

# Optimal Control of Variable Stiffness Policies: Dealing with Switching Dynamics and Model Mismatch

Andreea Radulescu, Jun Nakanishi, David J. Braun  
and Sethu Vijayakumar

**Abstract** Controlling complex robotic platforms is a challenging task, especially in designs with high levels of kinematic redundancy. Novel variable stiffness actuators (VSAs) have recently demonstrated the possibility of achieving energetically more efficient and safer behaviour by allowing the ability to simultaneously modulate the output torque and stiffness while adding further levels of actuation redundancy. An optimal control approach has been demonstrated as an effective method for such a complex actuation mechanism in order to devise a control strategy that simultaneously provides optimal control commands and time-varying stiffness profiles. However, traditional optimal control formulations have typically focused on optimisation of the tasks over a predetermined time horizon with smooth, continuous plant dynamics. In this chapter, we address the optimal control problem of robotic systems with VSAs for the challenging domain of switching dynamics and discontinuous state transition arising from interactions with an environment. First, we present a systematic methodology to simultaneously optimise control commands, time-varying stiffness profiles as well as the optimal switching instances and total movement duration based on a time-based switching hybrid dynamics formulation. We demonstrate the effectiveness of our approach on the control of a brachiating robot with a VSA considering multi-phase swing-up and locomotion tasks as an illustrative application of our proposed method in order to exploit the benefits of the VSA and intrinsic dynamics of the system. Then, to address the issue of model

---

A. Radulescu · J. Nakanishi · D.J. Braun · S. Vijayakumar (✉)  
School of Informatics, University of Edinburgh, Edinburgh, UK  
e-mail: sethu.vijayakumar@ed.ac.uk

A. Radulescu  
Department of Advanced Robotics, Istituto Italiano di Tecnologia, Genova, Italy  
e-mail: andreea.radulescu@iit.it

J. Nakanishi  
Department of Micro-Nano Systems Engineering, Nagoya University, Nagoya, Japan  
e-mail: nakanishi@mein.nagoya-u.ac.jp

D.J. Braun  
Engineering Product Development, Singapore University of Technology and Design,  
Tampines, Singapore  
e-mail: david\_braun@sutd.edu.sg

discrepancies in model-based optimal control, we extend the proposed framework by incorporating an adaptive learning algorithm. This performs continuous data-driven adjustments to the dynamics model while re-planning optimal policies that reflect this adaptation. We show that this augmented approach is able to handle a range of model discrepancies in both simulations and hardware experiments.

## 1 Introduction

Modern robotic systems are used in various fields and operate in environments highly dangerous to humans (e.g., space and deep sea exploration, search and rescue missions). Controlling these robotic platforms is a challenging task due to the design complexity and the discontinuity in the dynamics, e.g., introduced by mechanical contact with the environment. Inspired by the efficiency of biological systems in locomotion and manipulation tasks, the robotics community has recently developed a new generation of actuators equipped with an additional mechanically adjustable compliant mechanism [12, 24, 42]. These variable stiffness actuators (VSAs) can provide simultaneous modulation of stiffness and output torque with the purpose of achieving dynamic and flexible robotic movements. However, this adds further levels of actuation redundancy, making planning and control of such systems even more complicated, especially in the case of underactuated systems.

Several studies of stiffness modulation in the context of domains with contacts showed that VSAs provide a significant improvement in *energy efficiency* due to their energy storage capabilities and ability to modulate the system's dynamics [22, 40]. The use of stiffness modulation in scenarios involving interaction with the environment has been shown to provide several *safety* benefits [11, 38]. Other advantages of variable stiffness capabilities have been observed in terms of *robustness* and *adaptability*. These are often required by tasks involving unpredictable changes in the environment and noise [9, 47].

In this chapter, we first introduce a systematic methodology for movement optimisation with multiple phases and switching dynamics in robotic systems with VSAs arising from contacts and impacts with the environment [21]. By modelling such tasks as a hybrid dynamical system with time-based switching, our proposed method simultaneously optimises control commands, time-varying stiffness profiles and temporal aspect of the movement such as switching instances and total movement duration to exploit the benefits of the VSA and intrinsic dynamics of the system. We present numerical simulations and hardware experiments on a brachiating robot with a VSA to demonstrate the effectiveness of our proposed framework in achieving a highly challenging task on an underactuated system.

