

Adaptive Control of a Spring-Mass Hopper

İsmail Uyanık, Uluç Saranlı and Ömer Morgül

Abstract—Practical realization of model-based dynamic legged behaviors is substantially more challenging than statically stable behaviors due to their heavy dependence on second-order system dynamics. This problem is further aggravated by the difficulty of accurately measuring or estimating dynamic parameters such as spring and damping constants for associated models and the fact that such parameters are prone to change in time due to heavy use and associated material fatigue. In this paper, we present an on-line, model-based adaptive control method for running with a planar spring-mass hopper based on a once-per-step parameter correction scheme. Our method can be used both as a system identification tool to determine possibly time-varying spring and damping constants of a miscalibrated system, or as an adaptive controller that can eliminate steady-state tracking errors through appropriate adjustments on dynamic system parameters. We present systematic simulation studies to show that our method can successfully accomplish both of these tasks.

I. INTRODUCTION

The utility of legged morphologies and associated dynamic behaviors for robust and efficient locomotion across rough terrain has long been established [15, 31]. Nevertheless, despite the discovery of simple mathematical models [8, 18, 28] and associated analytical solutions [2, 13, 24] that can accurately describe biological runners and support the design of hierarchical controllers for complex legged morphologies [3, 20, 23, 26], physical realization of dynamic legged behaviors has mostly been based on intuition and manual tuning [1, 22, 25, 31] with a few notable exceptions [10, 11]. More recently, however, there has been increasing interest in using model-based analysis and control methods in this context [19, 21], with experimental success for some behaviors [27].

However, even though dynamic models for which we have a sufficiently good analytic understanding can support physically relevant controller designs, the measurement and estimation of particularly the dynamic parameters, such as spring and damping constants for flexible components of a robotic platform, is still a challenging problem. This problem is further aggravated by the possibly time-varying and unpredictable nature of these parameters for autonomous platforms that may remain operational for extended durations of time. Fortunately, this issue is not confined to the control of legged locomotion and received considerable attention from the adaptive control community [5, 17]. Motivated by work in this area, this paper presents a new model-based

adaptive control method for running with the well-known Spring-Loaded Inverted Pendulum (SLIP) model (see Fig. 1), emphasizing on-line estimation of unknown or miscalibrated dynamic system parameters.

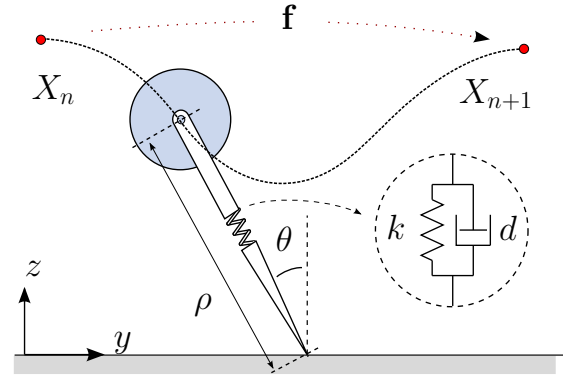


Fig. 1. The Spring-Loaded Inverted Pendulum (SLIP) model. Dashed curve illustrates a single stride from one apex event to the next, defining the return map $X_{n+1} = \mathbf{f}(X_n, \mathbf{u}_n)$.

In contrast to the approach we adopt in this paper, existing research on adaptive control of legged locomotion almost exclusively focuses on how cyclic behaviors of the mechanical locomotor dynamics can be tuned through their coupling with independently running internal clocks (Central Pattern Generators, CPGs) whose dynamics can then be controlled at lower bandwidth [7, 9, 12, 16]. These methods mirror established principles from neurobiology, where groups of neurons in simple organisms were found to remain functional in isolation, producing cyclic control signals even without any high-level control authority [14]. Similar to controller designs based on neural networks and learning [6, 30], such approaches are advantageous in their ability to operate without accurate models, increasing their robustness under unknown environmental conditions such as rough terrain. On the other hand, their structure is often not suitable for incorporating accurate mathematical models when they are in fact available.

