2012 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 7-12, 2012. Vilamoura, Algarve, Portugal

Controllers for Robust Hopping with Upright Trunk based on the Virtual Pendulum Concept

Maziar Ahmad Sharbafi, Christophe Maufroy, H. Moritz Maus, Andre Seyfarth,
Majid Nili Ahmadabadi and Mohammad Javad Yazdanpanah

Abstract— **This paper presents a new control approach to achieve robust hopping with upright trunk in the sagittal plane. It relies on an innovative concept for trunk stabilization, called Virtual Pendulum concept, recently proposed, based on experimental finding in animal locomotion. With this concept, the trunk is stabilzed by redirecting the ground reaction force to a virtual support point, named Virtual Pivot Point (VPP). This concept is combined with a new leg adjustment scheme to induce stable hopping when an extended trunk is added to SLIP model. The stability is investigated by Poincaré map analysis. With fixed VPP position, stability, disturbance rejection and moderate robustness are achieved, but with low convergence speed. To improve the performances and attain higher robustness, event based control of VPP position is introduced, using feedback of the system state at apex. Dead beat control and Discrete LQR are alternatively considered to adjust the feedback gains. In both cases, considerable enhancements with respect to stability, convergence speed and robustness against perturbations are acheieved.**

I. INTRODUCTION

The ability to perform efficient and robust locomotion is a crucial condition for the massive use of legged robots in real world applications. In that respect, robots can learn from animals, if the principles underlying locomotion in biological legged systems can be transfered to their articificial counterparts. A great progress in this direction was introducing simple models, coined “templates” [1], able to represent the overall dynamics of animal gaits. One of the most famous models is spring-loaded inverted pendulum (SLIP) [2][3] which consists of a point mass atop a massless spring and provides a good description of human gaits, such as walking [4], hopping and running [2]. Despite its high level of abstraction, it supported and inspired the development of successful legged robots [5][6] or was used as explicit targets for control [7], over the years.

However, as the upper body is represented by a point mass, stabilization of the upright posture (or *posture control*) in robots or humans cannot be addressed with this template. For that purpose, the SLIP must be extended to include a model of the upper body. With such models, various methods for posture control have been proposed. However, most of them rely on the same principle, i.e. the feedback control of the trunk orientation with respect to an absolute referential frame [5][7][8]. Recently, Maus et al. [9] proposed an innovative concept for posture stabilization, coined Virtual Pendulum (VP), based on observations in a variety of animals including humans. It relies on the creation of a virtual support point (VPP), placed above the center of mass. This is achieved by redirecting the ground reaction force (GRF) vector towards this point, needles to know the trunk absolute orientation. This approach was validated in simulations, where it yielded stable upright walking and running patterns [10].

In this paper, we apply this concept to achieve robust hopping, defined by running with zero forward velocity. In contrast to running [11], hopping cannot be stabilized by placing the leg at a given fixed angle with the ground. Hence, we have to introduce a control layer adjusting the leg angle during the swing phase which was not considered in former studies dealing with VP-based control [10][9].

In previous implementations of the VP concept, the VPP was always fixed in the referential frame attached to the trunk. In this way, posture could be stabilized but the convergence to steady state is slow and the robustness against perturbations is only moderate. Previous investigations also showed that placing the VPP out of the trunk axis could be used for maneuver [10] or compensation of energy losses [12]. In the present study, we use a similar approach to solve the issues regarding disturbance rejection and robustness. With the VP concept, the complex process of generating a suitable hip torque pattern is simplified to the specification of the position of one point. Hence, it is particularly suited for the application of event-based control. It is performed by a feedback law using the states at the apex event in order to adjust the VPP position during the next stance phase. Two design strategies are described in Sct. II, beside simulation model and a short presentation of the VP concept. In Sct.III, the results are presented for different controllers. Finally, Sct.IV discusses the benefits and costs of the extension of the original approach and gives an outline of the future work.

II. METHODS

A. Simulation model

Our simulation model (TSLIP[1] for Trunk-SLIP) is an extension of the traditional SLIP, where the point mass is replaced by a trunk, as represented in Fig.1-left. The model parameters (given in Table I) are set to match the characteristics of a human.

M. A. Sharbafi, M. Nili and M. J. Yazdanpanah are with College of Engineering, School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran, {sharbafi, yazdan, mnili}@ut.ac.ir

M. A. Sharbafi, C. Maufroy, H.M. Maus and A. Seyfarth are with Lauflabor Locomotion Laboratory, Technical University of Darmstadt, Darmstadt, Germany, {sharbafi, cmaufroy, mmaus, seyfarth}@sport.tu-darmstadt.de

[1] In [13] a similar model was introduced namely ASLIP, for “Asymmetric SLIP”. However, as this term can also designate a SLIP model with asymmetric leg properties, we prefer to use the appellation TSLIP.

