Coordination of Monopedal SLIP Models Towards Quadrupedal Running

Conference Paper · October 2016

DOI: 10.1109/IROS.2016.7759800

2 authors:

Mohammad Shahbazi
Istituto Italiano di Tecnologia
15 PUBLICATIONS **123** CITATIONS
SEE PROFILE

Gabriel a. d. Lopes
Delft University of Technology
59 PUBLICATIONS **1,277** CITATIONS
SEE PROFILE

Coordination of Monopedal SLIP Models Towards Quadrupedal Running

Mohammad Shahbazi, Gabriel A. D. Lopes

***Abstract*— This paper presents a coordination controller for the Dual-SLIP model, a novel template for quadrupedal steady and transitional running. The model consists of a pair of "physically-unconnected" Spring-Loaded Inverted Pendulums (SLIPs), each representing a part of the body of a quadruped (see Figure 1). For this model, we propose a spatio-temporal coordination controller that describes the evolution of coordination parameters by simple difference equations. A "time-aware" deadbeat low-level controller is also proposed to realizing the generated control specifications in each SLIP individually. Evaluation of the proposed coordination controller for the Dual-SLIP model in simulation shows that even with remarkably off-phase initial conditions and ground height variation disturbances, quadrupedal bounding, pronking and different transitions between them can be realized.**

I. Introduction

The study of legged locomotion using *template models* [1] provides insights to the dominant features of the dynamics while abstracting out less important details [2]. The resulting platform-independent reductive models typically admit analytical representations that are more tractable than the original higher-order robot models. On the other hand, oversimplified representations are not preferred as their *anchoring* [1] on the real robot would be overcomplicated. Hence, the choice of a sufficiently descriptive template for a given robot is of particular importance.

The monopedal Spring-Loaded Inverted Pendulum (SLIP) has long been established as the standard template for the running behavior of humans and animals of different morphologies [3], [4]. However, the SLIP represents a body with a point mass, thereby cannot be solely useful in capturing the pitch dynamics of the body, as discussed in [5]. Having realized this, a number of template models have been proposed exclusively representing quadrupedal running systems, most of which assume rigid torsos [6]–[11], while only a few recent ones feature flexible torsos [12], [13].

Cao and Poulakakis [12] focused on investigating the consequences of torso flexibility on quadrupedal template running, with an emphasis on passively generated bounding gaits. Similarly, Pouya et al. [13] analyzed different types of actuation in the spine of a segmented torso in simulation to evaluate the bounding gaits characteristics such as periodicity and stability. In both studies, a successful gait generation depends on the identification of proper limit cycles for which extensive numerical search is often needed.

The authors are with Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands. M. Shahbazi is also with Department of Advanced Robotics, Istituto Italiano di Tecnologia, Genoa, Italy. Email: {m.shahbaziaghbelagh; g.a.delgadolopes}@tudelft.nl.

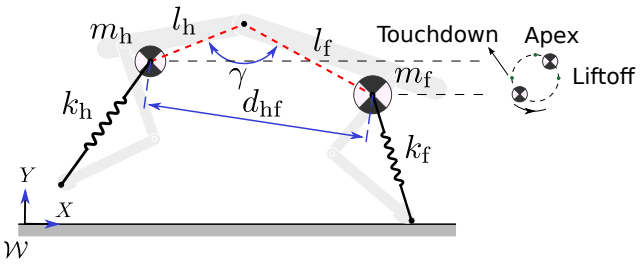

Fig. 1: Dual-SLIP template model for quadrupedal running.

In this paper we approach the problem of generating quadrupedal running in a template setting from a fresh point of view. The template model we propose, referred to as the Dual-SLIP model, is composed of a pair of physically-unconnected SLIPs each representing a part of the body of a quadruped (see Figure 1). We show that if the fore and hind SLIPs are coordinated properly, the Dual-SLIP template can represent different steady and transitional quadrupedal running behaviors. To the best of authors' knowledge, such a "spatio-temporal" coordination framework for the SLIP model remained unexplored in the related literature. Compared to the aforementioned studies, the significance of the proposed approach is that it does not need an attractor limit cycle of the whole system to be identified a priori, and that it can benefit from the mature literature of SLIP for the low-level control structures.

Currently, Central Pattern Generators (CPGs) are standard tools in generating references whose realization leads to coordination of legs in robotics (see [14] for a survey on CPGs). Although used widespread, the application of the CPGs-based methods presents its own challenges, mainly due to the nature of their foundation as sets of coupled differential equations. Alternatively, we explore a reference generation method in which, similar to our previous works [15], [16], the evolutions of the coordination parameters are described using difference equations. This makes the interpretation of the resulting behaviors easier, particularly for transient phases.

