2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

An Adaptive Supervisory Control Approach to Dynamic Locomotion Under Parametric Uncertainty

Prem Chand, Sushant Veer, and Ioannis Poulakakis

Abstract— **This paper presents an adaptive control scheme for robotic systems that operate in the face of—potentially large—structured uncertainty. The proposed adaptive controller employs an on-line supervisor that utilizes logic-based switching among a finite set of controllers to identify uncertain parameters, and adapt the behavior of the system based on a current estimate of their value. To achieve this, the adaptive control approach in this paper combines on-line parameter estimation and feedback control while avoiding some of the inherent difficulties of classical adaptive control strategies. Furthermore, the proposed supervisory control architecture is modular as it relies on established "off-the-shelf" feedback control law and estimator design approaches, instead of customizing the overall design to the specific requirements of an adaptive control algorithm. We demonstrate the efficacy of the method on the problem of a dynamically-walking bipedal robot delivering a payload of unknown mass, and show that, by switching to the controller that is the "best" according to a current estimate of the uncertainty, the system maintains a low energy cost during its operation.**

I. Introduction

To extend their operational capabilities, robotic platforms must adapt to uncertainty that arises partially due to the nature of the task at hand or due to the inexact modeling of the system. Our focus in this paper is on structured uncertainty; i.e., the case of unknown parameters with values in some known set. While a single controller provides the ability to handle only a certain amount of uncertainty, combining multiple controllers designed for different ranges of uncertain parameters can enlarge the span of influence of the augmented control strategy in order to accomplish the task without inadvertently impairing the system's performance.

We propose here a supervisory adaptive control architecture that achieves adaptation by leveraging the availability of a library of feedback control laws and orchestrating switching among them in an on-line fashion. Supervisory control approaches to adaptation have been rigorously studied in the control systems literature—see [1], [2] and [3] for an overview—and have also been utilized to adapt underwater robots to large-scale uncertainty [4]. However, these approaches rely on the fundamental assumption that the closed-loop systems resulting from the controllers in the library, all share a *common* equilibrium. This assumption—though not very restrictive in certain settings—can be quite limiting in many robotics applications where the individual dynamical movement primitives [5] that are used to tackle drastically varying conditions do not share a common equilibrium point.

To address these limitations, our recent work [6], [7] developed theoretical tools for studying switching among systems with *multiple* equilibria; these tools are relevant to motion planning [8]–[10] and collaborative object transportation [7] in the context of dynamically-stable bipeds. The focus in these papers has been on utilizing switching for robust stabilization in the presence of *known* planning commands or *measured* exogenous input signals, which though vary so widely that cannot be accommodated by a single controller. As a matter of fact, the idea of achieving robust dynamic locomotion by switching[1] among a finite collection of feedback controllers has been used successfully in a variety of applications, ranging from hopping with varying speeds [18] to walking on rough terrain [19]–[21].

The supervisory control architecture proposed here is similar in flavor to the aforementioned approaches in that it uses switching among pre-existing controllers. However, the fundamental difference is that the supervisor incorporates an on-line estimation unit, which enables controller switching on the basis of current estimates of the uncertainty. This way the supervisor can select the "best" feedback controller in a truly adaptive fashion—i.e. through the current estimate of the unknown parameters—resulting in substantial performance improvements compared to typical robustness approaches. Indeed, the later provide "worst-case" guarantees of stability without necessarily addressing performance. An exception is [22], which addresses adaptive control design for a dynamically-stable bipedal robot through a single control law obtained using a Lyapunov-based method akin to [23].

In this paper, we propose a supervisory approach to the case where a dynamic bipedal walker is tasked with transporting an object, the mass of which is unknown. The low-level controllers are designed on the basis of the Hybrid Zero Dynamics (HZD) method. Note however that the proposed supervisory controller is *not* tied to a particular design technique for constructing the library of controllers; any other dynamic locomotion controller design method could have been used, e.g. [23]–[26]. Using on-line estimation and a monitoring signal generator, the supervisor's switching logic

P. Chand and I. Poulakakis are with the Department of Mechanical Engineering, University of Delaware, Newark, DE 19716, USA; e-mail: {rhaegar, poulakas}@udel.edu. S. Veer is with the Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ 08544, USA; e-mail: sveer@princeton.edu.

This work is supported in part by NSF CAREER Award IIS-1350721 and by NRI-1327614.

[1] In this literature review, we focus on switching approaches. An alternative approach relies on "robustifying" the *individual* controllers instead of switching among them. In the context of dynamic locomotion, control techniques along this vein include, for example, [11] that addresses parameter mismatch; [12]–[14] that address rough terrain; and [15]–[17] that accommodate exogenous forces. These approaches can be combined with supervised switching to extend the range of operation of the robot.

978-1-7281-7395-5/20/$31.00 ©2020 IEEE

decides to place in the feedback loop the controller, which is currently the "best" based on a suitable notion of proximity to the unknown parameter. Adaptation is fast, typically occurs within 2-3 steps, and it leads to considerable reduction in the energy consumption of the biped by roughly 18% per step compared to the case where a single, sufficiently robust, controller is used. Note finally that this paper concentrates primarily on performance and implementation aspects of the method. A rigorous stability analysis of the method uses the theoretical developments of [7] and [27, Chapter 6], and will not be presented here due to space constraints.