978-1-4673-1736-8/12/\$31.00 ©2012 IEEE

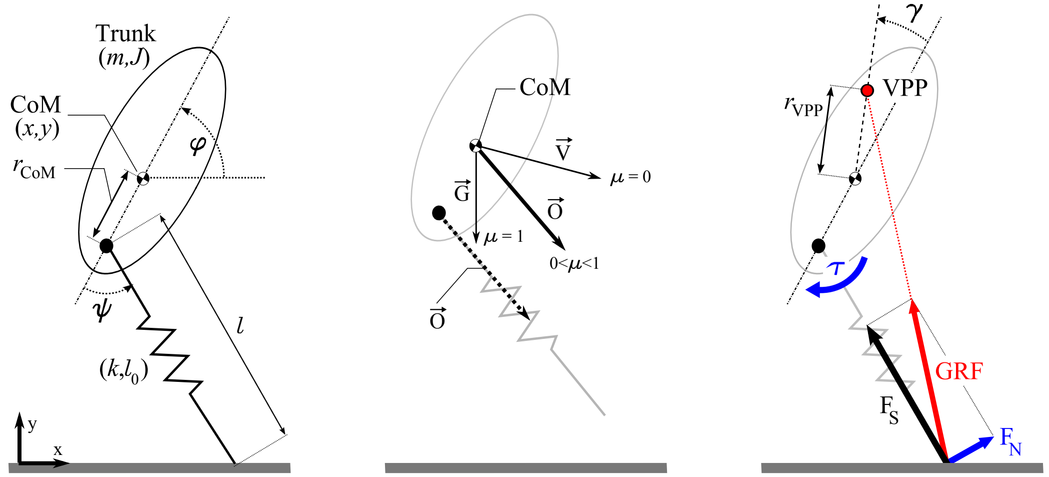

Fig. 1. **Left**: TSLIP model with a rigid trunk and a leg modeled as a massless prismatic spring with parameters of Table I. **Middle**: Velocity-based leg adjustment (VBLA) during flight phase. **Right**: Virtual pendulum-based posture control (VPPC) during stance phase.

During the stance phase, the force generated along the leg axis is given by: $F_S = k\ (l_0 - l)$, where l, l_0 and k are respectively the current leg length, leg rest length and the spring stiffness (for convenience, we define the dimensionless leg stiffness $\tilde{k} = kl_0/mg$ that is used hereafter instead of k). A torque can be applied at the hip to stabilize the posture of the trunk. Liftoff occurs when the leg reaches its rest length. During the flight phase, the leg orientation can be arbitrarily adjusted (as the leg is massless).

B. System analysis

The model is investigated using Poincaré return map analysis. Assume that x, y and φ are respectively the center of mass (CoM) horizontal and vertical positions and the trunk orientation. Then, the event used for the Poincaré section is the apex, characterized by $\dot{y} = 0$ with $\ddot{y} < 0$. Using this definition, the system state at apex is described with the reduced[2] state vector: $\mathbf{S} = [y, \varphi, \dot{x}, \dot{\varphi}]$. The Poincaré return map $\mathbf{F}$ between two consecutive apices is thus defined by $\mathbf{S}_{k+1} = \mathbf{F}(\mathbf{S}_k)$. Periodic hopping motions correspond to the fixed points of $\mathbf{F}$ (i.e. $\mathbf{S}^*$ such that $\mathbf{S}^* = \mathbf{F}(\mathbf{S}^*)$) with general form: $\mathbf{S}^* = [y^*, 90°, 0, 0]$. Hereafter, nominal hopping height $y^* = 1.125$ m (i.e. the foot is 2.5 cm above the ground at apex), which corresponds to the preferred hopping height in human [14], is considered for hopping motion. The local stability of a periodic motion is investigated by computing the eigenvalues λ_j of the Jacobian matrix $A = \frac{\partial \mathbf{F}}{\partial \mathbf{S}}(\mathbf{S}^*)$, evaluated numerically. As the leg is a perfect spring, the periodic motion is always neutrally stable w.r.t hopping height (y^*) changes, characterized by an eigenvalue equal to 1. We consider that the periodic motion is stable if the other eigenvalues are smaller than 1.