The success of the coordination framework proposed relies also on the presence of a low-level controller that is capable of realizing spatio-temporal references. In this regard, we also contribute to the existing control methods of SLIP by proposing a *time-aware* deadbeat control scheme, explicitly accounting for the control of certain event timings.

The organization of the paper is as follows. In Section II we describe the structure of the Dual-SLIP model, and formulate the gait specifications intended to be realized in this study. Section III presents the structure of the developed spatio-temporal coordination framework by describing the

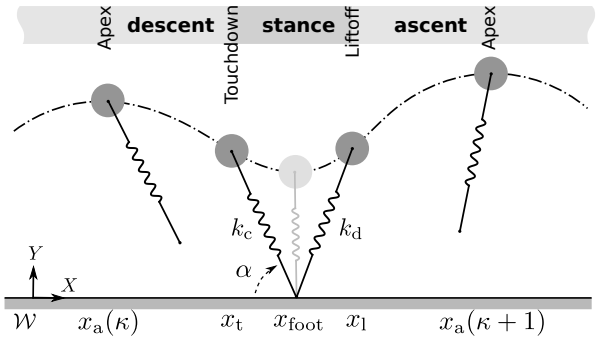

Fig. 2: Monopedal SLIP model and its phases of motion.

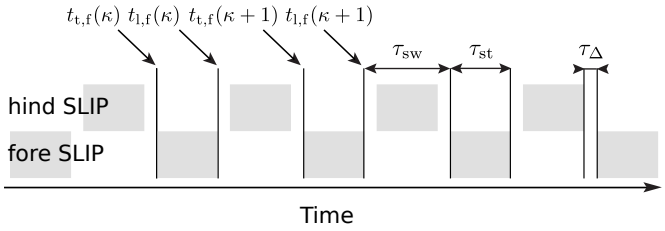

Fig. 3: Event schedule for a bounding gait of the Dual-SLIP model. Gray rectangles represent stance and white space represents swing.

Function 1 Bounding gait temporal state scheduling

function BOUNDING($s(\kappa-1)$)
  $t_{t,f}(\kappa) \leftarrow \max(t_{l,f}(\kappa-1)+\tau_{sw}, t_{l,h}(\kappa-1)+\tau_\Delta)$
  $t_{l,f}(\kappa) \leftarrow t_{t,f}(\kappa)+\tau_{st}$
  $t_{t,h}(\kappa) \leftarrow \max(t_{l,h}(\kappa-1)+\tau_{sw}, t_{l,f}(\kappa)+\tau_\Delta)$
  $t_{l,h}(\kappa) \leftarrow t_{t,h}(\kappa)+\tau_{st}$
  return $s(\kappa)$
end function

underlying mechanisms and their functions. Next, in Section IV we employ the developed controller to systematically generate quadrupedal bounding, pronking and their transitions, whose robustness against remarkably off-phase initial conditions and ground height variations are verified in simulation. Finally, we conclude the paper in Section V.

II. Dual-SLIP Template Model

A. Model description

As shown in Figure 1, the proposed Dual-SLIP model consists of a fore and hind SLIP models representing the fore and hind bodies of a quadruped. We assume the mass of fore (hind) legs and the respective part of torso are lumped in m_f (m_h). As depicted in the dashed red lines, a virtual articulated massless mechanism is also considered to loosely simulate a torso. This *virtual* torso has no effect on the system dynamics and is meant only for evaluating the feasibility of the model. Nevertheless, the effect of actual torso is implicitly captured in the states of m_f and m_h.

The considered SLIP models are of the standard form frequently used in the relevant literature (see, for example, [17]). As illustrated in Figure 2, each SLIP represents a body with a point mass connected to a compliant massless leg, moving in the sagittal plane with gravitational acceleration g. The leg is endowed with a linear stiffness, and its rest length is denoted by l_{rest}.

A common cycle of motion for both SLIPs is a sequence of swing and stance phases separated by *touchdown* and *liftoff* events. The stance phase can further be divided to spring compression and decompression portions with associated stiffnesses k_c and k_d, respectively. Also note that a cycle initiates from the *apex* where $\dot{y}=0$, and the leg touches the ground with the touchdown angle α, as illustrated in Figure 2.