Then, we present an augmented method to improve the robustness of the proposed framework with respect to model uncertainty by incorporating an adaptive learning algorithm [29]. The performance of model-based control is dependent on the accuracy of the dynamics models, which are traditionally obtained by model-based parameter identification procedures. However, certain elements cannot be fully represented by

simple analytical models (e.g., complex friction of the joints or dynamics resulting from cable movement [35]), while changes in the behaviour of the system can occur due to wear and tear, or due to the use of a tool [36]. Our proposed adaptive learning method performs continuous data-driven adjustments to the dynamics model while re-planning optimal policies that reflect this adaptation. We build on prior efforts to employ adaptive dynamics learning in improving the performance of robot control [15, 19, 23]. We present results showing that our augmented approach is able to handle a range of model discrepancies in both simulations and hardware experiments on a brachiating robot platform with a VSA.

## 2 Spatio-Temporal Optimisation of Multi-phase Movements in Domains with Contacts

Traditional optimal control approaches have focused on the formulation over a pre-determined time horizon with smooth, continuous plant dynamics. In this section, we present our framework of optimal control problems for tasks with multiple phase movements including switching dynamics and discrete state transition and its application to the control of robotic systems with VSAs [21].

Dynamics with intermittent contacts and impacts such as locomotion and juggling are often modelled as hybrid dynamical systems [2, 6, 14]. From a control theoretic perspective, a significant effort has been made to address optimal control problems of various classes of hybrid systems [25, 34, 44]. However, only a few robotic applications of optimal control with hybrid dynamics formulation can be found on movement of optimisation over multiple phases [7, 14]. Instead of employing hybrid dynamics modelling formulation, different optimisation approaches to dealing with multiple contact events have been proposed such as model predictive control with smooth approximation of contact forces [37], a direct multiple-shooting based method [18], direct trajectory optimisation methods [26, 43], and a further extension of the direct collocation method with linear quadratic regulator and quadratic programming [27].

In this section, we present an approach to the hybrid optimal control problem proposed in [21] with an extension to the iterative linear quadratic regulator (iLQR<sup>1</sup>) algorithm [13] and generalisation of the time-based switched LQ control with state jumps [44]. We also incorporate temporal optimisation in order to simultaneously optimise control commands and temporal parameters (e.g., movement duration) [30]. iLQR/G is a practically effective method for iteratively solving optimal control problems and has been used in our previous work, e.g., [3, 4, 16]. Time-based switching approximation in hybrid dynamics is motivated due to the difficulties associated with the state-based switching formulation in dealing with the need of imposing constraints and finding the time for switching [5, 46]. Discussions on the benefits and

---

<sup>1</sup>iLQG is the stochastic extension to iLQR [13] and in the sequel, we may refer to these two interchangeably.

practical difficulties of alternative optimal control approaches such as indirect methods (e.g., multiple-shooting methods), direct methods, and successive approximation methods (e.g., iLQR/G and differential dynamic programming) can be found in [4]. Numerical simulations and hardware experiments on a brachiating robot driven by a VSA will be presented to demonstrate the effectiveness of the proposed approach.

## ***2.1 Outline of Multi-phase Spatio-Temporal Hybrid Optimal Control Approach***

We address spatio-temporal stiffness optimisation problems for tasks that consist of multiple phases of movements including switching dynamics and discrete state transitions (arising from interaction with the environment) in order to exploit the benefits of VSAs. In addition to optimising control commands and stiffness, we develop a systematic methodology to simultaneously optimise the temporal aspect of the movement (e.g., movement duration). Our proposed formulation also provides an optimal feedback control law while many trajectory optimisation algorithms typically compute only optimal feedforward controls.

The main ingredients of our proposed optimal control framework are as follows:

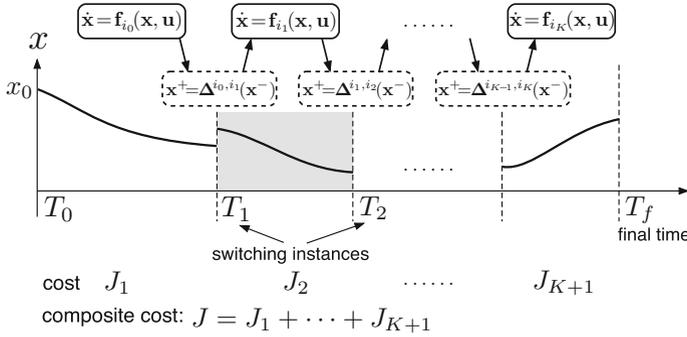
1. use of nonlinear time-based switching dynamics with continuous control input to model the dynamics of multi-phase movements;
2. use of nonlinear discrete state transition to model contacts and impacts;
3. use of realistic plant dynamics with a VSA model;
4. introduction of a composite cost function to describe task objectives with multi-phase movements;
5. simultaneous optimisation of joint torque and stiffness profiles across multiple phases;
6. optimisation of switching instances and total movement duration.