TABLE I
MODEL PARAMETERS

fixed parameters	symbol	value [units]
trunk mass	m	80 [kg]
trunk moment of inertia	J	4.58 [kg m^2]
distance hip-CoM	r_{CoM}	0.1 [m]
leg rest length	l_0	1 [m]
variable parameters	symbol	[units]
leg stiffness	$\tilde{k}$	[-]
distance CoM-VPP	r_{VPP}	[m]
leg adjustment parameter	μ	[-]
VPP angle	γ	[deg]

The transient behavior of the system is investigated by applying perturbations of the horizontal velocity and the trunk angle to the model. The robot is dropped from the nominal hopping height with initial horizontal speed $\dot{x}_0 \neq 0$ and/or trunk angle $\varphi_0 \neq 90°$. The same approach is used to evaluate the robustness of the stable hopping motions. Each type of perturbation is then applied separately and the tolerable amounts are computed as follows:

- $\Delta\dot{x}_{max}$: maximum $\dot{x}_0$ that the controller is able to reduce to less than 0.05 m/s in 50 steps.
- $\Delta\varphi_{max}$: maximum $\Delta\varphi_0 = |\varphi_0 - 90°|$ that the controller is able to reduce to less than $\Delta\varphi_0/10$ in 50 steps.

C. Leg adjustment during the swing phase

Unlike running [11] and walking [4], the simplest leg placement strategy (i.e. with a given pre-defined angle of attack with respect to the ground) cannot yield stable hopping motion. This requires the adjustment of the leg orientation during the flight phase. Most leg adjustment strategies rely on sensory information about the CoM velocity, following the approach pioneered by Raibert in which the foot landing position is adjusted based on the horizontal velocity [15]. Recently, various strategies were investigated by Peuker et al. [16]. The most robust strategy for hopping and running with SLIP model, was placing the leg with respect to both the CoM velocity and the gravity. In this paper, a modified version of this strategy is used: the leg direction is given by

[2] The absolute horizontal position x is omitted because it does not influence the evolution of the system from one apex event to the next.

vector $\vec{O}$ as a weighted average of the CoM velocity vector $\vec{V}$ (in dimensionless form) and the unitary gravity vector $\vec{G}$.

$$\begin{array}{rcl}\vec{V} & = & [v_x, v_y]^T/\sqrt{gl}\ ;\ \vec{G} = [0, -1]^T \\ \vec{O} & = & (1-\mu)\vec{V} + \mu\vec{G}\end{array} \quad (1)$$

The portion of each vector is determined by coefficient $0 < \mu < 1$ (see Fig.1-center). When $\mu = 0$, the leg is parallel to the CoM velocity vector and, for $\mu = 1$, the leg is exactly vertical. In the rest of the paper, we will refer to this strategy as the Velocity-Based Leg Adjustment or VBLA.

D. Virtual pendulum concept for postural stabilization

The key idea of the VP concept is to create a point of virtual support (virtual pivot point or VPP) located above the CoM by redirecting the GRF vector towards this point, at each instant of stance phase. Hence, the trunk behavior is transformed, from an inverted pendulum mounted at the hip to a regular virtual pendulum suspended at the VPP (see [9] for details). The VP concept is implemented in the simulation model by applying a torque (τ) at the hip during the stance phase (Fig.1-right). This generates a force perpendicular to the leg axis (F_N) and the torque needed to redirect the GRF towards the VPP can be computed by:

$$\tau = F_S\ l\ \frac{r_{\text{CoM}}\ sin\ \psi + r_{\text{VPP}}\ sin(\psi - \gamma)}{l + r_{\text{CoM}}\ cos\ \psi + r_{\text{VPP}}\ cos(\psi - \gamma)} \quad (2)$$

We call this approach: *Virtual Pendulum based Posture Control* or VPPC. As shown in Eq. 2, it does not require information about the absolute trunk orientation φ (only the force F_S and the leg orientation w.r.t. the body ψ are needed).

1) VPPC with fixed VPP position (VPPC-FP): In the original implementation of the VPPC [9], the VPP position is held constant. Since trunk posture in hopping should be upright, from the concept of virtual pivot, γ must be zero, i.e. the VPP must lie on the trunk axis. Hence, only r_{VPP} is used to characterize the VPP position. We refer to this controller as VPPC-FP.