II. Overview: Adaptation by Supervised Switching

This section describes the proposed supervisory control architecture in general terms; see also Fig. 1. Our intention here is to highlight the broad applicability of the method and its advantages; subsequent sections particularize the relevant constructions to the design of an adaptive dynamic locomotion controller. We begin by considering a system, the true model $\mathcal{F}_{p^*}$ of which belongs to the family

$$\mathbb{F} = \bigcup_{p \in \mathcal{P}} \mathcal{F}_p \tag{1}$$

of admissible (nominal) models. In (1), $\mathcal{P}$ is a compact—possibly *uncountable*—index set that captures the range of the uncertainty[2] and each $\mathcal{F}_p$ represents the nominal model corresponding to $p \in \mathcal{P}$. It is emphasized that although p^* is known to lie in $\mathcal{P}$, its exact value is *unknown* and so is the actual system model $\mathcal{F}_{p^*} \in \mathbb{F}$.

A. Library of feedback control laws: General setting

The approach relies on the availability of a library of parameterized candidate feedback control laws Γ_r together with their corresponding "authority" sets $\mathcal{D}_r \subset \mathcal{P}$, i.e.,

$$\mathbb{C} = \bigcup_{r \in \mathcal{R}} \{\Gamma_r, \mathcal{D}_r\} \ , \tag{2}$$

[2]For the case of parametric uncertainty where $\mathcal{F}_p$ depends on a collection of l parameters, the set $\mathcal{P}$ is a compact subset of $\mathbb{R}^l$.

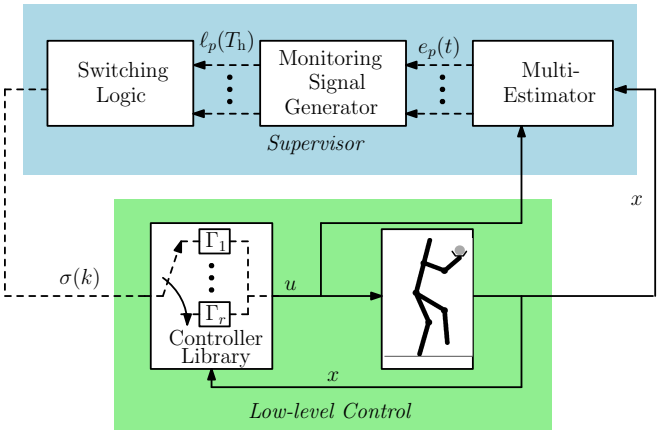

Fig. 1. Supervisory control block diagram. The high-level supervisor is shown in blue and the low-level execution loop is shown in green. At every step, the switching signal $\sigma(k)$ engages a controller from the controller library and the within-step supervisor assesses its performance using the multi-estimator and the monitoring signal generator components.

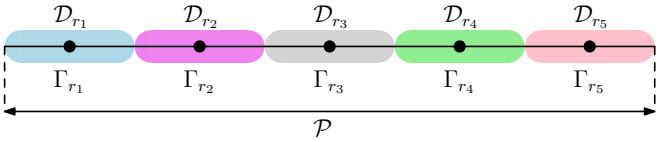

Fig. 2. Illustration of finite controllers spanning a set $\mathcal{P}$ within which an uncertain parameter takes values. The controller Γ_r is designed for a nominal parameter value $p_r \in \mathcal{P}$ and it works when the true parameter value p is in the corresponding control authority set $\mathcal{D}_r$.

where $\mathcal{R}$ is an index set. The standing assumption is that the collection of controllers $\{\Gamma_r,\ r \in \mathcal{R}\}$ is sufficiently rich so that every admissible system model $\mathcal{F}_p \in \mathbb{F}$ behaves according to certain stability and performance specifications when placing in the feedback loop a suitable controller Γ_r for some $r \in \mathcal{R}$, as Fig. 2 illustrates.

It is important to note that the index sets $\mathcal{P}$ and $\mathcal{R}$ can, in general, be different; in fact, the case where the parameter uncertainty set $\mathcal{P}$ is *uncountable* while the controller set $\mathcal{R}$ is *finite* often emerges in practical applications. The understanding here is that the controllers are designed so that $\mathcal{P}$ can be partitioned by the corresponding control authority sets $\mathcal{D}_r$ so that (i) $\mathcal{P}$ is covered by the subsets $\mathcal{D}_r$, i.e., $\mathcal{P} = \bigcup_{r \in \mathcal{R}} \mathcal{D}_r$, and (ii) at least one controller Γ_r achieves the desired control objectives for the subfamily of systems $\mathcal{F}_p$ with p ranging in $\mathcal{D}_r$. The latter requirement naturally defines a *controller assignment map* $\chi : \mathcal{P} \to \mathcal{R}$ by the rule $r = \chi(p)$ meaning that the controller with index $\chi(p) \in \mathcal{R}$ is responsible for $\mathcal{F}_p$ when $p \in \mathcal{D}_r$; see Fig. 2

Partitioning $\mathcal{P}$ to the subsets $\mathcal{D}_r$ depends on the control objectives. It is not necessary to select the regions $\mathcal{D}_r$ solely based on stabilization considerations, particularly in the case where—in addition to stability—performance is important. Suppose, for example, that a single controller Γ_r can stabilize the entire family of systems $\mathbb{F}$ in (1), but its performance degrades when the system's unknown parameters are significantly different from those in the nominal model used to design Γ_r. In this case, it would still be beneficial to employ a collection of controllers to ensure that performance does not decline to an unacceptable level. Finally, it is emphasized that the controllers used in $\mathbb{C}$ need *not* necessarily give rise to the same (nominal) equilibrium behavior. In fact, as we will see in the bipedal robot example discussed below, different controllers can correspond to different (nominal) walking gaits, thus providing additional flexibility in improving the performance of the system despite the presence of parameter uncertainty. This additional capability significantly extends existing supervisory control approaches as in [3, Chapter 6] at the expense of requiring additional theoretical tools, which have been developed in the authors' previous work [6].