Our adaptive control method is based on recently proposed analytic approximations to SLIP dynamics [24], briefly described in Section II. Similar to previous studies, we use a once-per-step deadbeat control strategy that relies on the inversion of an approximate return map for this system. However, unlike previous controllers which assume perfect knowledge of dynamic system parameters (spring and damping constants in particular), and ignore the effects of miscalibrated parameters illustrated in Fig. 2, our adaptive controller described in Section III explicitly considers and

I. Uyanık is with Dept. of Electrical and Electronics Engineering, Bilkent University, 06800 Ankara, Turkey uyanik@ee.bilkent.edu.tr

U. Saranlı is with Dept. of Computer Engineering, Bilkent University, 06800 Ankara, Turkey saranli@cs.bilkent.edu.tr

Ö. Morgül is with Dept. of Electrical and Electronics Engineering, Bilkent University, 06800 Ankara, Turkey morgul@ee.bilkent.edu.tr

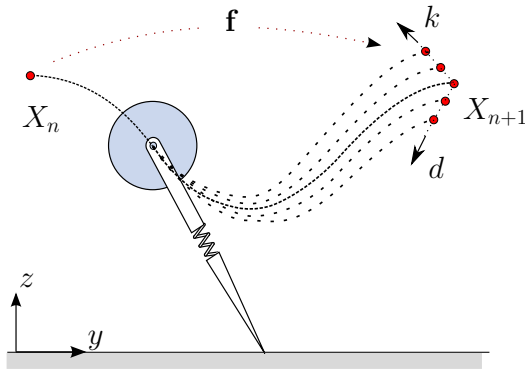


Fig. 2. Impact of miscalibrated dynamic parameters on SLIP trajectory predictions. Arrows indicate directions of change in the apex as a result of increasing k and d . The curve in the middle shows the unperturbed trajectory while the dotted curves show trajectories with different parameter values.

compensates for such errors. We provide simulation results in Section IV, systematically evaluating the performance of our method in the presence of parameter and modeling errors.

II. THE SLIP MODEL

The SLIP model, illustrated in Fig. 1, consists of a point mass attached to a massless leg with a linear spring k and viscous damping d . During running, this model alternates between stance and flight phases with the toe fixed on the ground during the former and the body following a ballistic trajectory during the latter. The highest point in the flight phase is defined as the *apex* point for each stride with the associated state of the system for the n^{th} stride defined as

$$X_n := [z_n, \dot{y}_n]^T. \quad (1)$$

We will also find it convenient to collect relevant dynamic parameters of the system in a single vector as

$$\mathbf{p} := [k, d]^T. \quad (2)$$

Control of SLIP locomotion is achieved through three discrete control inputs at every stride: The leg angle θ_t and leg length ρ_t at touchdown, transitioning from flight to stance and the leg length ρ_l at liftoff, transitioning from stance to flight. Given the control input vector $\mathbf{u} := [\theta_t, \rho_t, \rho_l]$, a Poincaré section at apex with $\dot{z} = 0$ enables us to define a discrete apex return map as

$$X_{n+1} = \mathbf{f}_{\mathbf{p}}(X_n, \mathbf{u}_n). \quad (3)$$

where the dependence of the map on the dynamic parameters \mathbf{p} is explicitly shown. Table I details the notation we use throughout the paper.

Unfortunately, stance dynamics of the SLIP model are not integrable in closed form, making it impossible to find exact analytic expressions for the apex return map [28]. Consequently, in the present paper, we use analytic approximations proposed by Ankarali [24] (AAS approximations), which can successfully incorporate the effects of both damping and gravity. We now define a new, approximate return map as

$$\hat{X}_{n+1} = \hat{\mathbf{f}}_{\mathbf{p}}(X_n, \mathbf{u}_n). \quad (4)$$

TABLE I
NOTATION USED THROUGHOUT THE PAPER

SLIP States and Parameters	
ρ_t, θ_t	Touchdown leg length and angle
ρ_l, θ_l	Liftoff leg length and angle
y, z, \dot{y}, \dot{z}	Body positions and velocities
z_a, \dot{y}_a	Apex height and velocity
Return Maps and Parameters	
\mathbf{f}	Exact plant model
$\hat{\mathbf{f}}$	Analytic approximate solution
k, d	Actual values of spring and damping constants
\hat{k}, \hat{d}	Estimated values of spring and damping constants

Details of how this approximate map is derived can be found in [24]. During running, our adaptive control performs once-per-step corrections to the dynamic parameters in \mathbf{p} , to which both of these return maps depend on. Consequently, we will find it useful to capture the dependence of apex velocity and height coordinates to these parameters through the Jacobian matrices of both of these return maps. For both the exact plant model and the AAS approximations, the associated Jacobians are defined as

$$\mathbf{J} := \partial \mathbf{f} / \partial \mathbf{p} = \begin{bmatrix} \partial \dot{y}_{i+1} / \partial k & \partial \dot{y}_{i+1} / \partial d \\ \partial z_{i+1} / \partial k & \partial z_{i+1} / \partial d \end{bmatrix}, \quad (5)$$

$$\hat{\mathbf{J}}_{\text{AAS}} := \partial \hat{\mathbf{f}} / \partial \mathbf{p} = \begin{bmatrix} \partial \hat{\dot{y}}_{i+1} / \partial k & \partial \hat{\dot{y}}_{i+1} / \partial d \\ \partial \hat{z}_{i+1} / \partial k & \partial \hat{z}_{i+1} / \partial d \end{bmatrix}. \quad (6)$$

While we derive $\hat{\mathbf{J}}_{\text{AAS}}$ analytically by simple differentiation of the approximations given in [24], we will use numerical differentiation to compute the plant Jacobian \mathbf{J} in the comparative experimental results of Section IV.