The continuous state vector for each SLIP in the inertial frame $\mathcal{W}$ is represented by $r := \begin{bmatrix} x & \dot{x} & y & \dot{y} \end{bmatrix}^T$, with $r \in \mathcal{R} \subset \mathbb{R}^4$. A discrete state vector is also considered as $q := \begin{bmatrix} x_{foot} & y_{foot} & \mathcal{M} & \mathcal{Q}_j & \mathcal{Q}_e \end{bmatrix}^T$, where $\mathcal{M} \in \{descent, compr., decompr., ascent\}$, and $\mathcal{Q}_j$ and $\mathcal{Q}_e$ are ordered lists containing events information described in detail in Section III-B.

The well-know Poincaré section of each SLIP can be taken at associated apex states. The resulting Poincaré state vector is then defined as $z := \begin{bmatrix} x_a & \dot{x}_a & y_a \end{bmatrix}^T$, in which $\dot{y}_a$ is omitted by definition. Now one can define a discrete abstraction of the system dynamics (i.e., the Poincaré map $\mathcal{P}(\cdot)$) as a mapping between two consecutive apex states:

$$z(\kappa+1) = \mathcal{P}\left(z(\kappa), u(\kappa)\right), \tag{1}$$

where $u(\kappa) = \begin{bmatrix} \alpha & k_c & k_d \end{bmatrix}^T$ are the control parameters, whose calculation procedure is detailed in Section III-A.

B. Gait definition

As briefly outlined earlier, we are interested in developing a leg coordination framework that is event-driven, i.e., governed by difference equations rather than differential equations as in CPGs-based methods. As we have shown in [15] for different gaits of a hexapod, the temporal synchronization can be realized using the *maximization* operator in the equations describing the evolution of the gait parameters. Following a similar approach, here we derive such equations for the bounding and pronking gaits of the Dual-SLIP model. Let s be the *temporal state scheduling vector* (hereon, scheduling vector or schedule), defined as:

$$s(\kappa) = \begin{bmatrix} t_{t,f}(\kappa) & t_{t,h}(\kappa) & t_{l,f}(\kappa) & t_{l,h}(\kappa) \end{bmatrix}^T, \tag{2}$$

where $\kappa \in \mathbb{N}$ is the cycle index, and $t_{t,i}(\kappa)$ and $t_{l,i}(\kappa)$, $i \in \{f, h\}$, are the touchdown and liftoff time instants in the κth cycle, respectively. Additionally, we denote by τ_{st}, τ_{sw} and τ_Δ the stance, swing and double-swing durations, which at this point are assumed to be constant during time, for the sake of simplicity.

A symmetric bounding gait for the Dual-SLIP model is illustrated in Figure 3. The underlying equations that generate this temporal schedule are represented in Function 1. Following a similar procedure, the corresponding equations for the pronking gait are formulated in Function 2.

Function 2 Pronking gait temporal state scheduling

function PRONKING($s(\kappa-1)$)
  $t_{\mathrm{t,f}}(\kappa) \leftarrow \max(t_{\mathrm{l,f}}(\kappa-1), t_{\mathrm{l,h}}(\kappa-1)) + \tau_{\mathrm{sw}}$
  $t_{\mathrm{t,h}}(\kappa) \leftarrow \max(t_{\mathrm{l,h}}(\kappa-1), t_{\mathrm{l,f}}(\kappa)) + \tau_{\mathrm{sw}}$
  $t_{\mathrm{l,f}}(\kappa) \leftarrow t_{\mathrm{t,f}}(\kappa) + \tau_{\mathrm{st}}$
  $t_{\mathrm{l,h}}(\kappa) \leftarrow t_{\mathrm{t,h}}(\kappa) + \tau_{\mathrm{st}}$
  return $s(\kappa)$
end function

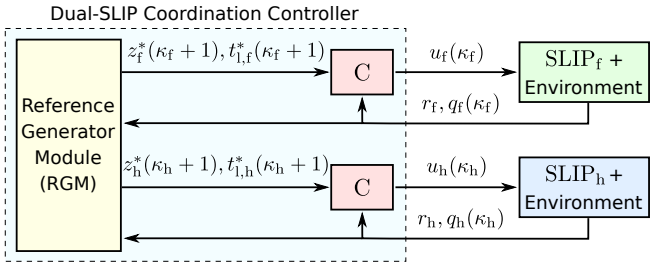

Fig. 4: Overall structure of the proposed coordination controller.

By definition, every event schedule produced by Functions 1 and 2 ensures constant swing durations and adjusts the stance durations for synchronization. It is important to realize that this touchdown constraint is not unique or strictly needed to synchronize legs, as the opposite condition would also be valid.

III. Coordination Controller

This section presents the control strategy proposed to induce spatio-temporal coordination between the SLIPs in the Dual-SLIP model. Theoretically, the method can be applied to compositional treatments of cyclic systems of any types, in which the spatial and temporal states can (partly) be controlled at least once per cycle. However, with respect to the scope of the present work, we illustrate the method exclusively for the Dual-SLIP model.