B. Supervisor: Fundamental building blocks

The objective of the supervisor is to devise an on-line strategy for deciding which controller $\Gamma_r \in \mathbb{C}$ must be placed in the feedback loop to realize the "best" performance given the current state and input of the system. As shown in Fig. 1, the supervisor is composed of three main blocks: (i) the multi-estimator; (ii) the monitoring signal generator;

and (iii) the switching logic. In more detail, the multi-estimator is a dynamical system which monitors the state x and the input u applied to the actual system, and generates a collection of estimator errors $e_p = \hat{x}_p - x$, $p \in \mathcal{P}$, measuring the difference between the actual state x and its estimate $\hat{x}_p$ should $\mathcal{F}_p$ were the actual system dynamics. Then, the monitoring signal generator processes the estimation errors to construct suitable monitoring signals ℓ_p, $p \in \mathcal{P}$, that will be used to make decisions as to which controller should be placed in the loop. This is realized by the switching logic, which effectively selects out of the library $\mathbb{C}$ the controller which corresponds to the smallest monitoring signal. The end result is a piecewise constant switching signal σ, the value of which determines the index r of the controller that is active at each step of the process.

In the following sections, we particularize the supervisory control scheme described above to an application in which a dynamically walking biped is tasked with transporting an object of unknown mass; see Fig. 3. Before continuing through, it is important to highlight here a few advantages of the proposed approach. One advantage is *modularity*: the behavior of the overall system relies on the properties of the individual controllers that are placed in the loop, but not on the design and implementation specifics of how these properties are achieved by each controller. An immediate consequence of this aspect is the additional advantage of being able to use "off-the-shelf" feedback control design methods as long as the combined span of influence of the resulting controllers covers the parameter uncertainty set, as depicted in Fig. 2. In practical applications, the ability to re-use existing control methods provides greater flexibility as it does not require tailoring the control design approach to the specifics of the application at hand.

III. Task: Carrying an Unknown Mass

To demonstrate the general approach described above, we consider the model of Fig. 3 that corresponds to the morphology of the bipedal robot Rabbit [28, Table 6.3, pp 177], modified to include a 2-link manipulator as in [29]. The task is to transport an object of unknown mass and to provide an estimate of its actual value $p^* = m_{\mathrm{u}}$. To simplify the exposition of the main ideas, we will assume that the rest of the dynamic parameters of the system are known; note however that, with minor modifications, the method can be applied to the case where multiple parameters are unknown.

The model has seven degrees of freedom (DoF), six of which are directly actuated and correspond to the hip and knee joints of the two legs, and the shoulder and elbow joints of the manipulator. There is one unactuated DoF representing the contact between the toe of the leg providing support and the ground. Figure 3 presents a choice of generalized coordinates q taking values in a set Q that includes feasible robot configurations. Let $x := (q^{\mathrm{T}}, \dot{q}^{\mathrm{T}})^{\mathrm{T}} \in TQ$ denote the system's state. Following [28], walking can be modeled as alternating sequences of swing and instantaneous double support phases, resulting in the system

$$\mathcal{F}_p : \begin{cases} \dot{x} = f_p(x) + g_p(x)u, & x \in TQ \setminus \mathcal{S} \\ x^+ = \Delta_p(x^-), & x^- \in \mathcal{S} \ . \end{cases} \tag{3}$$

Here, f_p and g_p describe the continuous-time dynamics during the swing phase under the control inputs u, and the map Δ_p captures the effect of the impact of the swing leg with the ground surface

$$\mathcal{S} := \{(q, \dot{q}) \in TQ \mid \mathrm{p_v}(q) = 0, \ \dot{\mathrm{p}}_{\mathrm{v}}(q, \dot{q}) < 0\} \ , \tag{4}$$

where $\mathrm{p_v}(q)$ is the height of the swing foot. The dependence of the maps f_p, g_p and Δ_p on the parameter p representing the unknown mass of the object is made explicit in (3) to illustrate the fact that as p varies in the uncertainty set $\mathcal{P}$ an entire family $\mathbb{F}$ of nominal models $\mathcal{F}_p$ is generated; see (1). However, since the true value p^* of p is unknown, the actual system model $\mathcal{F}_{p^*}$ is unknown. Finally, note that, in this example, the uncertainty set $\mathcal{P}$ is a compact interval of $\mathbb{R}$ that includes all possible mass values of interest.

IV. Adaptive Supervisory Control Design

This section provides information regarding the design of the library of low-level controllers and the high-level supervisor of the proposed architecture; see Fig. 1.

A. Library of feedback control laws

Our task here is to construct a library $\mathbb{C}$ of suitable feedback control laws Γ_r, as described in Section II. Note that an important advantage of the proposed approach is that established control synthesis methods can be used to design the individual controllers. Thus, devising new methods for addressing uncertainty that is larger than what each controller can handle individually without negatively affecting performance is not necessary.