III. ADAPTIVE CONTROL OF SLIP RUNNING

In the presence of a sufficiently accurate model, gait control of the SLIP model can be achieved through a deadbeat strategy as described in [24]. Given a desired apex state X^* , inversion of the apex return map for the z and \dot{y} components of the state yields the controller

$$\mathbf{u} = \hat{\mathbf{f}}_{\mathbf{p}}^{-1}(X^*, X_n). \quad (7)$$

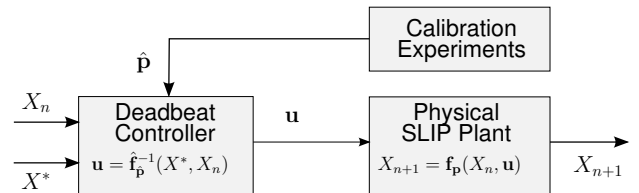


Fig. 3. Deadbeat SLIP gait control through the inversion of the approximate plant model.

Note, however, that the approximate return map and hence its inversion can only rely on possibly inaccurate parameter estimates $\hat{\mathbf{p}}$ for spring and damping constants. As shown in the block diagram of Fig. 3, these estimates are often obtained through calibration experiments on the platform but may not provide sufficiently good accuracy.

The core of our adaptive control strategy relies on once-per-step corrections to these estimates based on the difference

between predicted and measured apex states for each apex. This corrective parameter adjustment is very similar to how estimation methods such as Kalman filters use innovation on sensory measurements to perform state updates [29].

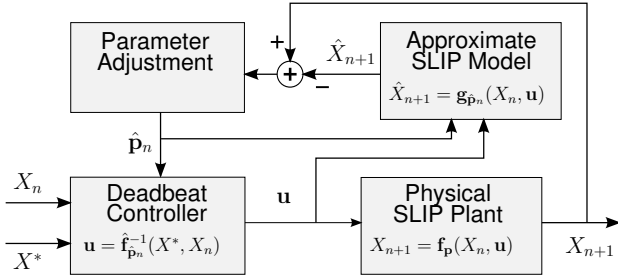


Fig. 4. The proposed adaptive control strategy. Prediction errors of an approximate plant model \mathbf{g} (computed either using exact plant simulations \mathbf{f} or AAS approximations $\hat{\mathbf{f}}$) are used to dynamically adjust parameter estimates $\hat{\mathbf{p}}_n$.

Fig. 4 illustrates the block diagram for the adaptive parameter correction scheme we propose in this paper. Our method relies on the availability of an approximate return map \mathbf{g} that can predict the apex state outcome of a single step, given the apex states of the previous step X_n and associated control inputs \mathbf{u}_n . In this paper, we consider two alternatives for this approximate predictor model:

- 1) *Exact SLIP Model (ESM)*: This alternative uses $\mathbf{g} = \mathbf{f}$, computed through numerical simulation of SLIP dynamics. The associated Jacobian \mathbf{J} is also computed numerically.
- 2) *Approximate Analytic Solution (AAS)*: This option uses $\mathbf{g} = \hat{\mathbf{f}}$, adopting the approximate analytic solutions of [24] as a predictor for SLIP trajectories. The associated Jacobian $\hat{\mathbf{J}}_{AAS}$ is analytically derived through straightforward differentiation.

As we will show in Section IV, the first option is useful for accurate identification of the dynamic parameters of the system, whereas the second option will be useful in eliminating steady-state tracking errors for the gait-level control of SLIP running and is much more suitable for real-time implementation on a physical platform.

Regardless of which predictor is chosen, an apex state prediction error is computed at every step as

$$\mathbf{e} := X_{n+1} - \hat{X}_{n+1} = \mathbf{f}_p(X_n, \mathbf{u}) - \mathbf{g}_{\hat{\mathbf{p}}_n}(X_n, \mathbf{u}). \quad (8)$$

Note that the computation of this error requires measurement of actual apex states X_{n+1} at every stride, which can be accomplished through proper instrumentation and state estimation techniques. More importantly, however, the predictor is expected to use the *updated estimates* of dynamic system parameters $\hat{\mathbf{p}}_n$ rather than their unknown physical values experienced by the SLIP plant, making it relevant in computing corrections on these parameters.