2) VPPC with Event-based Control: To improve the performance and robustness of the hopping motion, we introduce event-based control. In this method, at each apex, the VPP position is adapted for the next stance phase using the current system state. To design the controller, the Poincarre return map is linearized around a nominal fixed point S^*, while considering r_{VPP} and γ as inputs:

$$\Delta\mathbf{S}_{k+1} = \mathcal{A}_S\ \Delta\mathbf{S}_k + \mathcal{B}_\gamma\ (\gamma_k - \gamma^*) + \mathcal{B}_r\ (r_{\text{VPP}} - r^*_{\text{VPP}}) \quad (3)$$

with:

$$\Delta\mathbf{S}_k = \mathbf{S}_k - \mathbf{S}^* = [\Delta y_k, \Delta\varphi_k, \Delta\dot{x}_k, \Delta\dot{\varphi}_k] \quad (4)$$

$$\mathcal{A}_S = \frac{\partial\mathbf{F}}{\partial\mathbf{S}}(\mathbf{S}^*, \gamma^*, r^*_{\text{VPP}}) \quad (5)$$

$$\mathcal{B}_\gamma = \frac{\partial\mathbf{F}}{\partial\gamma}(\mathbf{S}^*, \gamma^*, r^*_{\text{VPP}})\ ;\ \mathcal{B}_r = \frac{\partial\mathbf{F}}{\partial r_{\text{VPP}}}(\mathbf{S}^*, \gamma^*, r^*_{\text{VPP}}) \quad (6)$$

Following the same argument as for the VPPC with fixed VPP position, the nominal VPP angle γ^* must be equal to 0. Additionally, the vector $\mathcal{B}_r$ is also equal to 0. Indeed, varying r_{VPP} has no effect on the motion when the trunk is exactly vertical (the model configuration corresponding to the fixed point). The remaining matrixes $\mathcal{A}_S$ and $\mathcal{B}_\gamma$ have the following form:

$$\mathcal{A}_S = \begin{pmatrix} 1 & 0 \\ 0 & A \end{pmatrix} \quad ; \quad \mathcal{B}_\gamma = \begin{pmatrix} 0 \\ B \end{pmatrix} \quad (7)$$

where A and B are respectively [3×3] and [3×1] matrixes. This indicates that, in the first order, the evolution of the vertical position is decoupled (hence not controllable using γ_k) and neutrally stable. Defining $\mathbf{X}_k = [\Delta\varphi_k, \Delta\dot{x}_k, \Delta\dot{\varphi}_k]$, we can rewrite the linear system as:

$$\left\{\begin{array}{rcl}\Delta y_{k+1} & = & \Delta y_k \\ \mathbf{X}_{k+1} & = & A\ \mathbf{X}_k + B\ \gamma_k\end{array}\right. \quad (8)$$

If the pair (A, B) is controllable, the evolution of $\mathbf{X}_k$ can be controlled by using a linear state feedback law for γ_k:

$$\gamma_k = -K\ \mathbf{X}_k \quad (9)$$

Two methods, investigated in this paper to select the coefficients of K are described in the following.

a) Dead-beat control (VPPC-DB): For discrete linear systems, the control policy that brings the output to the steady state in the minimum number of steps is called dead-beat control. It is achieved when all the poles of the transfer function are at the origin. We use the same approach here by choosing the elements of K to set all the eigenvalues of the system matrix $A - BK$ to zero.

b) Discrete LQR (VPPC-LQR): Dead-beat control aims at cancelling the disturbance as fast as possible, but does not consider the behavior of the system during the transient phase. Hence, with such approach, a smooth convergence of the state variables to their steady state values is not guarantied. This can be achieved by using optimal feedback control. Here, we use Discrete LQR [17], with a cost function J that only consider the output error:

$$J = ||W\mathbf{X}_n||_2 = \sum_{n=1}^{\infty} \mathbf{X}_n^T Q \mathbf{X}_n \quad (10)$$

where W is a diagonal weight matrix and $Q = W^T W$. The optimal gain vector K is given by Eq. 11, in which P is a symmetric positive definite matrix, the solution of the discrete Riccati Algebraic Equation (Eq.12).

$$K = (B^T P B)^{-1} B^T P A \quad (11)$$

$$P = Q + A^T(P - PB(B^T P B)^{-1} B^T P)A \quad (12)$$

The weight matrix W can be used to give more importance to the behavior of some of state variables.

III. RESULTS

In this section, the stability and robustness of different aforementioned controllers are compared. As a standard platform, TSLIP dynamical model for hopping with parameters of Table I is simulated in MATLAB/SIMULINK 2011b using ode45 solver. The main analyzing method for stability is investigating eigenvalues of Poincaré return map.