To begin, pick some nominal parameter value $p_r \in \mathcal{P}$; this corresponds to selecting a *candidate* value for the mass of the object. Then, use *any* "off-the-shelf" locomotion control design method to construct a feedback controller

$$u = \Gamma_r(x) \tag{5}$$

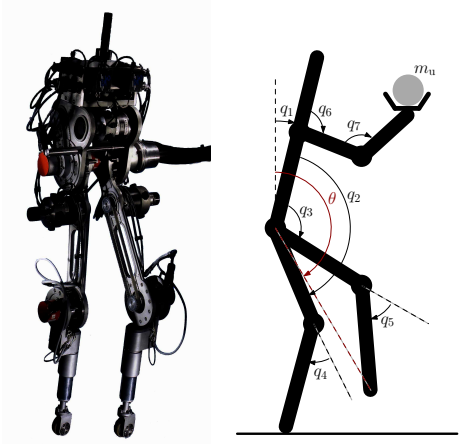

Fig. 3. The uderactuated bipedal robot Rabbit (left) and a corresponding model with a manipulator carrying an object of unknown mass m_{u} (right).

capable of generating exponentially stable walking gaits in the nominal plant model $\mathcal{F}_{p_r}$. In more detail, the nominal system (3) for $p = p_r$ in closed-loop with (5) becomes

$$\mathcal{F}_{p_r}^{\text{cl}}: \begin{cases} \dot{x} = f_{p_r}^{\text{cl}}(x), & x \in TQ \setminus \mathcal{S} \\ x^+ = \Delta_{p_r}(x^-), & x^- \in \mathcal{S} \ , \end{cases} \tag{6}$$

where

$$f_{p_r}^{\text{cl}}(x) = f_{p_r}(x) + g_{p_r}(x)\Gamma_r(x) \ .$$

If the controller (5) is suitably designed, the system (6) possesses an exponentially stable limit cycle $\mathcal{O}_r$, which corresponds to the fixed point $\bar{x}_r$ of the discrete system

$$x_{k+1} = P_r(x_k) \ , \tag{7}$$

where P_r is the Poincaré map associated with (6). Among several alternatives for designing Γ_r, in this work we have used the Hybrid Zero Dynamics (HZD) method due to its experimental success[3] in stabilizing periodic gaits on Rabbit; see [28] for implementation details.

Repeating this process for different choices of parameter values p_r one can construct a finite collection of controllers Γ_r indexed by the corresponding set $\mathcal{R}$, which constitute the library of control laws $\mathbb{C}$. It is important to emphasize that, unlike the classical supervisory control approach described in [3, Chapter 6] where *all* controllers Γ_r stabilize the *same* equilibrium, in our approach different controllers correspond to different equilibrium behaviors; that is, $\mathcal{O}_{r_1} \neq \mathcal{O}_{r_2}$ when $r_1 \neq r_2$. As a result, the range of behaviors of the system can be significantly expanded.

Next, we analyze the performance of Γ_r in closed loop with the nominal model $\mathcal{F}_p$ given by (3) with p different from the one used to design Γ_r, i.e., $p \neq p_r$. This results in the perturbed closed-loop system

$$\mathcal{F}_{p}^{\text{cl}}: \begin{cases} \dot{x} = f_{p_r}^{\text{cl}}(x) + \upsilon_p, & x \in TQ \setminus \mathcal{S} \\ x^+ = \Delta_{p_r}(x^-) + w_p, & x^- \in \mathcal{S}. \end{cases} \tag{8}$$

where $f_{p_r}^{\text{cl}}$ and Δ_{p_r} are the maps defining $\mathcal{F}_r^{\text{cl}}$ in (6) and

$$\begin{aligned} \upsilon_p &= [f_p(x) - f_{p_r}(x)] + [g_p(x) - g_{p_r}(x)] \ \Gamma_r(x) \\ w_p &= \Delta_p(x^-) - \Delta_{p_r}(x^-) \end{aligned}$$

are "disturbance" signals due to the discrepancy between the maps f_p, g_p, Δ_p corresponding to the nominal system model with parameter p and the maps $f_{p_r}, g_{p_r}, \Delta_{p_r}$ used to design the controller Γ_r. Note that, in the absence of the disturbances, (8) reduces to (6). As a result, since $\bar{x}_r$ is a locally exponentially stable fixed point of (7), [16, Theorems 1 and 2] can be used to establish that the corresponding limit cycle $\mathcal{O}_r$ of (6) is a locally input-to-state stable limit cycle of (8). This property essentially implies that trajectories will remain close to $\mathcal{O}_r$ for reasonably small parameter mismatch.

[3]Note that a useful property of the HZD method is its inherent dimensional reduction, which is important for analyzing the performance of each controller and its implications to safe switching, as in [6], [7]. Due to space limitations, a detailed analysis of the stability properties of the supervisory control scheme will be presented elsewhere.

B. Supervisor

We now describe the structure of the supervisor, which given the library $\mathbb{C}$ of controllers, makes on-line decisions regarding which controller $\Gamma_r \in \mathbb{C}$ should be placed in the loop to facilitate adaptation. In the context of the bipedal robot example, it is natural to assume that the supervisor monitors the continuous-time evolution of the state of the system and its input during the current step, and uses that information to decide which controller to apply at the next step provided that certain safety conditions are satisfied.