The goal of our adaptive control approach is to bring the steady-state value of this prediction error to zero. In other words, we seek to have

$$\lim_{n \rightarrow \infty} (\hat{X}_n - X_n) = 0, \quad (9)$$

which will also indirectly yield steady-state parameter estimates as

$$\hat{\mathbf{p}} = \lim_{n \rightarrow \infty} (\hat{\mathbf{p}}_n). \quad (10)$$

We accomplish both of these goals using a conceptually simple yet effective parameter adjustment strategy based on the Jacobians defined in (5) and (6). By definition, these Jacobian matrices relate infinitesimal changes in the apex state predictions to infinitesimal changes in the dynamic system parameters with

$$\delta \hat{X}_{n+1} = (\partial \mathbf{g} / \partial \mathbf{p})|_{X_n} \delta \mathbf{p}. \quad (11)$$

Based on this relation and the prediction errors computed at every stride, we propose the parameter update strategy

$$\hat{\mathbf{p}}_{n+1} = \hat{\mathbf{p}}_n + K_e (\partial \mathbf{g} / \partial \mathbf{p})^{-1}|_{X_n} \mathbf{e} \quad (12)$$

where $K_e < 1$ is a gain coefficient that can be used to tune convergence and prevent oscillatory behavior. This yields an on-line adaptation mechanism that can be used for both predictor choices, with the *ESM* choice resulting in accurate system identification and the *AAS* choice yielding adaptive gait control as we will show in Section IV.

It is important to note that practical applicability of our adaptive control method inevitably depends on the accuracy of the underlying SLIP model. Even though the linear spring model we use in this paper was previously shown to result in reasonable predictive accuracy for biological runners [28], extensions to the model and the associated analytical approximations may be needed for systems with more complex, nonlinear springs.

IV. PERFORMANCE ANALYSIS

A. Simulation Environment and Performance Criteria

The two related but different goals of our adaptive control method are the estimation of unknown or miscalibrated dynamic system parameters and accurate tracking of desired apex states. Both of these goals can be defined as a function of the steady-state behavior of the system. Consequently, we define three different percentage error measures

$$SSE_k := 100 \lim_{n \rightarrow \infty} (|\hat{k}_n - k|/k), \quad (13)$$

$$SSE_d := 100 \lim_{n \rightarrow \infty} (|\hat{d}_n - d|/d), \quad (14)$$

$$SSE_a := 100 \lim_{n \rightarrow \infty} (||X_n - X^*||/||X^*||), \quad (15)$$

with SSE_k and SSE_d capturing system identification performance and SSE_a characterizing the tracking performance of the adaptive controller.

TABLE II
SIMULATION APEX GOAL AND PARAMETER RANGES

z_a^* (m)	y_a^* (m/s)	k (N/m)	d (Ns/m)	m (kg)
[1.25, 1.75]	[1.25, 2.75]	[800, 2000]	[3, 15]	1

In order to characterize the performance of our adaptive control strategy, we ran a large number of simulations using different apex goal settings X^* as well as different choices

of dynamic parameters \mathbf{p} within ranges specified in Table II, chosen to be consistent with biomechanics literature [4] as well as existing legged robots [25] to increase the relevance of our results. The hybrid SLIP plant dynamics in Figures 3 and 4 were simulated in Matlab using a fourth-order, adaptive time-step Runge-Kutta integrator with exact detection of touchdown and liftoff events. Simulations were run until steady-state was reached with a tolerance of 10^{-4} in the norm of the apex state. Steady-state trajectories were found to be independent of initial apex states. However, since the convergence behavior of (12) depends on the choice of the predictor and the update gain, we will consider different initial parameter estimates \mathbf{p}_0 for our simulations.

B. Accurate Control with the AAS Predictor

In this section, we present apex goal tracking simulations with the AAS predictor introduced in Section III. Before we proceed with more systematic performance results, however, Fig. 5 illustrates an example SLIP simulation started with a non-adaptive controller in the presence of 20% errors for the estimates of both the spring and damping constants, with subsequent activation of our adaptive controller using the AAS predictor around $t = 2s$, finally followed by a step change in the apex goal around $t = 4.55s$.