An overview of the coordination controller is depicted in Figure 4. The success of the framework relies on the presence of an effective individual controller, denoted by "C", for the SLIP models. As can be seen, the controllers of the fore and hind SLIPs are not directly connected to each other. The compositional behavior is coordinated in the Reference Generator Module (RGM), whose basic function is to provide the individual controllers with desirable spatio-temporal references.

The RGM generates two event-driven sets of references for each SLIP separately: the spatial reference, z^*, and the temporal reference, s^*. However, the evolution of spatial and temporal states in the real system are not generally independent, i.e., there could be situations in which the produced references are conflicting. This issue needs to be carefully addressed in the design of controller so that the overall behavior remains feasible. In what follows, we first describe the structure of the low-level controller and then the underlying mechanism that produces and updates the appropriate references in the RGM.

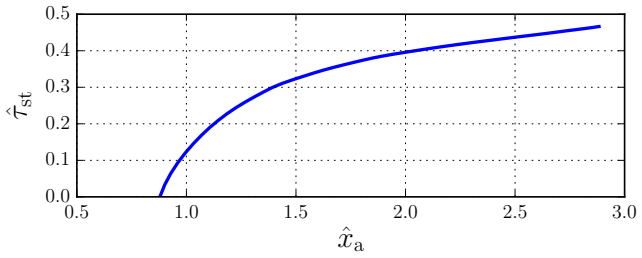

Fig. 5: Relationship between the normalized stance time, $\hat{\tau}_{\mathrm{st}} = \tau_{\mathrm{st}}/(\tau_{\mathrm{st}} + \tau_{\mathrm{sw}})$, and the normalized step length, $\hat{x}_{\mathrm{a}}(\kappa) = (x_{\mathrm{a}}(\kappa) - x_{\mathrm{a}}(\kappa-1))/l_{\mathrm{rest}}$, for different control parameters.

A. Individual controller

The majority of the existing control methods for the SLIP model in the literature focus on the stabilization of Poincaré states in the deadbeat scheme. For the temporal coordination, however, one needs to grant control authority over the event timings. Though, any adjustments on the timings could also affect the Poincaré states and vice versa.

To simplify the problem, let us assume the spatial control specifications are to keep constant the horizontal speed and vertical position, from the κth to the $(\kappa+1)$th apex at the Poincaré section. Following this, one can observe that there is a monotonic relationship between the normalized stance time, $\hat{\tau}_{\mathrm{st}} = \tau_{\mathrm{st}}/(\tau_{\mathrm{st}} + \tau_{\mathrm{sw}})$, and the normalized step length, $\hat{x}_{\mathrm{a}}(\kappa) = (x_{\mathrm{a}}(\kappa) - x_{\mathrm{a}}(\kappa-1))/l_{\mathrm{rest}}$, for different control parameters, as shown in Figure 5. Moreover, the resulting in-stride trajectories are symmetric as a consequence of the particular spatial control specifications posed. As such, the problem of controlling the horizontal position of the system at Poincaré section can easily be translated to the control of liftoff time instant since $\hat{t}_{\mathrm{l}}^*(\kappa+1) = \hat{\tau}_{\mathrm{sw}}/2 + \hat{\tau}_{\mathrm{st}}$. Taking this into consideration, we formulate the control calculations in the following optimization problem:

$$u(\kappa) = \begin{bmatrix}\alpha & k_{\mathrm{c}} & k_{\mathrm{d}}\end{bmatrix}^T = \operatorname{argmin} J, \tag{3}$$

subject to:

$$\begin{aligned} &50^{\circ} < \alpha < 90^{\circ}, \\ &k_{\mathrm{c}}, k_{\mathrm{d}} > 0, \\ &0.9 l_{\mathrm{rest}} \leq y_{\mathrm{a}}(\kappa+1) \leq 1.25 l_{\mathrm{rest}}, \end{aligned}$$

where J is the cost function:

$$\begin{aligned} J =& w_1(t_{\mathrm{l}}^*(\kappa+1) - t_{\mathrm{l}}(\kappa+1))^2 + w_2(\dot{x}_{\mathrm{a}}^*(\kappa+1) - \\ & \dot{x}_{\mathrm{a}}(\kappa+1))^2 + w_3(y_{\mathrm{a}}^*(\kappa+1) - y_{\mathrm{a}}(\kappa+1))^2. \end{aligned} \tag{4}$$

w_i, $i \in \{1,2,3\}$, are used to weigh different terms of the cost function. New control parameters are calculated when a new set of references is generated in the RGM, and only if the corresponding SLIP is in the *descent* phase.