As presented below, we formulate this problem as time-based switching hybrid optimal control with temporal optimisation.

## ***2.2 Problem Formulation***

### ***2.2.1 Time-Based Switching Hybrid Dynamics***

In order to represent multi-phase movements having interactions with an environment, we consider the following time-based switching hybrid dynamics formulation composed of multiple sets of continuous dynamics (1) and discrete state transition (2) as in [8, 45]:



**Fig. 1** A hybrid system with time-based switching dynamics and discrete state transition with a known sequence. The goal is finding an optimal control command  $\mathbf{u}$ , switching instances  $T_i$  and final time  $T_f$  which minimises the composite cost  $J$

$$\dot{\mathbf{x}} = \mathbf{f}_{i_j}(\mathbf{x}, \mathbf{u}), \quad T_j \leq t < T_{j+1} \text{ for } j = 0, \dots, K \tag{1}$$

$$\mathbf{x}(T_j^+) = \Delta^{i_{j-1}, i_j}(\mathbf{x}(T_j^-)) \text{ for } j = 1, \dots, K \tag{2}$$

where  $\mathbf{f}_{i_j} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  is the subsystem  $i_j$ ,  $\mathbf{x} \in \mathbb{R}^n$  is a state vector and  $\mathbf{u} \in \mathbb{R}^m$  is a control input vector for subsystems. At the moment of dynamics switching from  $\mathbf{f}_{i_{j-1}}$  to  $\mathbf{f}_{i_j}$ , we assume an instantaneous discrete (discontinuous) state transition according to the impact map in (2), where  $\mathbf{x}(T_j^+)$  and  $\mathbf{x}(T_j^-)$  denote the post- and pre-transition states, respectively (see Fig. 1).

### 2.2.2 Robot Dynamics with Variable Stiffness Actuation for Multi-phase Movement

Given the general form of the plant dynamics of our concern in a hybrid dynamics representation introduced in Sect. 2.2.1, we consider the following multiple set of robot dynamics with VSAs to describe multi-phase movements:

$$\mathbf{M}_i(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}_i(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}_i(\mathbf{q}) + \mathbf{D}_i\dot{\mathbf{q}} = \boldsymbol{\tau}_i(\mathbf{q}, \mathbf{q}_m) \tag{3}$$

where  $i$  denotes the  $i$ -th subsystem corresponding to its associated phase of the movement,  $\mathbf{q} \in \mathbb{R}^n$  is the joint angle vector,  $\mathbf{q}_m \in \mathbb{R}^m$  is the motor position vector of the VSA,  $\mathbf{M}_i \in \mathbb{R}^{n \times n}$  is the inertia matrix,  $\mathbf{C}_i \in \mathbb{R}^n$  is the Coriolis term,  $\mathbf{g}_i \in \mathbb{R}^n$  is the gravity vector,  $\mathbf{D}_i \in \mathbb{R}^{n \times n}$  is the viscous damping matrix, and  $\boldsymbol{\tau}_i \in \mathbb{R}^n$  is the joint torque vector from the VSA given in the form:

$$\boldsymbol{\tau}_i(\mathbf{q}, \mathbf{q}_m) = \mathbf{N}_i^T(\mathbf{q}, \mathbf{q}_m)\mathbf{F}_i(\mathbf{q}, \mathbf{q}_m) \tag{4}$$

where  $\mathbf{N}_i \in \mathbb{R}^{p \times n}$  ( $p \geq n$ ) is the moment arm matrix and  $\mathbf{F}_i \in \mathbb{R}^p$  is the forces generated by the elastic elements. The joint stiffness is defined by

$$\mathbf{K}_i(\mathbf{q}, \mathbf{q}_m) = -\frac{\partial \tau_i(\mathbf{q}, \mathbf{q}_m)}{\partial \mathbf{q}}. \quad (5)$$

We model the servo motor dynamics in the VSA as a critically damped second order dynamical system:

$$\ddot{\mathbf{q}}_m + 2\alpha_i \dot{\mathbf{q}}_m + \alpha_i^2 \mathbf{q}_m = \alpha_i^2 \mathbf{u} \quad (6)$$

where  $\alpha_i$  determines the bandwidth of the servo motors<sup>2</sup> and  $\mathbf{u} \in \mathbb{R}^m$  is the motor position command [3]. We assume that the range of the control command  $\mathbf{u}$  is limited as  $\mathbf{u}_{min} \leq \mathbf{u} \leq \mathbf{u}_{max}$ .