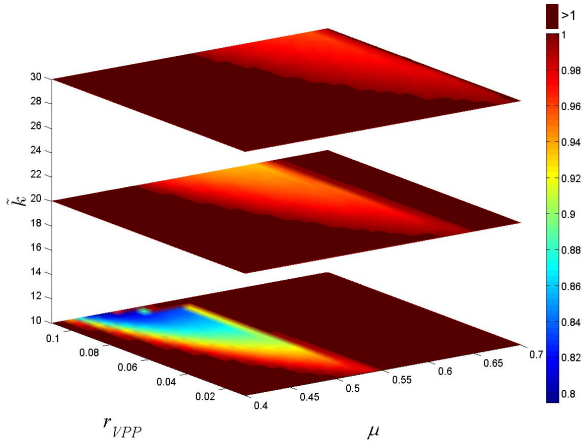

Fig. 2. Stability of the hopping motion with nominal hopping height for VPPC-FP with different parameter values. The color bar shows the magnitude of λ_{max}.

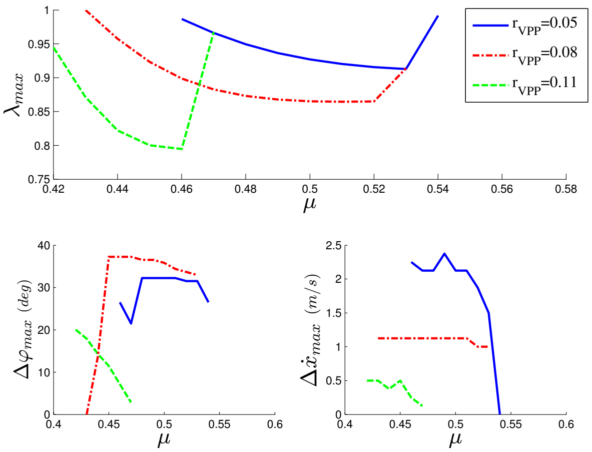

Fig. 3. Stability and robustness for VPPC-FP with $\tilde{k} = 10$. The results are represented for three values of r_{VPP}.

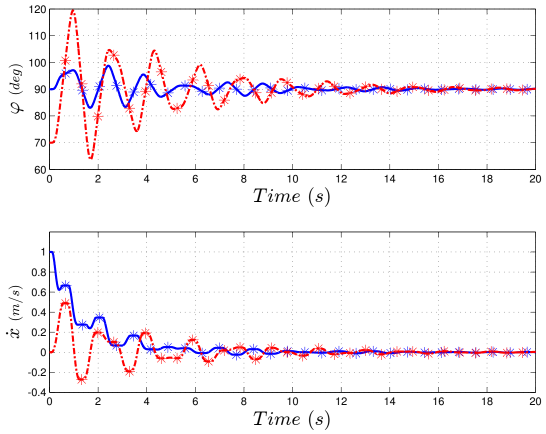

Fig. 4. Responses for VPPC-FP with $[\tilde{k}, \mu, r_{\text{VPP}}] = [10, 0.5, 0.08]$ after perturbations of the initial horizontal velocity ($\dot{x}_0 = 1$ m/s, dashed line) and the initial trunk angle ($\varphi_0 = 70°$, solid line). Apex events are displayed with the $*$ symbol.

Additionally, characteristics of time response are employed in order to evaluate the robustness.

A. VPPC with fixed VPP position (VPPC-FP)

To find stable hopping motions with sufficient robustness, the parameter space is explored. The controller parameters are limited to μ and r_{VPP}, when in hopping with fixed VPP, γ must be zero (from the concept of virtual pivot). We also consider $\tilde{k}$ as a variable parameter, that could be adjusted to improve the hopping performance.

First, the stability of the controlled system with nominal hopping height (as defined with y^* in Sct. II-B), is investigated. For that purpose, we consider the magnitude of the largest effective (not neutral) eigenvalue (λ_{max}). In Fig. 2, the results for three stiffness values are represented. It shows that stable hopping is achieved in a large parameter region. However, these motions associate with relatively high eigenvalues (greater than 0.8) which is not desirable.

Next, we evaluate the robustness of the hopping motions with the stiffness value $\tilde{k} = 10$ for which a large stable region with the highest convergence rate (i.e. the lowest eigenvalues) is found. The results are represented in Fig. 3, for different values of the two other parameters (r_{VPP} and μ). It shows that there is a trade-off between performance and robustness, as choosing r_{VPP} to reduce the settling time (smaller eigenvalues) results in lower robustness.

To illustrate the response of the system after perturbation, we select $r_{\text{VPP}} = 0.08$ m and $\mu = 0.5$ which makes a compromise between convergence rate ($\lambda_{Max} = 0.865$) and robustness. Fig. 4 displays the behavior of the trunk angle and horizontal speed for two different perturbations. As expected from the magnitude of λ_{max}, the convergence to the steady state is slow. Oscillations of the trunk angle subsists up to 15 s after the occurrence of the perturbation.