As expected, using the non-adaptive controller with miscalibrated dynamic parameters results in a substantial steady-state error due to prediction errors in the analytic approximations of [24]. When the adaptive controller is switched on around $t = 2s$, this error is quickly eliminated and estimated values of both the spring and damping constants quickly converge towards their physical values as shown in Fig. 5. The last five steps of the simulation shows that steady-state tracking remains accurate even when a step input with a large magnitude is given to the system.

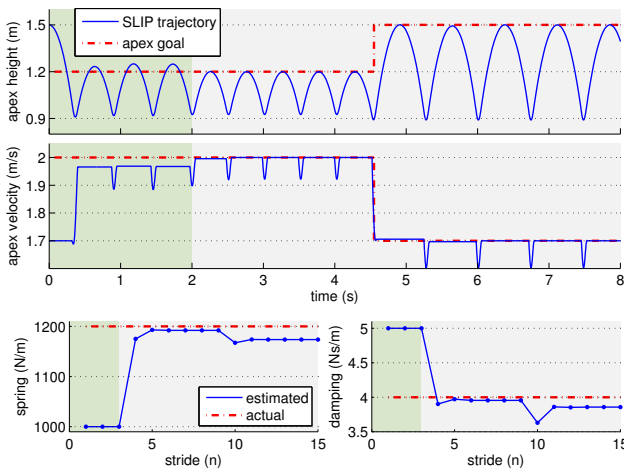


Fig. 5. An example SLIP simulation started with a non-adaptive controller (dark shaded region) and 20% error in both the spring and damping constants. Our adaptive controller with the AAS predictor was started around $t = 2s$ and a step change in the apex goal was given around $t = 4.55s$.

More generally, Fig. 6 illustrates the average tracking performance of our adaptive controller across the range of

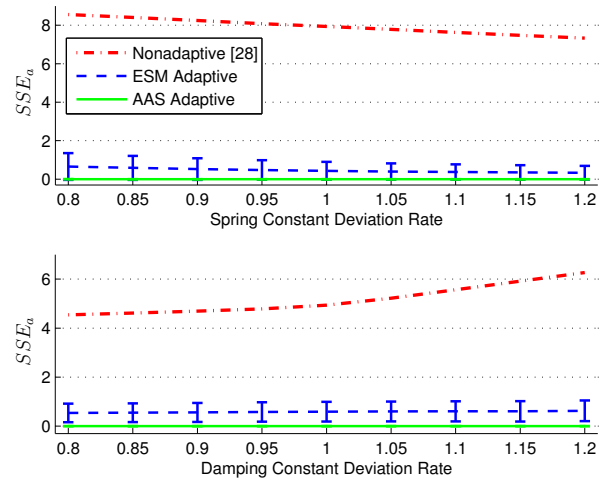


Fig. 6. Steady-state apex goal tracking errors for the non-adaptive, AAS adaptive and ESM adaptive controllers. Error measures were averaged across 321489 simulation runs with different goals and initial parameter estimates. Vertical bars show standard deviations and were omitted for the non-adaptive case since they were very large.

apex goals and parameter choices given in Table II corresponding to 321489 simulation runs. The top and bottom plots respectively show the dependence of average errors and their standard deviations on the initial deviations of the spring and damping constants for a non-adaptive controller as well as our adaptive controller with both the AAS and *ESM* predictors. As expected, the non-adaptive controller results in large tracking errors (with very high standard deviations, omitted from the figure for clarity) whereas the AAS Adaptive controller reduces the steady state error to zero. Average apex tracking and parameter estimation errors and their standard deviations across all simulations are also given in Table III.

It may be surprising that the AAS predictor outperforms the *ESM* predictor based on the exact SLIP model for apex goal tracking. However, note that the deadbeat controller of (7) is based on the inversion of the AAS analytic approximations. Naturally, when dynamic system parameters are adapted such that the predictions of these approximations are error-free, the resulting controller achieves zero tracking error in steady-state. In contrast, while the *ESM* predictor can accurately estimate the dynamic parameters as shown in Section IV-C, some remaining prediction errors still remain, leading to the small steady-state tracking errors of Fig. 6.