In order to formulate an optimal control problem, we consider the following state space representation of the combined plant dynamics composed of the rigid body dynamics (3) and the servo motor dynamics (6):

$$\dot{\mathbf{x}} = \mathbf{f}_i(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \mathbf{x}_2 \\ \mathbf{M}_i^{-1}(\mathbf{x}_1) (-\mathbf{C}_i(\mathbf{x}_1, \mathbf{x}_2)\mathbf{x}_2 - \mathbf{g}_i(\mathbf{x}_1) - \mathbf{D}_i\mathbf{x}_2 + \tau_i(\mathbf{x}_1, \mathbf{x}_3)) \\ \mathbf{x}_4 \\ -\alpha_i^2 \mathbf{x}_3 - 2\alpha_i \mathbf{x}_4 + \alpha_i^2 \mathbf{u} \end{bmatrix} \quad (7)$$

where  $\mathbf{x} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \mathbf{x}_3^T, \mathbf{x}_4^T]^T = [\mathbf{q}^T, \dot{\mathbf{q}}^T, \mathbf{q}_m^T, \dot{\mathbf{q}}_m^T]^T \in \mathbb{R}^{2(n+m)}$  is the augmented state vector consisting of the robot state and the servo motor state.

### 2.2.3 Composite Cost Function for Multi-phase Movement Optimisation

For the given hybrid dynamics (1) and (2), in order to describe the full movement with multiple phases, we consider the following composite cost function

$$J = \phi(\mathbf{x}(T_f)) + \sum_{j=1}^K \psi^j(\mathbf{x}(T_j^-)) + \int_{T_0}^{T_f} h(\mathbf{x}, \mathbf{u}) dt \quad (8)$$

where  $\phi(\mathbf{x}(T_f))$  is the terminal cost,  $\psi^j(\mathbf{x}(T_j^-))$  is the via-point cost at the  $j$ -th switching instance and  $h(\mathbf{x}, \mathbf{u})$  is the running cost. When optimising multi-phase movements, it is possible to optimise each phase in a sequential manner. However, the total cost of such a sequential optimisation could result in a suboptimal solution [30].

For the given plant dynamics (1) and state transition (2), the optimisation problem we consider is to a) find an optimal feedback control law  $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$  which minimises

---

<sup>2</sup> $\alpha = \text{diag}(a_1, \dots, a_m)$  and  $\alpha^2 = \text{diag}(a_1^2, \dots, a_m^2)$  for notational convenience.

the composite cost (8) and b) simultaneously optimise switching instances  $T_1, \dots, T_k$  and the final time  $T_f$ . Note that in our formulation, we denote the final time separately from switching instances for notational consistency with the case of single phase optimisation. However, the final time can be absorbed as a part of switching instances, e.g.,  $T_f = T_{K+1}$ .

## 2.2.4 Spatio-Temporal Multi-Phase Optimisation Algorithm

In this section, we present an overview of our framework for spatio-temporal optimisation for multi-phase movements. First, the iLQR method [13] is extended in order to incorporate timed switching dynamics with discrete and discontinuous state transitions. Then, we present a temporal optimisation algorithm to obtain the optimal switching instances and the total movement duration. For more detailed description, we refer the interested readers to [21].

In brief, the iLQR method solves an optimal control problem of the locally linear quadratic approximation of the nonlinear dynamics and the cost function around a nominal trajectory  $\bar{\mathbf{x}}$  and control sequence  $\bar{\mathbf{u}}$  in discrete time, and iteratively improves the solutions.

