphi_{2,des}$  are the relative hip angle and the relative knee angle respectively, and  $\theta_{inv,1}$ ,  $\theta_{inv,2}$  the angles computed from the two-segment leg inverse kinematics as:

$$\begin{aligned} \theta_{inv,1} &= a \tan 2(y_{toe}, x_{toe}) - a \tan 2(l_{v2} s_{\theta 2}, l_{v1} + l_{v2} c_{\theta 2}) \\ \theta_{inv,2} &= a \tan 2(s_{\theta 2}, c_{\theta 2}) \end{aligned} \quad (7)$$

with:

$$\begin{aligned} c_{\theta 2} &= \cos(\theta_{inv,2}) = (x_{toe}^2 + y_{toe}^2 - l_{v1}^2 - l_{v2}^2) / (2l_{v1}l_{v2}) \\ s_{\theta 2} &= \sin(\theta_{inv,2}) = \pm \sqrt{1 - c_{\theta 2}^2} \end{aligned} \quad (8)$$

In (7) and (8),  $x_{toe}$ ,  $y_{toe}$  are the toe coordinates w.r.t. the hip-fixed frame, which are assigned with the values of  $x_{ir}$ ,  $y_{ir}$  resulting from the trajectory planning part, while  $l_{v1}$ ,  $l_{v2}$  are the lengths of the virtual two-segment leg. The signs in (8) correspond to different knee configurations.

##### Active Compliance Joint Control

This is the key part of the low-level controller, whose main role is to ensure the desired compliant behavior during interaction with the ground in a controllable manner via software. The desired joint angles are passed to this active compliance controller, with which the whole motion derives from virtual spring-damper systems acting between the current joint angles and the desired ones, as shown in Fig. 2, using control torques computed as:

$$\tau_1 = -k_p (\varphi_1 - \varphi_{1,des}) - k_v \dot{\varphi}_1 \quad (9)$$

$$\tau_2 = -k_p (\varphi_2 - \varphi_{2,des}) - k_v \dot{\varphi}_2 \quad (10)$$

where  $k_p$ ,  $k_v$  are tunable gains that actually correspond to the stiffness and damping coefficients of two virtual springs located at the hip and knee joints. This is the simplest indirect force control scheme that could be applied to the joints, and it has a decentralized character – interaction and cou-

pling effects between the joints are considered as disturbances. In this way, any joint stiffness can be selected on the fly, adapting to a wide range of situations, if the actuation system permits it. Examples of related schemes that have appeared in literature are: active compliance control, [15], and threshold control [16]. As a result, the elliptical path is not followed strictly, and this is why we call it virtual. Note that the passive compliance located at each ankle also contributes in this deviation from the reference elliptical trajectory. Also, the only variables we need to measure or estimate in the control loop, are the body pitch angle, and the motor angles and their rates.

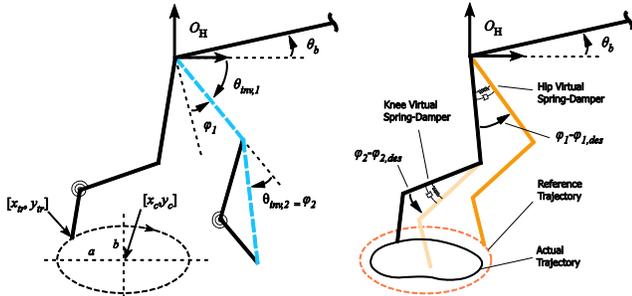


Figure 2. Left: The virtual two-segment leg (blue), used in the inverse kinematics algorithm, and the virtual toe-trajectory (elliptical primitive) used for trajectory planning, Right: The actuators act as virtual spring-damper systems that drive the leg through a sequence of desired configurations (orange); the ellipse is not followed in a strict manner.

### B. High-Level Controller

The role of the high-level controller is to properly tune the various control parameters that are used by the low-level controller in order to achieve stable gaits, see Fig. 3. The way that animals move can help in designing a biomimetic controller that achieves stable running following a desired speed profile. It has been measured that the cheetah increases both *stride frequency* and *stride length* in order to reach higher forward velocities, while the greyhound only increases its stride length maintaining a constant stride frequency, [18]. Although these are data for galloping, we assume data for trotting cannot be much different. This indicates that a quadruped can control its forward velocity in various ways, and it finally chooses the method that better suits its design, including its bone structure and its actuation system.