To solve the optimization problem (3), we have tested different solvers including the `COBYLA` method of NLopt[1], and a modification of Powell hybrid method implemented in the `root` function of Scipy.optimize[2]. We eventually found the latter more effective in our particular setting as it allows defining multiple cost functions. To implement constraints, we used common parameter transformation methods and also applied necessary penalties in the cost functions.

B. Reference Generator Module (RGM)

The RGM mainly performs the following tasks:

i) updating the schedule for a given cycle index κ;
ii) producing a schedule for the next cycle for each SLIP upon its touchdown event.

In doing so, the gait specification relations, given in Functions 1 and 2 for the bounding and pronking gaits, respectively, are used. In the rest of this section, we explain why, when and how the RGM performs the above-mentioned tasks.

Why does a previously produced schedule for the κth cycle need to be updated? This is due to: (i) the limitations on the realization of the generated references, as discussed in the previous section; and (ii) deviations possibly imposed by external disturbances and system uncertainties. Moreover, according to the definition of gaits, given in Functions 1 and 2, updating the current schedule substantially influences the $(\kappa+1)$th schedule vector as well, which contains the temporal references needed in the control calculations (see (4)).

When does an update occur in the schedule? (i) Upon the occurrence of a touchdown or liftoff event of either of SLIPs in the κth cycle, the corresponding schedule vector, $s(\kappa)$, must be updated. For that, the actual event time instant is stored in the list of *just-occurred* events, $\mathcal{Q}_{\mathrm{j}}(\kappa) \in q(\kappa)$ (see Section II-A); (ii) Moreover, whenever an event (touchdown or liftoff) is expected to occur but has not occurred yet, the schedule needs to be updated as well. In this case, an internal predictor estimates a new time instant for the delayed event, and the new information is stored in the list of *expected-to-occur* events, $\mathcal{Q}_{\mathrm{e}}(\kappa) \in q(\kappa)$. Note that a delayed touchdown event influences $s(\kappa+1)$ instead of $s(\kappa)$, even though the cycle index is not actually incremented in this case.

How do we update the schedule? The update procedure is described in Algorithm 1. We shall need to elaborate on the difference between updating $s(\kappa)$ (line 5 of the algorithm) and producing $s(\kappa+1)$ (line 6). For both tasks we use the gait specifications formulated, for instance, for the bounding and pronking gaits in Functions 1 and 2. The difference between the tasks lies on the fact that we *update* a schedule (say, $s(\kappa)$) once a just-occurred/expected-to-occur event in the κth cycle is detected, while we *produce* a new schedule for subsequent uses. As such, during the execution of Functions 1 and 2 in the case of updating, the "known" temporal states are enforced by their corresponding values in $Q(\kappa)$.

[1]http://ab-initio.mit.edu/nlopt
[2]http://www.scipy.org

Algorithm 1 Reference Generator Module (RGM) functioning

1: $\kappa_{\min}$: minimum of the cycle indexes corresponding to just-occurred/expected-to-occur events
2: $\kappa_{\max} = \max(\kappa_{\mathrm{f}}, \kappa_{\mathrm{h}})$
3: $Q(\kappa)$: list of just-occurred/expected-to-occur, and already occurred events and their time instants in the κth cycle
4: **for** $\kappa = \{\kappa_{\min}, \kappa_{\min}+1, \ldots, \kappa_{\max}\}$ **do**
5: **update** $s(\kappa)$ with respect to $Q(\kappa)$
6: **produce** $s(\kappa+1)$
7: **end for**

TABLE I: Simulated system parameters

	Symbol	Value
Fore SLIP mass	m_{f}	$30\,\mathrm{kg}$
Hind SLIP mass	m_{h}	$20\,\mathrm{kg}$
Leg rest length	l_{rest}	$0.7\,\mathrm{m}$
Fore torso length	l_{f}	$0.7\,\mathrm{m}$
Hind torso length	l_{h}	$0.4\,\mathrm{m}$

IV. Simulated Gaits and Their Transitions

In this section various simulation tests are performed to evaluate the capabilities of the proposed template in generating different gaits and feasible transitions and its robustness to external disturbances (in the form of ground elevation). The numerical values of the parameters associated with the Dual-SLIP model used for the simulation are listed in Table I. The simulation was carried out using a hybrid solver that we have developed in Python 2.7. The solver is equipped with a variable step size integrator that captures the phase switchings precisely. Due to variable step size integrations, the solver must also ensure that the time evolution remains the same between the fore and hind SLIPs. Please see the attached video for the simulated gaits and their transitions.