1) Multi-estimator: Since the supervisor uses the within-step values of the state and the input, the design of the multi-estimator will focus on the continuous-time part of (3). The objective of the multi-estimator is to generate an estimate $\hat{x}_p$ of the state x if the actual (unknown) system were the nominal model with parameter $p \in \mathcal{P}$ regardless of the control input that is applied. In other words, we would like $\hat{x}_p$ to converge to x if the actual system is the nominal model with parameter p. To achieve this, we design the multi-estimator as follows. Suppose that $p^* \in \mathcal{P}$ is the true, albeit unknown, value of the parameter so that the dynamics of the actual system in continuous time is

$$\dot{x} = f_{p^*}(x) + g_{p^*}(x)u \ , \tag{9}$$

where x and u are the state and the control input of the actual system, respectively. Then, the multi-estimator takes as inputs the state x and control values u of (9) and produces the estimate $\hat{x}_p$ according to

$$\dot{\hat{x}}_p = A_p(\hat{x}_p - x) + [f_p(x) + g_p(x)u], \quad p \in \mathcal{P} \ , \tag{10}$$

where the matrices $\{A_p, \ p \in \mathcal{P}\}$ are assumed to be Hurwitz. The understanding here is that $\hat{x}_p$ provides an approximation of x in the sense that if p were equal to the true value p^*, the estimation error $e_{p^*} = \hat{x}_{p^*} - x$ would converge to zero according to $\dot{e}_{p^*} = A_{p^*} e_{p^*}$. Hence, owing to the design of the multi-estimator (10), the estimation error e_{p^*} decreases at an exponential rate. On the other hand, there is no reason in general for the rest of the estimation errors e_p to also decrease or to be small. This intuitive observation implies that it is reasonable to select the index $\hat{p}$ of the smallest estimation error in the collection $\{e_p, \ p \in \mathcal{P}\}$ as a current estimate of p^*. Below we discuss this issue further.

2) Monitoring signal generator: Instead of just using the instantaneous values $e_p(t)$ of the error signals produced by (10) to make decisions, it is advantageous to incorporate their past values in the decision-making process. To take this requirement into account, the supervisor incorporates a monitoring signal generator, i.e., a filter with inputs the error signals $\{e_p, \ p \in \mathcal{P}\}$ and outputs the corresponding monitoring signals, which we define as

$$\ell_p(t) = \int_0^t \|e_p(s)\|^2 ds, \quad p \in \mathcal{P} \ . \tag{11}$$

A decision must be issued by the end of the current step, so that the selected controller can be implemented at the next step, provided that certain *safety* guarantees are satisfied;

this issue will be discussed further in Section V below. As a result, during each step, a finite time window $[0, T_{\rm h}]$ is available for generating the monitoring signals, where the horizon $T_{\rm h}$ should not exceed the total duration $T_{\rm s}$ of the step[4], i.e., $T_{\rm h} \leq T_{\rm s}$. Then, at the end of the within-step window $[0, T_{\rm h}]$, the output of the monitoring signal generator is the collection $\{\ell_p(T_{\rm h}),\ p \in \mathcal{P}\}$ that will be used by the switching logic to make a decision. Note here that, once a step is completed and a decision is made regarding the controller that will be placed in the loop, the estimation errors e_p and the corresponding monitoring signals ℓ_p are reset to zero for all $p \in \mathcal{P}$, thus providing a "forgetting" effect.

3) Switching logic: According to the discussion of the multi-estimator, we will select the current estimate $\hat{p}$ of the unknown parameter p^* to be the index of the smallest element in $\{\ell_p(T_{\rm h}),\ p \in \mathcal{P}\}$; that is,

$$\hat{p} = \arg\min_{p \in \mathcal{P}} \ell_p(T_{\rm h}) \ . \tag{12}$$

With reference to Fig. 2, since $\hat{p} \in \mathcal{P} = \bigcup_{r \in \mathcal{R}} \mathcal{D}_r$, there is (at least) one set $\mathcal{D}_r$ such that $\hat{p} \in \mathcal{D}_r$. Then, the index $r = \chi(\hat{p})$ with χ being the control assignment map associated with the library $\mathbb{C}$ is the index of the controller that is turned active in the next step. This process gives rise to a switching signal $\sigma : \mathbb{Z}_+ \to \mathcal{R}$, which maps the current step number k to the index $\chi(p) \in \mathcal{R}$ of the controller that is executed at the k-th step; i.e., $\sigma(k) = \chi(p)$.

V. Implementation and Results

This section discusses a few important implementation aspects of the method, namely safety and on-line computation, and concludes with a numerical evaluation of the approach.

A. Implementation aspects

1) Safety: In general implementations of the method, one has to ensure that switching among different controllers does not destabilize the system. As discussed above, the decisions issued by the supervisor can be represented as a "descending" switching signal $\sigma : \mathbb{Z}_+ \to \mathcal{R}$ mapping step number to the controller that is turned active at that step. The process gives rise to the discrete switched system with *multiple* equilibria

$$x_{k+1} = P_{\sigma(k)}(x_k, d_k) \ , \tag{13}$$

where P is the perturbed Poincaré map associated with (8), and d_k captures the parameter mismatch between (6) and (8). The stability properties of (13) can be analyzed in a tractable fashion using the theoretical tools developed in [6], [7]. These tools provide explicit *average dwell-time* constraints [30] on the switching signal, so that the state of (13) remains trapped within a compact set Ω_2 provided that the initial conditions are within a compact set $\Omega_1 \subset \Omega_2$. Obtaining such safety conditions *analytically* will be presented elsewhere due to space constraints. Instead, below we will employ a numerical approach to derive a *fixed* dwell-time bound $N_{\rm d} \geq 1$ which corresponds to the minimum number of steps that must be completed before the supervisor can switch *safely* to a new controller.