B. VPPC with Event-based Control

Event-based control allows to improve the system behavior by placing the poles of the matrix $A - BK$ in desired positions, if the pair $[A, B]$ is controllable. The latter condition is satisfied in whole of the parameter space, considered in Fig. 2. Hence, the stability of the hopping motion does not depend on the selection of specific model parameters and can be guarantied by proper controller design. It means that some parameters like stiffness could be adjusted to satisfy other performance indexes like energy consumption. Additionally, because the response to perturbations is very less sensitive to parameters variations, the results are illustrated only for one set $[\tilde{k}, \mu, r_{\text{VPP}}] = [20, 0.5, 0.01]$. Stiffness and r_{VPP} are applied as controller designing parameters, but to compare only the results of posture stabilization, μ is selected similar to VPPC-FP.

On the other hand, the performance of the controller depends on the accuracy of the system description provided by matrixes A and B. These are computed for a given fixed point $\mathbf{S}_0^*$. While the system is neutrally stable with respect to the hopping height, after occurrence of the perturbation, the system may converge to another fixed point $\mathbf{S}_1^*$ with

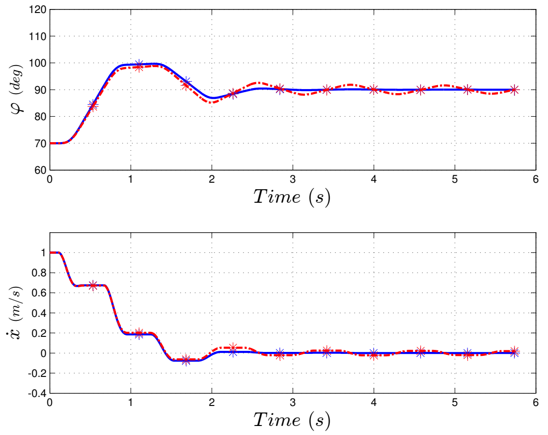

Fig. 5. Responses for VPPC-DB for $[\dot{x}_0, \Delta\varphi_0] = [1 \text{ m/s}, 20°]$ and initial hopping height set to the nominal value. The solid and dashed lines are the responses with and without final fixed point estimation, respectively. Apex events are displayed with the $*$ symbol.

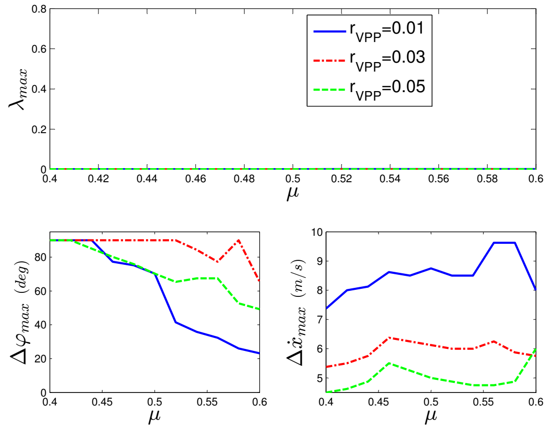

Fig. 6. Stability and robustness for VPPC-DB with $\tilde{k} = 20$. The results are represented for three values of r_{VPP}. For the evaluation of the robustness, estimation of the final fixed point is used systematically.

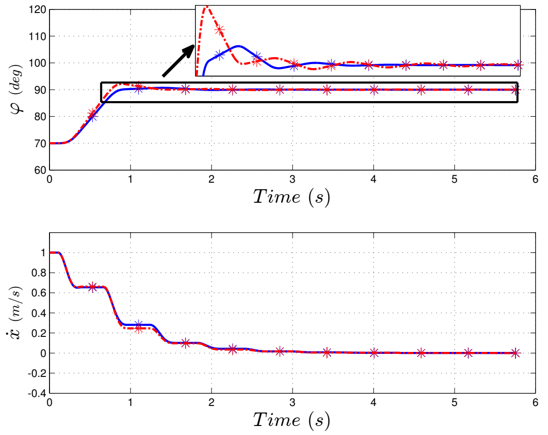

Fig. 7. Responses for VPPC-LQR for $[\dot{x}_0, \Delta\varphi_0] = [1 \text{ m/s}, 20°]$ and initial hopping height set to the nominal value. The solid and dashed lines are the responses with and without final fixed point estimation, respectively. Apex events are displayed with the $*$ symbol.