A. Robust quadrupedal bounding

Several metrics are proposed in the literature to evaluate the disturbance rejection properties of a controlled legged robot (see [18] and references therein). In our setting, the Dual-SLIP template is assigned to converge to a bounding gait, while accommodating two variations in the ground height equal to %25 of the leg rest length. Figure 6 (top) shows the trajectories of the SLIPs together with the snapshots corresponding to the touchdown moments of either of legs. The respective horizontal and vertical ground reaction forces, normalized by the corresponding weights, are plotted in the middle and bottom panels of the figure. Moreover, the vertical positions of the SLIPs and the torso joint angle, γ, with respect to time are depicted in Figure 7. The readers may find a more complete illustration of the results in the video attachment.

As can be seen, the spatio-temporal coordination controller proposed in this paper enforces the nominal gait quickly after the disturbance (and after the sufficiently off-phase initial condition). Our simulation tests show that remarkably larger

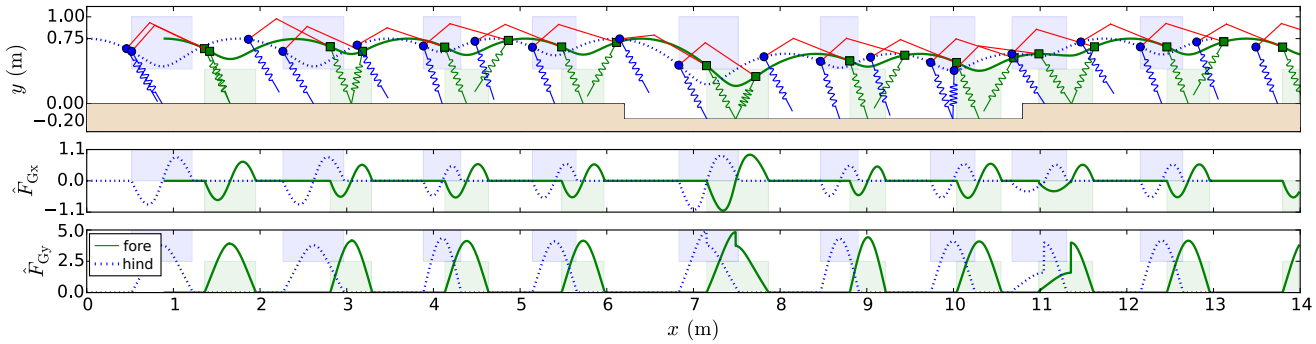

Fig. 6: Robustness evaluation of a quadrupedal bounding realized in the Dual-SLIP model. Colored rectangles represent the corresponding stance phases and white space represents swing.

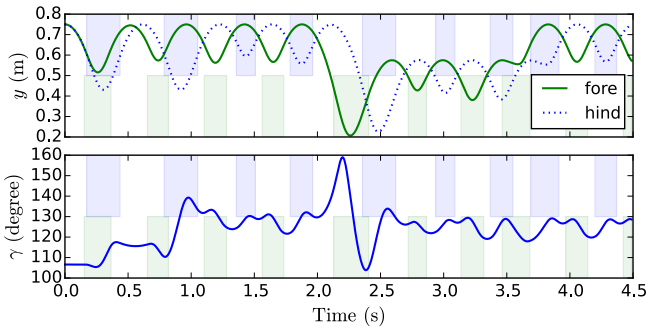

Fig. 7: Vertical positions of the SLIPs (top) and the torso angle (bottom) during the ground height variation test. Colored rectangles represent the corresponding stance phases and white space represents swing.

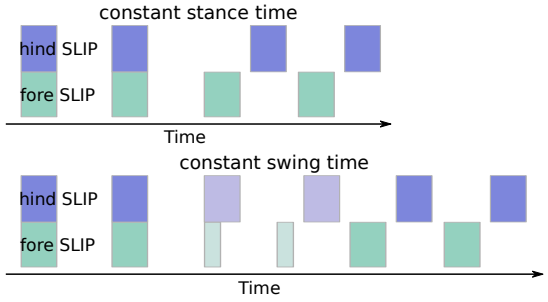

Fig. 8: Temporal references generated for the transitioning from pronking to bounding with constant stance time (top) and constant swing time (bottom). Colored rectangles represent the corresponding stance phases and white space represents swing.

step-down disturbances can also be handled by the controller. As shown in Figure 7 (bottom), γ remains far from possibly critical value $\gamma = 180^{\circ}$, ensuring that the distance between the SLIPs, d_{hf}, will not exceed $(l_{\mathrm{f}} + l_{\mathrm{h}})$.