In order to incorporate switching dynamics and discrete state transition with a given switching sequence, the hybrid dynamics (1) and (2) are linearised in discrete time around the nominal trajectory and control sequence as

$$\delta \mathbf{x}_{k+1} = \mathbf{A}_k \delta \mathbf{x}_k + \mathbf{B}_k \delta \mathbf{u}_k \quad (9)$$

$$\delta \mathbf{x}_{k_j}^+ = \Gamma_{k_j} \delta \mathbf{x}_{k_j}^- \quad (10)$$

$$\mathbf{A}_k = \mathbf{I} + \Delta t_j \left. \frac{\partial \mathbf{f}_j}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_k}, \quad \mathbf{B}_k = \Delta t_j \left. \frac{\partial \mathbf{f}_j}{\partial \mathbf{u}} \right|_{\mathbf{u}=\mathbf{u}_k} \quad (11)$$

$$\Gamma_{k_j} = \left. \frac{\partial \Delta^{i_{j-1}, i_j}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_{k_j}^-} \quad (12)$$

where  $\delta \mathbf{x}_k = \mathbf{x}_k - \bar{\mathbf{x}}_k$ ,  $\delta \mathbf{u}_k = \mathbf{u}_k - \bar{\mathbf{u}}_k$ ,  $k$  is the discrete time step,  $\Delta t_j$  is the sampling time being optimised for the time interval  $T_j \leq t < T_{j+1}$ , and  $k_j$  is the  $j$ -th switching instance in the discretised time step. The composite cost function and the optimal cost-to-go function are locally approximated and the local optimal control problem is solved via modified Riccati-like equations as described in detail in [21]. Once we have a locally optimal control command  $\delta \mathbf{u}$ , the nominal control sequence is updated as  $\bar{\mathbf{u}} \leftarrow \bar{\mathbf{u}} + \delta \mathbf{u}$ . Then, the new nominal trajectory  $\bar{\mathbf{x}}$  and the cost  $J$  are computed by running the obtained control  $\bar{\mathbf{u}}$  on the system dynamics, and the above process is iterated until convergence (no further improvement in the cost within certain threshold).

In order to optimise the switching instance and the total movement duration, following our previous work in [30], we introduce a scaling parameter and sampling time for each duration of between switching as

$$\Delta t'_j = \frac{1}{\beta_j} \Delta t_j \quad \text{for } T_j \leq t < T_{j+1} \quad \text{where } j = 0, \dots, K. \quad (13)$$

By optimising the vector of temporal scaling factors  $\beta = [\beta_0, \dots, \beta_K]^T$  via gradient descent, we can obtain each switching instance  $T_{j+1} = (k_{j+1} - k_j)\Delta t'_j + T_j$  and the total movement duration  $T_f = \sum_{j=0}^K (k_{j+1} - k_j)\Delta t'_j + T_0$ , where  $k_0 = 1$  and  $k_{K+1} = N$ .

In the complete optimisation, computation of the optimal feedback control law and update of the temporal scaling parameters are iteratively performed in an alternate manner until convergence. As a result, we obtain the optimal feedback control law

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_{opt}(t) + \mathbf{L}_{opt}(t)(\mathbf{x}(t) - \mathbf{x}_{opt}(t)) \quad (14)$$

and the optimal switching instances  $T_1, \dots, T_K$  and the final time  $T_f$ , where  $\mathbf{u}_{opt}$  is the feedforward optimal control sequence,  $\mathbf{x}_{opt}$  is the optimal trajectory, and  $\mathbf{L}_{opt}$  is the optimal feedback gain matrix.

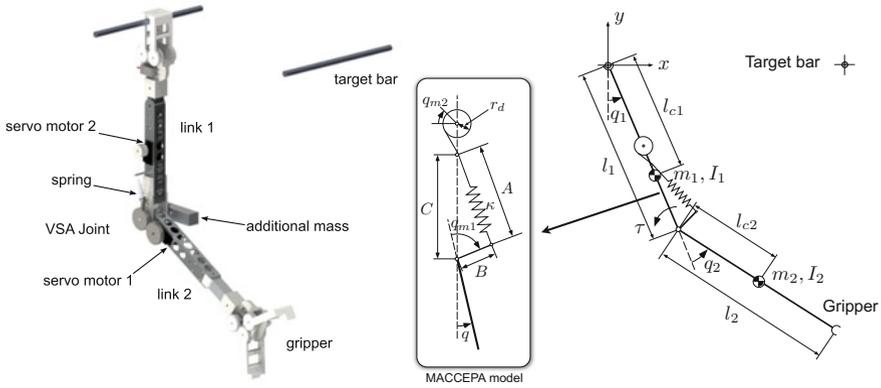
### 2.3 Brachiating Robot Dynamics with VSA