TABLE III

PERCENTAGE APEX TRACKING AND PARAMETER ESTIMATION ERRORS

Error Measure:	SSE_a	SSE_k	SSE_d
Non-adaptive	6.56 ± 4.64	10 ± 6.20	10 ± 6.20
AAS Adaptive	0.002 ± 0.001	2.34 ± 1.45	5.53 ± 2.81
ESM Adaptive	0.52 ± 0.45	0.0008 ± 0.0005	0.007 ± 0.005

Finally, Fig. 7 shows a comparison of the dynamic tracking performance for the non-adaptive controller and our adaptive controller with the AAS predictor. Once again, our controller quickly converges to the desired trajectory, outperforming the non-adaptive controller which suffers from miscalibrated parameter estimates. These results show that the proposed

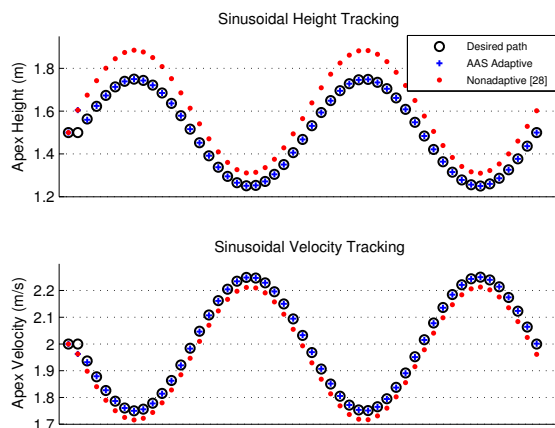


Fig. 7. Apex height (top) and speed (bottom) tracking performance for a sinusoidal reference trajectory. Each data point corresponds to a single apex event.

controller can maintain accurate tracking even for dynamic goal settings and not just for a single static target.

C. System Identification with the *ESM* Predictor

In this section, we present the system identification performance of our algorithm with the *ESM* predictor. Fig. 8 shows an example SLIP simulation similar to the example of the previous section, but with the *ESM* predictor instead. Once again, the first three steps were controlled with the non-adaptive strategy, activating the adaptive controller at $t = 2s$ and finally initiating a step change in the apex goal setting at $t = 4.55s$.

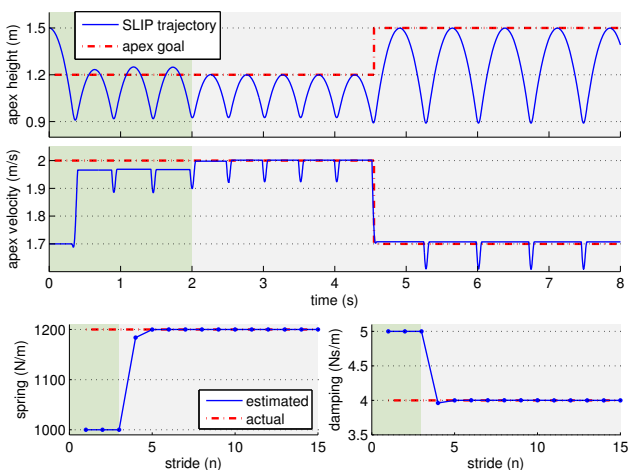


Fig. 8. An example SLIP simulation started with a non-adaptive controller (dark shaded region) and 20% error in both the spring and damping constants. Our adaptive controller with the *ESM* predictor was started around $t = 2s$ and a step change in the apex goal was given around $t = 4.55s$.

In contrast to the *AAS* predictor, the use of the *ESM* predictor allows better estimation of unknown dynamic parameters at the expense of steady-state tracking accuracy. This can be observed in the bottom two plots of Fig. 8

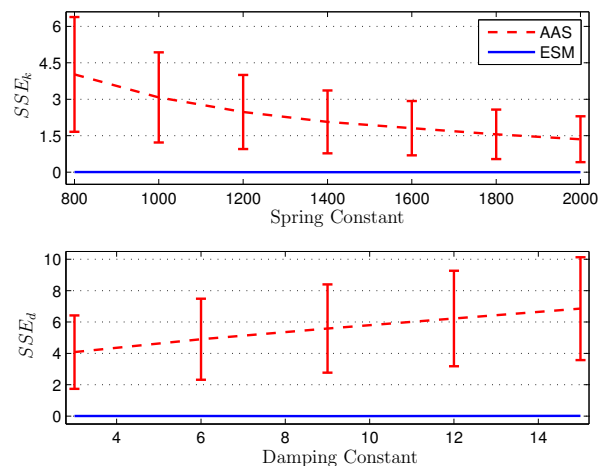


Fig. 9. Errors in steady-state estimations for the spring (top) and damping (bottom) constants using the *AAS* adaptive and *ESM* adaptive controllers. Error measures were averaged across 321489 simulation runs with different goals and initial parameter estimates. Vertical bars show standard deviations.

as well as the corresponding columns of Table III showing an increase in the average apex tracking error. This is not surprising since the elimination of prediction errors for the exact SLIP predictor corresponds to exact identification of the unknown dynamic parameters. For a physical robot, this would be the best way to estimate the spring and damping constants as accurately as possible.