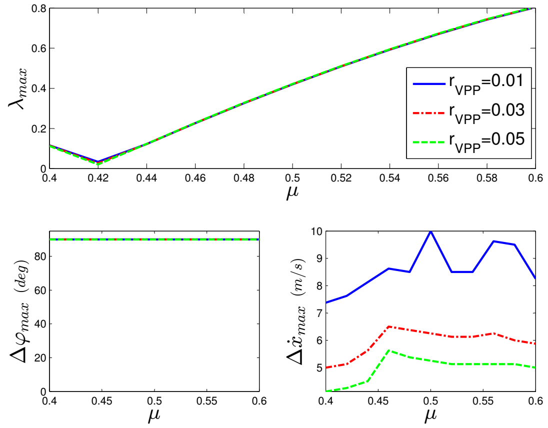

Fig. 8. Stability and robustness for VPPC-LQR with $\tilde{k} = 20$. The results are represented for three values of r_{VPP}. For the evaluation of the robustness, estimation of the final fixed point is used systematically.

a different hopping height. As the controller is designed utilizing the linear map derived for $\mathbf{S}_0^*$ and not $\mathbf{S}_1^*$, its performances will differ from the expected ones. This is especially a concern when large perturbations occur that change the system energy. To solve this problem, the system state after perturbation is used to estimate the final hopping height, by Eq. 13.

$$y_f = y_0 + (\dot{x}_0^2 + J\dot{\varphi}_0^2/m)/2g \qquad (13)$$

where subscript 0 indicates the values of the state variables at the apex, following the perturbation. The Poincaré map is then linearized around the estimated final fixed point $\mathbf{S}_f^* = [y_f, \pi/2, 0, 0]$.

1) Dead-beat control (VPPC-DB): The response of the system after large perturbation of both the initial speed and the trunk angle is represented in Fig. 5. This shows a considerable improvement of the response, especially with respect to the speed of disturbance rejection. With VPPC-DB, it takes less than 3s to vanish the disturbance in comparison with 15 s which was required for VPPC-FP. Using the linearized map around the fixed point with initial hopping height y_0 (dashed line), lightly damped oscillations remains after $t = 3$ s. This illustrates the alteration of the desired controller performance, since the states converge to a fixed point, different from one which is employed to design the controller. When the estimation of the final fixed point (with hopping height y_f) is used, final oscillations are removed and the behavior of the system is close to dead-beat response, as expected (solid line).

In Fig. 6, the results regarding the stability and robustness of the controller for three values of r_{VPP} are displayed. The controller design insures that the maximum eigenvalue is equal to zero. With $\mu = 0.44$ and $r_{\text{VPP}} = 0.01$ m, the controller is generally very robust and has the ability to stabilize the system even with initial perturbations of $\dot{x}_0 = 8$ m/s and $\Delta\varphi_0 = 90°$. Low r_{VPP} values improve the robustness against horizontal velocity perturbations, in contrast with trunk angle perturbations.

2) Discrete LQR (VPPC-LQR): Although fast convergence to steady-state was attained with VPPC-DB, large overshot was observed during the transient phase, especially for the trunk angle (Fig. 5). VPPC-LQR prevents this, while preserves the fast convergence, as shown in Fig. 7. Weight matrix W with diagonal elements $[8, 0.5, 0.5]$ is used, giving higher importance to the trunk angle. The controller performance seems also less affected by the discrepancy between initial (dashed line) and final (solid line) hopping heights. These results could be related to this fact that LQR is a robust controller against the model uncertainties [18]. Finally, the robustness against trunk angle perturbation is greatly improved (Fig. 8).

IV. DISCUSSION

The results show that our approach combining VPPC during stance phase and VBLA during flight phase qualified to make a stable hopping with upright trunk. The original VPPC implementation (VPPC-FP) is able to stabilize hopping in a large parameter region and to provide robustness against moderate perturbations. This achievement is remarkable, while this method does not explicitly control the trunk orientation or even uses information about it.

The original VPPC was extended using event-based control of the VPP position. With this approach, it is possible to achieve stable hopping in the whole investigated parameter region, because stability is guarantied by the controller design. This gives the freedom to select the value of the model parameters, such as the leg stiffness, to reach other objectives, including minimizing energy consumption or satisfying constraints on the maximal torque. Two methods to choose the linear feedback gains were presented. Both of them resulted in considerable improvements regarding the disturbance rejection (fast convergence to the steady state after perturbation) and the robustness against perturbations. In addition, the results were consistent with respect to parameter changes. In the best cases, the controller was able to reject very large perturbations, such as 8 m/s for the horizontal speed and 90° for the trunk angle.