To accommodate the ground height variations, we predict an estimated touchdown time instant and adjust the reference liftoff time instant accordingly. Thanks to this adjustment, the ground reaction forces do not experience large variations, as can be seen in Figure 6 (middle) and (bottom). However, an overly large adjustment requires larger leg flexions that can be infeasible in terms of realizing simulation results in real robotic hardware.

B. Gait transitions

In a simulation test, we now study the transition between the pronking and bounding gaits (with the same τ_{st} and τ_{sw}) in our Dual-SLIP template. Using Functions 2 and 1 consecutively during the transition, the corresponding temporal references generated are schematically depicted in Figure 8 (top). As can be seen, the stance time during the transition remains unchanged, which inevitably requires longer swing time. On the other hand, similar to what we have proposed in Section VI-B of [15], by introducing some auxiliary gaits (specifically, one or two gaits), the stance time of either of SLIPs during the transition can independently be shrunk so to achieve a unique swing time for all legs. We skip the details here due to the space limitation. The generated references following this method are depicted in Figure 8 (bottom).

Additionally, if one swaps the fore and hind leg roles in the bounding gait (Function 1), different transient behaviors will emerge during transition, although the steady-state behavior would be identical. Overall, four different cases for the transitioning from pronking to bounding can be considered:

i) bounding (Function 1) & τ_{st} constant,
ii) bounding (Function 1) & τ_{sw} constant,
iii) bounding (Function 1, swapped) & τ_{st} constant,
iv) bounding (Function 1, swapped) & τ_{sw} constant.

Figure 9 shows the simulated transitions for cases (i) (top) and (ii) (bottom)[3]. As can be seen, the duration of transitioning in case (i) is longer than case (ii). Also, notice that in case (i) the hind leg compresses significantly more than case (ii); however, the maximum magnitude of the ground reaction force in case (ii) is slightly larger than case (i).

Of particular interest is the way the coordination framework treats "idealistic" plans. Comparing the preplanned reference schedules (Figure 8) with those realized during

[3]Since the figure shows only the vertical positions, the results corresponding to cases (iii) and (iv) would be similar to cases (i) and (ii), respectively, provided that the role of fore and hind legs are swapped.

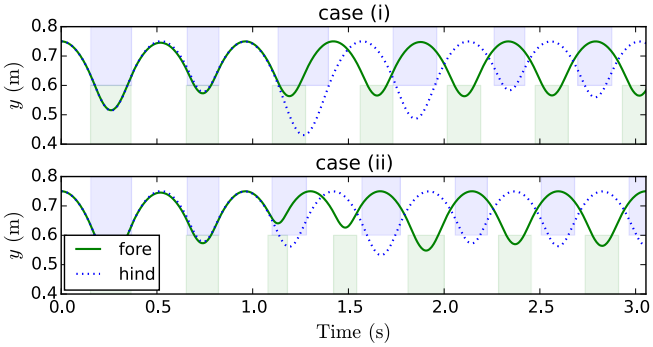

Fig. 9: Simulated transitions from pronking to bounding in the Dual-SLIP model. Colored rectangles represent the corresponding stance phases and white space represents swing.

simulation (Figure 9), one can observe that the schedules are automatically updated in accordance with the feasible behaviors designated by the real system. In other words, in case the low-level controllers cannot fulfill the assigned control specifications, this information is fed back to the RGM and a new schedule is produced subsequently.

A remarkable difference among the four possible transitions can be seen in terms of normalized relative distance, $\hat{d}_{\mathrm{hf}} = d_{\mathrm{hf}}/(l_{\mathrm{f}} + l_{\mathrm{h}})$, as shown in Figure 10. Notice that for the transition case (i) d_{hf} exceeds $(l_{\mathrm{f}} + l_{\mathrm{h}})$ for a small duration of time. However, this may not necessarily mean that the transition is infeasible, as arguments like this should be justified when anchoring the Dual-SLIP model in the real robot. Nevertheless, it would be worth exploring whether it is possible to control $\hat{d}_{\mathrm{hf}}$ in the current implementation. Our investigations show that by considering a torsional spring in the *virtual* torso joint (see Figure 1), and translating the resulting torque effect to both SLIP models, a desired $\hat{d}_{\mathrm{hf}}$ profile can be enforced.

V. Conclusion

In this paper a novel template for quadrupedal steady and transitional running has been proposed. The model represents a quadruped by means of two physically-unconnected SLIP models, coordinated in a way that the overall behavior simulates different quadrupedal gaits and their transitions. Thanks to the underlying coordination framework, the proposed template is simple to analyze, flexible with respect to control specifications, and potentially suitable for on-line implementation. We are currently developing a multi-layer control structure for a realistic quadrupedal robot model with articulated legs, in which the Dual-SLIP template will be anchored.