Following this isolated example, Fig. 9 shows the parameter estimation performance of our adaptive method both with the *ESM* and *AAS* predictors across a larger range of apex goal and parameter settings. Since the non-adaptive controller does not update parameter estimates in any way, we have not included it in the error figures. Our results for both the spring and damping constants show that while the *ESM* predictor perfectly estimates system parameters, the *AAS* predictor, which is much more practical and computationally feasible for on-line application due to its analytic nature, also performs very well and yields steady-state parameter estimation errors well below the 10-15% that would be expected from manual calibration alone.

In summary, our adaptive controller can be used both as a system identification tool through the use of the *ESM* predictor, or as an accurate gait controller for apex states with the *AAS* predictor. The latter option is much more suitable for on-line operation on a physical running robot since the approximate solutions of [24] and associated Jacobians are all formulated analytically, making them computationally feasible. The *ESM* predictor, however, not only requires simulated trajectory predictions, but also incorporates numeric differentiation around these simulated trajectories, making it much more suitable for offline system identification.

Nevertheless, in all cases, our adaptive methods perform much better than the non-adaptive approach both for gait control and system identification. Our contributions in this paper clearly illustrate that when analytic solutions to the dynamics of a legged platform are available, their structure and efficiency can be exploited to yield effective solutions

both for control and system identification.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel adaptive control algorithm to both support on-line identification of unknown dynamic system parameters and improve steady-state tracking performance of previously proposed control algorithms for the well known SLIP model. Our method used as a system identification tool addresses the practical difficulty of measuring possibly time-varying dynamic system parameters such as spring and damping constants and associated degradation in controller performance when they cannot be correctly estimated. In contrast, our method used as an adaptive controller allows effective elimination of steady-state tracking errors under different types of modeling errors for inverse dynamics controllers.

The choice between these two different modes of operation depends on the choice of a predictor model against which state measurements are compared at each step. We show through systematic simulations that a predictor based on numerical integration of SLIP dynamics is capable of accurate system identification, whereas a predictor based on the analytic approximations proposed in [24] allows elimination of steady-state tracking errors for a deadbeat controller based on the same approximations. Extensive simulation results for both predictors show that they successfully realize these objectives and substantially improve on control performance relative to existing non-adaptive controllers.

Our longer term goal is to design legged platforms that can reactively negotiate rough terrain. The applicability of mathematical models that are relevant for this purpose critically depends on our ability to accurately estimate associated parameters to be used by model-based planners. Consequently, starting from a direct implementation of the method we propose in this paper on a monopodal platform, our future work includes extensions to more complex legged models and locomotion controllers. In this context, we believe that our work shows some of the benefits offered by analytic solutions to mathematical models of locomotory behaviors.

VI. ACKNOWLEDGMENTS

We thank M. Mert Ankaralı and Melih Çakmakçı for their ideas and support. This work was supported by National Scientific and Technological Council of Turkey (TUBITAK), through project 109E032 and İsmail Uyanık's scholarship.