While the low-level controller already presented can be used with multiple strategies and with many combinations of the control parameters, here, one of the simplest strategies is proposed, which can generate dynamically stable locomotion through the adjustment of a single control parameter during the whole gait. The main idea is to regulate the robot’s forward velocity by adjusting the current leg step size, i.e. how close this is to a large fixed leg step, while doing this in a compliant way. We show that a controller of this type can achieve stable trotting, by only regulating the virtual stiffness of all actuated joints, i.e. the control gain  $k_p$  of the active compliance controller, while maintaining the virtual toe trajectories and the stride frequency fixed at their

maximum values. We call the gain  $k_p$  the *system virtual stiffness*, since it indicates how compliant the actuated joints are to the commanded behavior. The gain  $k_v$  contributes to the stability of the whole motion and is considered a function of  $k_p$ . The higher the *system virtual stiffness* is, the closer to the “max step” trajectories the actual toe trajectories get, and the stiffer the legs’ motion become against ground forces, thus increasing the stride length of the gait. This strategy, which mimics the way a greyhound controls its forward velocity, [18], can generate walking and running gaits, presenting stable fixed points at various forward velocities. Although this scheme requires tuning of several control parameters, this is a rather easy task with many solutions due to the inherent stability and robustness characterizing the low-level controller. Results from simulations that were conducted to prove these claims will be presented next.

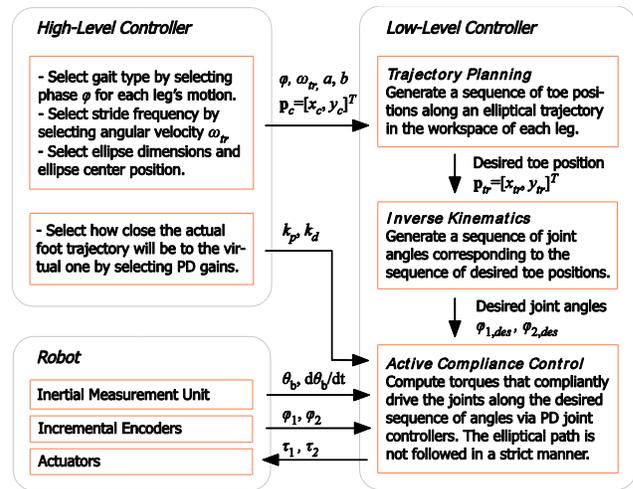


Figure 3. Block diagram of the controller for a single leg.

## IV. SIMULATION EXPERIMENTS

In this section, we study the controller stability using Poincaré maps, and show the robot response as a function of the system virtual stiffness.

### A. Simulation Environment

The EoM derived in Section II were solved numerically in our simulation environment built in Matlab, in which we used the ODE15s solver, with absolute tolerance  $10^{-5}$ , relative tolerance  $10^{-4}$ , and maximum step  $10^{-4}$ .

### B. Model Parameters

The model parameters were selected in order for the results to be applicable in most legged robots developed in academia and industry. Taking inspiration from the cat bone structure, [17], we used a biomimetic configuration, in which the fore and the hind knees point inwards, see Fig. 1. Although cats have longer hind than fore legs, equal lengths were used, since most robots use the same actuators and designs for all legs. Being again close to biological data and without loss of generality, we chose equal lengths of 0.25m

for the upper two segments of each leg, and a slightly smaller length of 0.15m for the third segment. The lengths and the last segment's orientation were selected such that the points of ground contact and the two hips almost form a square in the sagittal plane when the robot stands in a resting position, as biology suggests, [17].

A realistic body mass of  $m_b = 20\text{kg}$  was considered, with a body length of  $l_b = 0.6\text{m}$ . The masses of the three leg segments were  $m_1 = 0.4\text{kg}$ ,  $m_2 = 0.3\text{kg}$ , and  $m_3 = 0.15\text{kg}$ . The MoI of each body was calculated considering the main body a rectangle, and the leg segments simple rods, yielding  $I_b = 1.3\text{kgm}^2$ ,  $I_1 = 2 \cdot 10^{-3}\text{kgm}^2$ ,  $I_2 = 15 \cdot 10^{-3}\text{kgm}^2$  and  $I_3 = 2.8 \cdot 10^{-3}\text{kgm}^2$ . The ankle spring-damper coefficients were set to  $k_a = 300\text{Nm/rad}$ , and  $b_a = 2\text{Nsm/rad}$ , and the viscous coefficient for every other joint was set to  $b_j = 0.3\text{Nsm/rad}$ . A stiff terrain with  $k_g = 400,000\text{N/m}$ , and  $b_g = 120,000\text{Ns/m}$  was finally considered, with a kinematic friction coefficient  $\mu_c = 0.8$  and a static coefficient  $\mu_s = 0.9$ . The velocity threshold and the Stribeck velocity were  $u_e = 10^{-3}\text{m/s}$ , and  $u_s = 10^{-2}\text{m/s}$  respectively. Motor characteristics were not included, in order to get a better understanding of the system's needs, before imposing saturation limits and properties valid only for a specific actuation system (electrical, hydraulic or other).