The dynamics of a two-link brachiating robot with a VSA shown in Fig. 2 take the standard form of rigid body dynamics (3) where only the second joint has actuation (underactuation):

$$\mathbf{M}_i(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}_i(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}_i(\mathbf{q}) + \mathbf{D}_i\dot{\mathbf{q}} = \begin{bmatrix} 0 \\ \tau(\mathbf{q}, \mathbf{q}_m) \end{bmatrix} \quad (15)$$

where  $\mathbf{q} = [q_1, q_2]^T$  is the joint angle vector. The same definitions for the elements in the rigid body dynamics are used as in (3). The index  $i$  in (15) is introduced to specify the configuration of the robot to indicate which hand is holding the bar. Since we assume that the robot has an asymmetric structure in the dynamics, we have two sets of subsystems denoted by the subscripts  $i = 1$  (hand of link 1 is holding) and  $i = 2$  (hand of link 2 is holding). In the multi-phase brachiation, the effective model switches between  $i = 1$  and  $i = 2$  interchangeably at the switching instance when the robot grasps the bar.

We use MACCEPA (Fig. 2) as our VSA implementation of choice [39], which has the desirable property that the joint can be passively compliant. This allows free swinging with a large range of movement by relaxing the spring, which is highly suitable for the brachiation task we consider. MACCEPA is equipped with two



**Fig. 2** Two-link brachiating robot model with the MACCEPA joint with the inertial and geometric parameters. The parameters of the robot are given in Table 1

position controlled servo motors,  $\mathbf{q}_m = [q_{m1}, q_{m2}]^T$  which control the equilibrium position and the spring pre-tension.<sup>3</sup>

The joint torque for the MACCEPA model is given by:

$$\tau = \underbrace{\frac{BC \sin(q_{m1} - q)}{A}}_{\text{moment arm in (4)}} F \tag{16}$$

where  $A = \sqrt{B^2 + C^2 - 2BC \cos(q_{m1} - q)}$ ,  $q$  is the joint angle.<sup>4</sup>  $F$  is the spring tension

$$F = \kappa_s(l - l_0) \tag{17}$$

where  $\kappa_s$  the spring constant,  $l = A + r_d q_{m2}$  is the current spring length,  $l_0 = C - B$  is the spring length at rest and  $r_d$  is the drum radius (see Fig. 2). The joint stiffness can be computed as  $k = -\frac{\partial \tau}{\partial q}$ . Note that the torque and stiffness relationships in MACCEPA are dependent on the current joint angle and two servo motor angles in a complicated manner and its control is not straightforward.

To formulate the multi-phase movement optimisation in brachiation, we use the state space representation in (7). At the transition at handhold, an affine discrete state transition  $\mathbf{x}^+ = \Delta(\mathbf{x}^-) = \Gamma \mathbf{x}^- + \gamma$  is introduced to shift the coordinate system for the next handhold and reset the joint velocities of the robot to zero, which is defined as:

<sup>3</sup>We include position controlled servo motor dynamics as defined in (6). For the bandwidth parameters for the motors we use  $\alpha = \text{diag}(20, 25)$ . The range of the commands of the servo motors are limited as  $u_1 \in [-\pi/2, \pi/2]$  and  $u_2 \in [0, \pi/2]$ .

<sup>4</sup>In the brachiating robot model in Fig. 2,  $q = q_2$ .

**Table 1** Model parameters of the two-link brachiating robot and the VSA. The index  $i$  in this table denotes the link number in Fig. 2. The final column (numbers in *italic*) in the robot parameters table shows the change of parameters of the first link of the system under the changed mass distribution described in Sect. 3.2

Robot parameters		i=1	i=2	<i>i=1</i>	MACCEPA parameters		value
Mass	$m_i$ (kg)	1.390	0.527	<i>1.240</i>	Spring constant	$\kappa_s$ (N/m)	771
Moment of inertia	$I_i$ (kgm <sup>2</sup> )	0.0297	0.0104	<i>0.0278</i>	Lever length	$B$ (m)	0.03
Link length	$l_i$ (m)	0.46	0.46	<i>0.46</i>	Pin displacement	$C$ (m)	0.125
COM location	$l_{ci}$ (m)	0.362	0.233	<i>0.350</i>	Drum radius	$r_d$ (m)	0.01
Viscous friction	$d_i$ (Nm/s)	0.03	0.035	<i>0.03</i>			

$$\Gamma = \text{diag}(\Gamma_1, \dots, \Gamma_4), \quad (18)$$

where:

$$\Gamma_1 = \begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix}, \Gamma_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \Gamma_3 = \Gamma_4 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad (19)$$

and  $\gamma = [-\pi, 0, \dots, 0]^T$ . Note that in this example, we have  $\Delta = \Delta^{1,2} = \Delta^{2,1}$ .