REFERENCES

- [1] M. Ahmadi and M. Buehler. Controlled passive dynamic running experiments with the ARL-monopod II. *IEEE Trans. on Robotics*, 22(5):974–986, Oct. 2006.
- [2] M. Ankaralı, O. Arslan, and U. Saranlı. An analytical solution to the stance dynamics of passive spring-loaded inverted pendulum with damping. In *Mobile Robotics: Solutions and Challenges*, pages 693–700. World Scientific, Istanbul, Turkey, October 2009.
- [3] M. Ankaralı and U. Saranlı. Control of underactuated planar hexapedal pronking through a dynamically embedded SLIP monopod. In *Proc. of the Int. Conf on Robotics and Automation*, pages 4721–4727, Anchorage, Alaska, May 3-8 2010.
- [4] A. Arampatzis, G.-P. Brüggemann, and V. Metzler. The effect of speed on leg stiffness and joint kinematics in human running. *Journal of Biomechanics*, 32:1349–1353, 1999.
- [5] K. J. Astrom and B. Wittenmark. *Adaptive Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1994.
- [6] R. D. Beer and J. C. Gallagher. Evolving Dynamical Neural Networks for Adaptive Behavior. *Adaptive Behavior*, 1(1):91–122, 1992.
- [7] M. Berkemeier and K. Desai. Control of hopping height in legged robots using a neural-mechanical approach. In *Proc. of the Int. Conf. on Robotics and Automation*, pages 1695–1701 vol.3, 1999.
- [8] R. Blickhan. The spring-mass model for running and hopping. *Journal of Biomechanics*, 22:1217–1227, 1989.
- [9] G. Brambilla, J. Buchli, and A. Ijspeert. Adaptive four legged locomotion control based on nonlinear dynamical systems. In *Proc. of the Int. Conf. on the Simulation of Adaptive Behavior*, volume 4095 of *Lecture Notes in Computer Science*. Springer Verlag, 2006.
- [10] C. Chevallereau, E. R. Westervelt, and J. W. Grizzle. Asymptotically Stable Running for a Five-Link, Four-Actuator, Planar Bipedal Robot. *The International Journal of Robotics Research*, 24(6):431–464, 2005.
- [11] S. Collins, A. Ruina, R. Tedrake, and M. Wisse. Efficient Bipedal Robots Based on Passive-Dynamic Walkers. *Science*, 307(5712):1082–1085, 2005.
- [12] Y. Fukuoka, H. Kimura, and A. H. Cohen. Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts. *The Int. Journal of Robotics Research*, 22(3-4):187–202, 2003.
- [13] H. Geyer, A. Seyfarth, and R. Blickhan. Spring-mass running: simple approximate solution and application to gait stability. *Journal of Theoretical Biology*, 232:315–328, February 2005.
- [14] S. Grillner. Neurobiological bases of rhythmic motor acts in vertebrates. *Science*, 228(4696):143–149, 1985.
- [15] P. Holmes, R. Full, D. Koditschek, and J. Guckenheimer. The dynamics of legged locomotion: Models, analyses, and challenges. *SIAM Review*, 48(2):207–304, 2006.
- [16] W. Ilg, J. Albiez, H. Jedicke, K. Berns, and R. Dillmann. Adaptive periodic movement control for the four legged walking machine bisam. In *Proc. of the Int. Conf. on Robotics and Automation*, volume 3, pages 2354–2359 vol.3, 1999.
- [17] I. D. Landau, R. Lozano, and M. M'Saad. *Adaptive Control*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998.
- [18] T. McGeer. Passive dynamic walking. *International Journal of Robotics Research*, 9(2):62–82, 1990.
- [19] I. Poulakakis and J. Grizzle. The spring loaded inverted pendulum as the hybrid zero dynamics of an asymmetric hopper. *IEEE Trans. on Automatic Control*, 54(8):1779–1793, aug. 2009.
- [20] I. Poulakakis and J. W. Grizzle. Formal embedding of the spring loaded inverted pendulum in an asymmetric hopper. In *Proc. of the European Control Conference*, 2007.
- [21] I. Poulakakis, E. Papadopoulos, and M. Buehler. On the Stability of the Passive Dynamics of Quadrupedal Running with a Bounding Gait. *The International Journal of Robotics Research*, 25(7):669–687, 2006.
- [22] M. H. Raibert. *Legged robots that balance*. MIT Press, Cambridge, MA, USA, 1986.
- [23] U. Saranlı. *Dynamic Locomotion with a Hexapod Robot*. PhD thesis, The University of Michigan, Ann Arbor, MI, September 2002.
- [24] U. Saranlı, O. Arslan, M. Ankaralı, and O. Morgül. Approximate analytic solutions to non-symmetric stance trajectories of the passive spring-loaded inverted pendulum with damping. *Nonlinear Dynamics*, 2010.
- [25] U. Saranlı, M. Buehler, and D. E. Koditschek. RHEx: A simple and highly mobile robot. *The Int. Journal of Robotics Research*, 20(7):616–631, July 2001.
- [26] U. Saranlı and D. E. Koditschek. Template based control of hexapedal running. In *Proc. of the Int. Conf. On Robotics and Automation*, volume 1, pages 1374–1379, Taipei, Taiwan, September 2003.
- [27] U. Saranlı, A. A. Rizzi, and D. E. Koditschek. Model-based dynamic self-righting maneuvers for a hexapedal robot. *International Journal of Robotics Research*, 23(9):903–918, September 2004.
- [28] W. J. Schwind. *Spring loaded inverted pendulum running: a plant model*. PhD thesis, University of Michigan, Ann Arbor, MI, USA, 1998.
- [29] D. Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley, June 2006.
- [30] R. L. Tedrake. *Applied Optimal Control for Dynamically Stable Legged Locomotion*. Ph.d., Massachusetts Institute of Technology, Boston, MA, September 2004.
- [31] D. Wooden, M. Malchano, K. Blankespoor, A. Howardy, A. A. Rizzi, and M. Raibert. Autonomous navigation for BigDog. In *Proc. of the Int. Conf. on Robotics and Automation*, May 3-8 2010.