### C. Simulation Results

To show the inherent stability and robustness of the low-level controller, in the first experiment we started the model from stance with all the control parameters fixed:  $a = 0.2\text{m}$ ,  $b = 0.07\text{m}$ ,  $x_c = -0.1\text{m}$ ,  $y_c = -0.4\text{m}$ ,  $\omega_r = -22\text{rad/s}$ ,  $k_p = 250\text{Nm/rad}$  and  $k_v = 50\text{Nsm/rad}$ . After the forward velocity converged to  $1\text{m/s}$ , we suddenly changed the virtual stiffness of the joints  $k_p$  to  $450\text{Nm/rad}$  making the robot increase its forward velocity to  $1.7\text{m/s}$ , as shown in Fig. 4(a). Figs 4(d) and 4(e) show the two locally stable limit cycles that appeared in the phase portrait of the body height, and Fig. 4(f) the two corresponding fixed points of the Poincaré section taken at the apex height of each stride. Fig. 4(c) illustrates the footfall pattern of the gait corresponding to the second fixed point, as a graph in the style of Hildebrand; obviously, the gait is a running trot. We note that this is an example out of the numerous fixed points identified.

In this experiment, the model was able to reject undesired initial perturbations (it started from stance and converged to a stable running trot of  $1\text{m/s}$  in  $1\text{s}$ ), and to move easily to another fixed point (transition to a stable running trot of  $1.7\text{m/s}$ ). The results indicate that the controller exhibits inherent stability and robustness, coming into view through large regions of attraction around the various fixed points found for simple changes in the control parameters.

These results can lead to the hypothesis that a relationship exists between the system virtual stiffness  $k_p$  and the robot's forward velocity, if the other parameters are properly tuned. Based on this hypothesis, we conducted a second experiment, in which the system virtual stiffness linearly varied from  $k_p = 100\text{Nm/rad}$  to  $800\text{Nm/rad}$ , and then back to  $100\text{Nm/rad}$ , aiming at controlling a complete scenario of

acceleration, stable running and deceleration. The rest of the parameters were equal to those of the first experiment, except for the ellipse semi-axes that were set at  $a = 0.13\text{m}$ , and  $b = 0.05\text{m}$ . For the transient phases before  $t = 1\text{s}$  and after  $t = 11\text{s}$ , the robot was commanded to stay in a resting posture using simple joint controllers.

As shown in Figures 5(a) and 5(b), the model successfully performed the expected motion, beginning with a walking trot (no white areas in Fig. 5(a) for  $t < 1.8\text{s}$ ), reaching a stable running trot of  $1.5\text{m/s}$ , and decelerating again back to stance. The torques exerted on FR and HR hips and knees are plotted in Figures 5(c) and 5(d), showing increased values while accelerating. Figure 6 illustrates the FR and HR toe trajectories, showing that for higher forward velocities the actual trajectories approach the fixed virtual ones.

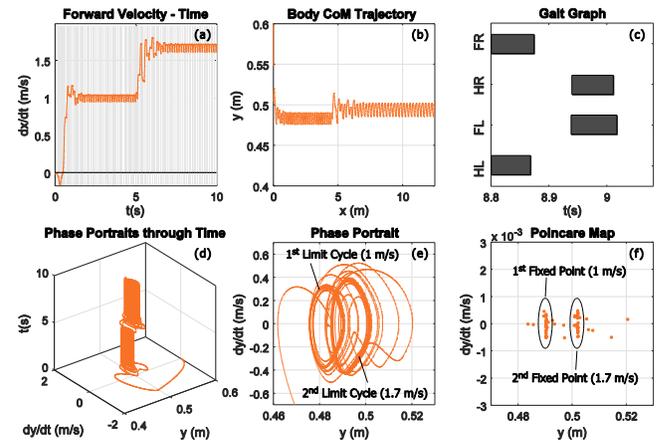


Figure 4. Experiment 1: (a) the body's forward speed, (b) the body CoM trajectory, (c) the gait graph for the second fixed point, showing stance phases in one stride, (d) the body height phase portraits through time, (e) the two limit cycles on the body height phase portrait, (f) the two corresponding fixed points of the Poincaré section taken at the apex height of each stride.