2) Computational considerations: The implementation of the supervisory controller as was described above relies on solving, at each step, the optimization problem (12), which, in turn, entails the construction of the multi-estimators (10) and the computation of the monitoring signals (11). This task can be computationally demanding, particularly when the set $\mathcal{P}$ is uncountable. One idea that can be used to reduce the computational burden is state sharing [3, Chapter 6], which results in low-dimensional representations of the multi-estimator. However, this method requires a special system structure, and still does not resolve the problem of computing (12). These issues are addressed here by employing a suitable discretization $\tilde{\mathcal{P}}$ of the set $\mathcal{P}$ at the expense of finding the true value p^* of the unknown parameter p only approximately, at an accuracy that depends on the resolution of the discretization. This modification relies on engaging, at each step, only a subset of the multi-estimators associated with the discrete set $\tilde{\mathcal{P}}$ according to a binary search procedure. This procedure significantly accelerates computation and can be scaled to the case where multiple parameters need to be estimated using multidimensional binary search trees [31].

B. Numerical evaluation

To numerically evaluate the approach, we assume that the unknown mass of the payload can take any value between 0 and 400 grams, i.e., $\mathcal{P} = [0,\ 400]$. These numbers are intentionally chosen so as not to challenge the robustness of the individual controllers in order to demonstrate the advantages of adaptation to performance; in the example here performance is measured in terms of energy efficiency.

As in Section V-A.2, the uncertainty set $\mathcal{P}$ is discretized as $\tilde{\mathcal{P}} := \{0, 50, ..., 400\}$ and, for each $p_r \in \tilde{\mathcal{P}}$, a nominal controller Γ_r, $r \in \mathcal{R} = \{1, 2, ..., 9\}$, is designed using the procedure in Section IV-A, Each of these controllers generates an exponentially stable limit cycle $\mathcal{O}_r$, $r \in \mathcal{R}$, with $\mathcal{O}_{r_1} \neq \mathcal{O}_{r_2}$ for $r_1, r_2 \in \mathcal{R}$ such that $r_1 \neq r_2$. As robustness is not an issue here[5], one could implement Γ_1 corresponding to zero mass to stabilize walking gaits for any payload with mass taking values in $\mathcal{P}$. However, as will be seen shortly, this choice causes degraded performance, as it increases energy consumption.

Switching among these controllers results in a perturbed discrete switched system with multiple equilibria of the form (13). Although stating analytical conditions that exclude unsafe switching scenarios is tractable using [6], [7], such analysis would take us too far afield. Instead, we follow here a simulation-based procedure similar to that suggested in [8]. In more detail, we densely discretize $\mathcal{P}$ in 400 candidate masses one gram apart, and, for each candidate mass, we exhaustively test switching scenarios by engaging controllers at each step for 100 steps according to a uniform distribution.

[4]The selection of the horizon $T_{\rm h}$ can affect decisions, particularly when noise is present. In general, $T_{\rm h}$ can be selected close to $T_{\rm s}$ while allowing ample time for switching in preparation of the next step.

[5]If the objective were to extend the range of uncertainty beyond $[0,\ 400]$, additional controllers corresponding to larger unknown masses could have been added in the library.

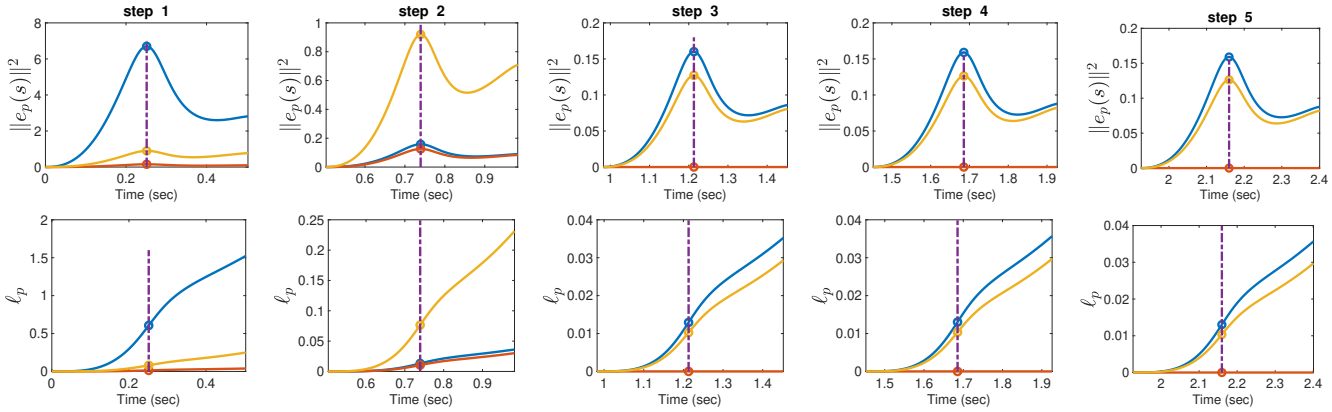

Fig. 4. Evolution with respect to time of the estimator errors $\|e_p\|^2$ (top) and monitoring signals ℓ_p (bottom) computed by (10) and (11), respectively, for the first five steps of the biped carrying an unknown mass $p^* = 250$. In step k, the supervisor employs only three estimators from (10) corresponding to indices $p \in \tilde{\mathcal{P}}^k = \{a, b, c\}$, which vary from one step to the next based on a binary search algorithm. In this case, $\tilde{\mathcal{P}}^1 = \{0, 400, 200\}$, $\tilde{\mathcal{P}}^2 = \{200, 400, 300\}$, $\tilde{\mathcal{P}}^3 = \{200, 300, 250\}$, $\tilde{\mathcal{P}}^k = \{250, 200, 300\}$ for $k \geq 4$. The vertical lines denote the time window T_h available for the decision. Note the rapid reduction in the values of the monitoring signals, which is responsible for the high adaptation rates.