VPPC-DB is the simplest method but it suffers from large overshoots of the trunk angle. This behavior would be problematic in real robots where large movements of the upper body are undesirable (if sensors are located in the torso for example). VPPC-LQR is superior to VPPC-DB regarding the transient behavior and the robustness against trunk angle perturbation. It also offers the possibility to adjust the controller behavior with the weight matrix W, in order to prioritize disturbance rejection for a specific system state. Regarding sensory information, the only cost for all these benefits is to measure or estimate the system state at apex.

Event-based control also requires an accurate linear map of the step-to-step behavior of the model, around a desired hopping motion. If the system settles to another hopping motion, the controller performance will deteriorate. This can happen with this model, for which the hopping height is neutrally stable. To solve this problem, we estimated the final hopping height based on the system state and used this estimation to derive the linear map. However, as perturbations can occur at any step, an update of the linear map may potentially be required at every apex event, with significant computational cost. A simpler solution is to extend the model and add an actuator along the leg axis. It would allow to use existing energy-stabilization schemes to insure convergence towards the desired hopping height.

Together with the issue of energy-management, we are currently working on the extension of our approach to a model including leg mass and damping. Preliminary results are encouraging and indicates that the approach could be implemented on a real robot.

REFERENCES

[1] R. J. Full and D. Koditschek, “Templates and anchors: Neuromechanical hypotheses of legged locomotion on land,” *Journal of Experimental Biology*, vol. 22, pp. 3325–3332, 1999.

[2] R. Blickhan, “The spring-mass model for running and hopping,” *Journal of Biomechanics*, vol. 22, no. 11, pp. 1217–1227, 1989.

[3] T. McMahon and G. Cheng, “The mechanics of running: How does stiffness couple with speed?” *Journal of Biomechanics*, vol. 23, pp. 65–78, 1990.

[4] H. Geyer, A. Seyfarth, and R. Blickhan, “Compliant leg behaviour explains basic dynamics of walking and running,” *Proc. Roy. Soc. B*, vol. 273, no. 1603, pp. 2861–2867, 2006.

[5] M. H. Raibert, *Legged Robots that Balance*. MIT Press, Cambridge MA, 1986.

[6] U. Saranlı, M. Buehler, and D. Koditschek, “Rhex: a simple and highly mobile robot,” *International Journal of Robotic Research*, vol. 20, no. 7, pp. 616–631, 2001.

[7] I. Poulakakis and J. W. Grizzle, “The spring loaded inverted pendulum as the hybrid zero dynamics of an asymmetric hopper,” *IEEE Transaction on Automatic Contro*, vol. 54, no. 8, pp. 1779–1793, 2009.

[8] S.-H. Hyon and T. Emura, “Symmetric walking control: Invariance and global stability,” in *In Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, 2005, pp. 1455–1462.

[9] H. M. Maus, S. Lipfert, M. Gross, J. Rummel, and A. Seyfarth, “Upright human gait did not provide a major mechanical challenge for our ancestors,” *Nature Communications*, vol. 1, no. 6, pp. 1–6, 2010.

[10] H. M. Maus, J. Rummel, and A. Seyfarth, “Stable upright walking and running using a simple pendulum based control scheme,” in *Advances in Mobile Robotics*, 2008.

[11] A. S. A, H. Geyer, M. Guenther, and R. Blickhan, “A movement criterion for running,” *Journal of Biomechanics*, vol. 35, no. 5, pp. 649–655, 2002.

[12] C. Maufroy, H.-M. Maus, and A. Seyfarth, “Simplified control of upright walking by exploring asymmetric gaits induced by leg damping,” in *IEEE Robio 2011*, 2011.

[13] I. Poulakakis and J. Grizzle, “Formal embedding of the spring loaded inverted pendulum in an asymmetric hopper,” in *in Proc. of the European Control Conference*, Kos, Greece, 2007.

[14] C. Farley and D. Morgenroth, “Leg stiffness primarily depends on ankle stiffness during human hopping,” *J Biomech*, vol. 32, pp. 267–273, 1999.

[15] M. H. Raibert, H. B. Brown, E. Hastings, J. Koechling, K. N. Murphy, S. S. Murthy, and A. J. Stentz, “Dynamically stable legged locomotion,” MIT Artificial Intelligence Laboratory, Tech. Rep., 1989.

[16] F. Peuker, C. Maufroy, and A. Seyfarth, “Leg adjustment strategies for stable running in three dimensions,” *Bioinspiration and Biomimetics*, 2011, (accepted).

[17] D. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientic, Belmont, MA., 2000.

[18] K. Zou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Prentice-Hall Inc, 1996.