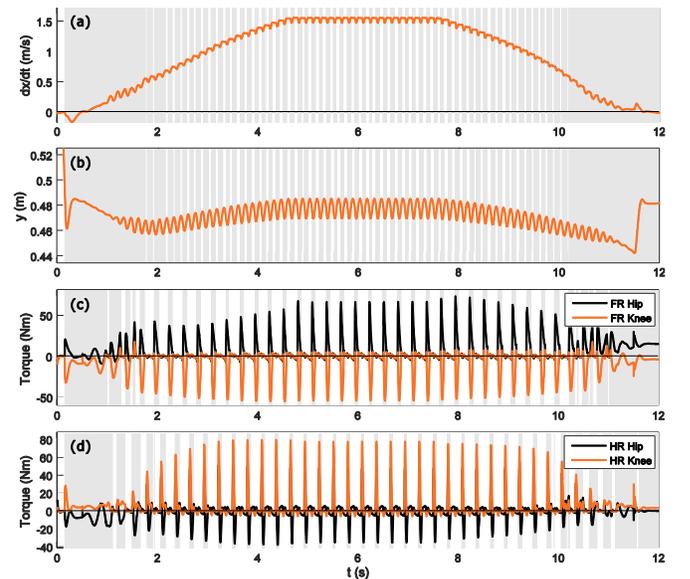


Figure 5. Experiment 2: (a) The body's forward velocity, (b) the body CoM trajectory, (c) the torques exerted on FR joints, and (d) on HR joints, gray areas denote phases when at least one leg contacts the ground.

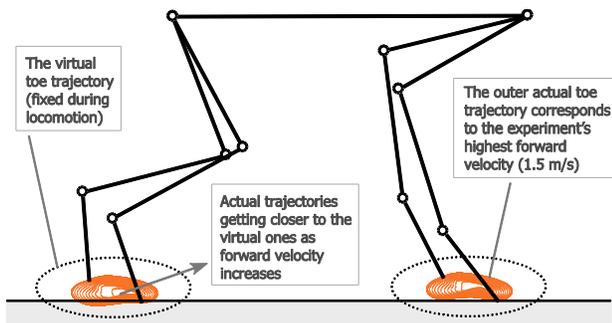


Figure 6. A snapshot from the second experiment, with the FR and HR actual and virtual toe trajectories plotted w.r.t. the moving hip-fixed frames.

#### D. Discussion

The results showed that the proposed framework can generate a wide range of stable motions, from slow walking gaits to more dynamic ones. Although some tuning was required, the proposed scheme provided a simple and intuitive way to find multiple stable fixed points spanning a wide range of forward velocities, and to smoothly move between them by properly adjusting the system virtual stiffness.

Among several other observations made throughout the experiments, it was evident that the higher the stride frequency, the easier was to stabilize the system, and the more robust the gait was, mainly due to the more frequent interaction with the ground. It was also apparent that the abscissa of the ellipse center  $x_c$  w.r.t. to a leg's hip-fixed frame  $O_H$  should be negative; this way the frictional forces pointed forward for a larger percentage of the stance phase, minimizing the deceleration in each cycle. The reference trajectory played a major role in performance, since, combined with the virtual stiffness of the joints, they produced the necessary torques for driving the robot. In general, the trajectories should lie close to the boundaries of the leg's workspace, in order to take the most from the robot's structure; how close the actual trajectories are, is defined by the system virtual stiffness. As a comment on the effect of the virtual damping  $k_v$ , we observed that the controller worked well for values close to  $k_v = 2\sqrt{k_p}$ , a relation that results in a damping ratio of  $\zeta = 1$  in a simple second order system. Furthermore, although the use of passive springs at the ankles is not required by the controller, we found out that the additional compliance helped stabilizing the gait and smoothed the robot's performance. In fact, most robots use springs for shock absorption and energy storing. It was observed that stiffer springs allowed for faster gaits.