This results in a total of 40,000 simulated steps without any falls observed, suggesting that a fixed dwell time $N_\mathrm{d} = 1$ can be selected. Although this estimate is not rigorous, it represents strong evidence that the supervisor can, in general, select a new controller at every step.

Next, we turn our attention to the design of the multi-estimator and the monitoring signal generator according to Sections IV-B.1 and IV-B.2, respectively. Note that directly replacing the (uncountable) uncertainty set $\mathcal{P}$ in the optimization problem (12) with the finite set $\tilde{\mathcal{P}}$ is still computationally intensive; thus, as was mentioned in Section V-A.2, we resort to a binary search method as follows. At step k the supervisor employs only three estimators corresponding to parameter values in $\tilde{\mathcal{P}}^k = \{a, b, c\} \subset \tilde{\mathcal{P}}$, resulting in the corresponding monitoring signal values $\mathbb{L}^k = \{\ell_a^k(T_\mathrm{h}), \ell_b^k(T_\mathrm{h}), \ell_c^k(T_\mathrm{h})\}$, once the estimation horizon T_h is reached. Then, the supervisor chooses to apply, at the next step $k+1$, the controller corresponding to the index $\arg\min_{i \in \tilde{\mathcal{P}}^k} \mathbb{L}^k$ of the smallest element in $\mathbb{L}^k$. In addition, it updates the indices in the new $\tilde{\mathcal{P}}^{k+1}$ through the replacements $a \leftarrow \arg\min_{i \in \tilde{\mathcal{P}}^k} \mathbb{L}^k$, $b \leftarrow \arg\min_{i \in \tilde{\mathcal{P}}^k \setminus \{a\}} \mathbb{L}^k$ and $c \leftarrow (a+b)/2$. The process continues as long as $\tilde{\mathcal{P}}^{k+1} \subset \tilde{\mathcal{P}}$. If this condition is violated, the supervisor keeps the best candidate controller corresponding to the previous step for all subsequent sets.

Figure 4 shows the time evolution of the estimator errors and the corresponding monitoring signals for the first five steps of a typical application of the algorithm. The estimators employed by the supervisor at each step vary according to the binary search algorithm described above. Note that the monitoring signals that are used by the supervisor in its decision-making process are monotonically increasing with time. The algorithm estimates the unknown parameter value p^* in three steps; in this case, $p^* = 250$. Overall, the method converges, within 2-3 steps, to an estimate $\hat{p}$ of the parameter p that is within at most 25 grams away from the unknown mass p^*, for all values $p^* \in \mathcal{P}$. Finally, Fig. 5 demonstrates the performance of the algorithm in terms of energy consumption. Switching to the controller corresponding to the estimated parameter value results in a minimally perturbed limit cycle, and the overall energy expenditure is kept low compared to the case where a single controller is used to cover the entire uncertainty range. The corresponding energy savings per step can be as much as 18% for parameter mismatch at the order of 400 grams.

VI. Conclusion

We presented a supervisory control scheme that enables adaptation to structured uncertainties (i) by estimating the unknown parameter and (ii) by orchestrating switching among a finite family of controllers. Our approach allows the use of off-the-shelf controllers, thereby, offering ease of implementation—by saving the effort of designing an adaptive controller—flexibility, and modularity. We applied the method to adapt the locomotion pattern of a dynamically walking biped tasked with carrying a payload of unknown mass. The supervisor converges, within 2-3 steps, to a suitable controller in the vicinity of the true value of the unknown parameter. As a result, the proposed approach offers enhanced performance by reducing the energetic cost of the task compared to the case of no switching.

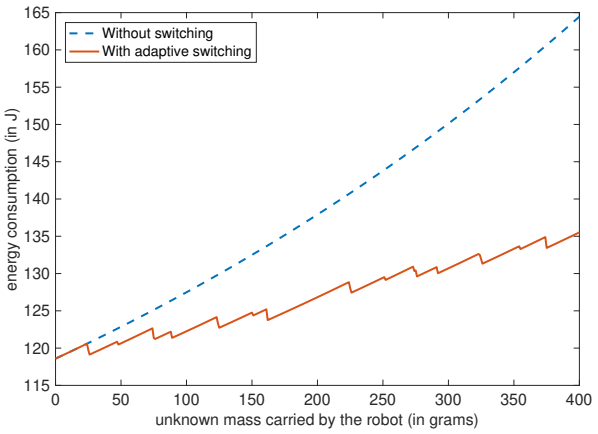

Fig. 5. Comparison of energy cost for 30 steps with the supervisor governing switching (continuous line) and with the controller Γ_1 corresponding to zero mass applied without switching (dashed line). The energy consumption increases at a much faster rate when no switching is employed.

REFERENCES

[1] A. S. Morse, "Supervisory control of families of linear set-point controllers, part 1: Exact matching," *IEEE Tr. on Automatic Control*, vol. 41, pp. 1413–1431, 1996.