Acknowledgment

The first author gratefully acknowledges the support of the Government of the Islamic Republic of Iran, Ministry of Science, Research and Technology.

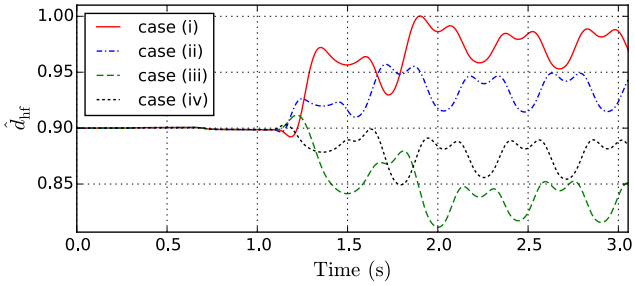

Fig. 10: Normalized relative distance, $\hat{d}_{\mathrm{hf}} = d_{\mathrm{hf}}/(l_{\mathrm{f}} + l_{\mathrm{h}})$, with respect to time for different cases of transition.

References

[1] R. J. Full and D. E. Koditschek, "Templates and anchors: neuromechanical hypotheses of legged locomotion on land," *Journal of Experimental Biology*, vol. 202, no. 23, pp. 3325–3332, 1999.

[2] M. H. Dickinson, C. T. Farley, R. J. Full, M. Koehl, R. Kram, and S. Lehman, "How animals move: an integrative view," *Science*, vol. 288, no. 5463, pp. 100–106, 2000.

[3] R. Alexander and A. Jayes, "Vertical movements in walking and running," *Journal of Zoology*, vol. 185, no. 1, pp. 27–40, 1978.

[4] R. Blickhan, "The spring-mass model for running and hopping," *Journal of biomechanics*, vol. 22, no. 11, pp. 1217–1227, 1989.

[5] R. Blickhan and R. Full, "Similarity in multilegged locomotion: bouncing like a monopode," *Journal of Comparative Physiology A*, vol. 173, no. 5, pp. 509–517, 1993.

[6] P. Nanua and K. Waldron, "Energy comparison between trot, bound, and gallop using a simple model," *Journal of biomechanical engineering*, vol. 117, no. 4, pp. 466–473, 1995.

[7] K. Leeser, "Locomotion Experiments on a Planar Quadruped Robot with Articulated Spine, Master's thesis," 1996.

[8] I. Poulakakis, E. Papadopoulos, and M. Buehler, "On the stable passive dynamics of quadrupedal running," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 1. IEEE, 2003, pp. 1368–1373.

[9] Z. G. Zhang, Y. Fukuoka, and H. Kimura, "Stable quadrupedal running based spring-loaded two-segment legged on a model," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3. IEEE, 2004, pp. 2601–2606.

[10] U. Culha and U. Saranli, "Quadrupedal bounding with an actuated spinal joint," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 1392–1397.

[11] A. K. Valenzuela and S. Kim, "Optimally scaled hip-force planning: A control approach for quadrupedal running," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2012, pp. 1901–1907.

[12] Q. Cao and I. Poulakakis, "Quadrupedal bounding with a segmented flexible torso: passive stability and feedback control," *Bioinspiration & biomimetics*, vol. 8, no. 4, p. 046007, 2013.

[13] S. Pouya, M. Khodabakhsh, A. Spröwitz, and A. Ijspeert, "Spinal joint compliance and actuation in a simulated bounding quadruped robot," *Autonomous Robots*, pp. 1–16, 2015.

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

[15] G. Lopes, B. Kersbergen, T. van den Boom, B. De Schutter, and R. Babuska, "Modeling and Control of Legged Locomotion via Switching Max-Plus Models," *IEEE Trans. on Robotics*, vol. 30, no. 3, pp. 652–665, 2014.

[16] M. Shahbazi and G. A. D. Lopes, "A Max-Plus Based Synchronization Controller for Multiple Spring-Mass Hoppers," in *Proceedings of IEEE 54st Annual Conference on Decision and Control (CDC)*, Dec 2015, pp. 1669–1674.

[17] O. Arslan, U. Saranli, and O. Morgul, "An approximate stance map of the spring mass hopper with gravity correction for nonsymmetric locomotions," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 2388–2393.

[18] J. D. Karssen and M. Wisse, "Running with improved disturbance rejection by using non-linear leg springs," *The International Journal of Robotics Research*, vol. 30, no. 13, pp. 1585–1595, 2011.