In conclusion, it was shown that the proposed control parameters can be combined in multiple ways producing a rich repertoire of gaits. The ranges of the parameters leading to stable gaits remain to be discovered, and closed loop strategies (e.g. following a desired speed profile) are left to be designed on the basis of this framework.

#### V. CONCLUSION

In this paper, a compact control scheme was proposed, consisting of a toe trajectory planning part and an active

compliance part driving each leg, leading to stable dynamic trotting at a wide range of forward velocities. The notion of the system virtual stiffness was introduced as a single parameter tuned to drive a quadruped through a complete gait scenario, with smooth acceleration, constant speed locomotion, and deceleration. The proposed ideas were validated through simulation experiments with a 2D biomimetic robot.

#### REFERENCES

- [1] H.-W. Park, S. Park and S. Kim, "Variable-speed quadrupedal bounding using impulse planning: Untethered high-speed 3D Running of MIT Cheetah 2," *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.
- [2] C. Semini, V. Barasuol, T. Boaventura, M. Frigerio, M. Focchi, D. G. Caldwell and J. Buchli, "Towards versatile legged robots through active impedance control," *The International Journal of Robotics Research*, 2015.
- [3] "http://spectrum.ieee.org/automaton/robotics/robotics-hardware/spot-is-boston-dynamics-nimble-new-quadruped-robot"
- [4] M. H. Raibert, *Legged robots that balance*, MIT press, 1986.
- [5] S. G. Roh and J. H. Park, "Gallop with speed change for quadruped robots," *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2011.
- [6] V. Barasuol, V. J. De Negri and E. R. De Pieri, "WCPG: A Central Pattern Generator for Legged Robots Based on Workspace Intentions," *Dynamic Systems and Control Conference and Bath/ASME Symposium on Fluid Power and Motion Control*, 2011.
- [7] A. Mutka, P. Frano, T. Reichenbach and Z. Kovacic, "Elliptical motion method for robust quadrupedal locomotion," *IEEE International Conference on Control Applications (CCA)*, 2012.
- [8] I. Poulakakis, E. Papadopoulos, and M. Buehler, "On the Stability of the Passive Dynamics of Quadrupedal Bounding," *The International Journal of Robotics Research*, vol. 25, no. 7, pp. 669-688, 2006.
- [9] A. Jafari, N. Tsagarakis and D. G. Caldwell, "Energy efficient actuators with adjustable stiffness: a review on AwAS, AwAS-II and ComACT VSA changing stiffness based on lever mechanism.," *Industrial Robot: An International Journal*, vol. 42, no. 3, 2015.
- [10] D. J. Hyun, S. Sangok, L. Jongwoo and K. Sangbae, "High speed trotting: Implementation of a hierarchical controller using proprioceptive impedance control on the MIT Cheetah," *The International Journal of Robotics Research*, vol. 33, no. 11, pp. 1417-1445, 2014.
- [11] G. Gilardi, and I. Sharf, "Literature survey of contact dynamics modeling," *Mechanism and machine theory*, vol. 37, no. 10, pp. 1213-1239, 2002.
- [12] D. W. Marhefka and D. E. Orin, "A compliant contact model with nonlinear damping for simulation of robotic systems," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 29, no. 6, pp. 566-572, 1999.
- [13] Bo, Li Chun, and D. Pavelescu, "The friction-speed relation and its influence on the critical velocity of stick-slip motion." *Wear* 82, no. 3 (1982): 277-289.
- [14] Barasuol, Victor, et al. "A reactive controller framework for quadrupedal locomotion on challenging terrain," *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2013.
- [15] Siciliano, B., et al, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2009.
- [16] Feldman, A. G., & Levin, M. F. "The equilibrium-point hypothesis—past, present and future," *In Progress in Motor Control* (pp. 699-726). Springer US, 2009.
- [17] Day, L. M., & Jayne, B. C. "Interspecific scaling of the morphology and posture of the limbs during the locomotion of cats (Felidae)." *Journal of Experimental Biology*, 210(4), 642-654, 2007
- [18] Hudson, P. E., et al, "High speed galloping in the cheetah (*Acinonyx jubatus*) and the racing greyhound (*Canis familiaris*): spatio-temporal and kinetic characteristics." *The Journal of experimental biology*, 215(14), 2425-2434, 2012.