[2] ——, "Supervisory control of families of linear set-point controllers, part 2: Robustness," *IEEE Tr. on Automatic Control*, vol. 42, pp. 1500–1515, 1997.

[3] D. Liberzon, *Switching in Systems and Control*. Birkhäuser, 2003.

[4] A. P. Aguiar and J. P. Hespanha, "Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty," *IEEE Tr. on Automatic Control*, vol. 52, no. 8, pp. 1362–1379, 2007.

[5] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, vol. 358, no. 1431, pp. 537–547, 2003.

[6] S. Veer and I. Poulakakis, "Safe adaptive switching among dynamical movement primitives: Application to 3D limit-cycle walkers," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2019.

[7] ——, "Switched systems with multiple equilibria under disturbances: Boundedness and practical stability," *IEEE Tr. on Automatic Control*, 2019, appeared online.

[8] R. D. Gregg, A. K. Tilton, S. Candido, T. Bretl, and M. W. Spong, "Control and planning of 3-D dynamic walking with asymptotically stable gait primitives," *IEEE Tr. on Robotics*, vol. 28, no. 6, pp. 1415–1423, 2012.

[9] M. S. Motahar, S. Veer, and I. Poulakakis, "Composing limit cycles for motion planning of 3D bipedal walkers," in *Proc. of IEEE Conf. on Decision and Control*, 2016, pp. 6368–6374.

[10] S. Veer, M. S. Motahar, and I. Poulakakis, "Almost driftless navigation of 3D limit-cycle walking bipeds," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2017, pp. 5025–5030.

[11] Q. Nguyen and K. Sreenath, "Optimal robust control for bipedal robots through control lyapunov function based quadratic programs," in *Proc. of Robotics: Science and Systems*, 2015.

[12] B. Griffin and J. Grizzle, "Walking gait optimization for accommodation of unknown terrain height variations," in *American Control Conference*. Chicago: IEEE, July 2015, pp. 4810–4817.

[13] K. A. Hamed, B. G. Buss, and J. W. Grizzle, "Exponentially stabilizing continuous-time controllers for periodic orbits of hybrid systems: Application to bipedal locomotion with ground height variations," *Int. J. of Robotics Research*, vol. 35, no. 8, pp. 977–999, 2016.

[14] S. Veer and I. Poulakakis, "Robustness of periodic orbits of impulsive systems à la Poincaré," in *Proc. of the IEEE Conf. on Decision and Control*, 2019.

[15] S. Veer, M. S. Motahar, and I. Poulakakis, "On the adaptation of dynamic walking to persistent external forcing using hybrid zero dynamics control," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015, pp. 997–1003.

[16] S. Veer, Rakesh, and I. Poulakakis, "Input-to-state stability of periodic orbits of systems with impulse effects via Poincaré analysis," *IEEE Tr. on Automatic Control*, vol. 64, no. 11, pp. 4583–4598, 2019.

[17] K. A. Hamed, V. R. Kamidi, W.-L. Ma, A. Leonessa, and A. D. Ames, "Hierarchical and safe motion control for cooperative locomotion of robotic guide dogs and humans: A hybrid systems approach," *arXiv preprint arXiv:1904.03158*, 2019.

[18] P. A. Bhounsule, A. Zamani, and J. Pusey, "Switching between limit cycles in a model of running using exponentially stabilizing discrete control lyapunov function," in *Proc. of the American Control Conference*, 2018.

[19] C. O. Saglam and K. Byl, "Robust policies via meshing for metastable rough terrain walking," in *Proceedings of Robotics: Science and Systems*, 2014.

[20] Q. Nguyen and K. Sreenath, "Safety-critical control for dynamical bipedal walking with precise footstep placement," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 147–154, 2015.

[21] X. Da, R. Hartley, and J. W. Grizzle, "First steps toward supervised learning for underactuated bipedal robot locomotion, with outdoor experiments on the wave field," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2017, pp. 3476–3483.

[22] Q. Nguyen and K. Sreenath, "L1 adaptive control for bipedal robots with control Lyapunov function based quadratic programs," in *American Control Conf.*, 2015, pp. 862–867.

[23] A. Ames, K. Galloway, J. Grizzle, and K. Sreenath, "Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics," *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 876–891, 2014.

[24] L. B. Freidovich, U. Mettin, A. S. Shiriaev, and M. W. Spong, "A passive 2-DOF walker: Hunting for gaits using virtual holonomic constraints," *IEEE Tr. on Robotics*, vol. 25, no. 5, pp. 1202–1208, 2009.

[25] R. D. Gregg and M. W. Spong, "Reduction-based control of three-dimensional bipedal walking robots," *Int. J. of Robotics Research*, vol. 29, no. 6, pp. 680–702, 2009.

[26] A. D. Ames, "Human-inspired control of bipedal walking robots," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1115–1130, 2014.

[27] S. Veer, "Composing motion primitives under disturbances: A switched systems approach," Ph.D. dissertation, University of Delaware, 2018.

[28] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback Control of Dynamic Bipedal Robot Locomotion*. Boca Raton, FL: CRC Press, 2007.

[29] M. S. Motahar, S. Veer, J. Huang, and I. Poulakakis, "Integrating dynamic walking and arm impedance control for cooperative transportation," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015, pp. 1004–1010.

[30] J. P. Hespanha and A. S. Morse, "Stability of switched systems with average dwell-time," in *Proc. of IEEE Conf. on Decision and Control*, vol. 3, 1999, pp. 2655–2660.

[31] L. Bentley, J., "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.