## 2.4 Exploitation of Passive Dynamics With Spatio-Temporal Optimisation of Stiffness

In this section, we explore the benefits of simultaneous stiffness and temporal optimisation for tasks exploiting the intrinsic dynamics of the system. Brachiation is an example of a highly dynamic manoeuvre requiring the use of passive dynamics for successful task execution [10, 20, 31, 32].

### 2.4.1 Optimisation of a Single Phase Movement in Brachiation Task

In this section, we consider the brachiation task of swing locomotion from handhold to handhold on a ladder. A natural and desirable strategy for a swing movement in brachiation would be to make good use of gravity by making the joints passive and compliant. For a system with VSAs, our idea in exploiting passive dynamics is to frame the control problem in finding an appropriate (preferably small) stiffness profile to modulate the system dynamics only when necessary and compute the virtual equilibrium trajectory to fulfil the specified task requirement.

To implement this idea of passive control strategy, we consider the following cost function to encode the task:

$$J = (\mathbf{y}(T) - \mathbf{y}^*)^T \mathbf{Q}_T (\mathbf{y}(T) - \mathbf{y}^*) + \int_0^T (\mathbf{u}^T \mathbf{R}_1 \mathbf{u} + R_2 F^2) dt \quad (20)$$

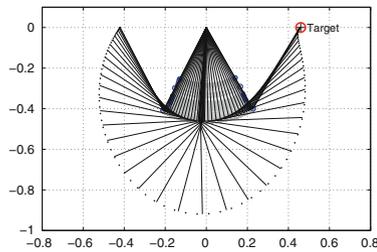
where  $\mathbf{y} = [\mathbf{r}, \dot{\mathbf{r}}]^T \in \mathbb{R}^4$  are the position and the velocity of the gripper in the Cartesian coordinates,  $\mathbf{y}^*$  contains the target values when reaching the target  $\mathbf{y}^* = [\mathbf{r}^*, \mathbf{0}]^T$  and  $F$  is the spring tension in the VSA given in (17).  $\mathbf{Q}_T$  is a positive semi-definite matrix,  $\mathbf{R}_1$  is a positive definite matrix and  $R_2$  is a positive scalar. This objective function is designed in order to reach the target located at  $\mathbf{r}^*$  at the time  $T$ , while minimising the spring tension  $F$  in the VSA. The term  $\mathbf{u}^T \mathbf{R}_1 \mathbf{u}$  is added for regularisation with a small choice of the weights in  $\mathbf{R}_1$ , which is necessary in practice since iLQG requires a control cost in its formulation to compute the optimal control law.

## 2.4.2 Benefit of Temporal Optimisation

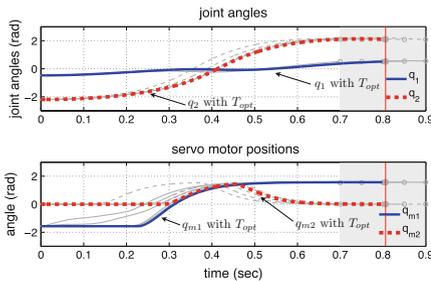
One of the issues in a conventional optimal control formulation is that the time horizon needs to be given in advance for a given task. While on fully actuated systems, control can be used to enforce a pre-specified timing, it is not possible to choose an arbitrary time horizon on underactuated systems. In a brachiation task, determination of an appropriate movement horizon is essential for successful task execution with reduced control effort.

Consider the swing locomotion task on a ladder with the intervals starting from the bar at  $d_{start} = 0.42$  m to the target located at  $d_{target} = 0.46$  m. We optimise both the control command  $\mathbf{u}$  and the movement duration  $T$ . We use  $\mathbf{Q}_T = \text{diag}(10000, 10000, 10, 10)$ ,  $\mathbf{R}_1 = \text{diag}(0.0001, 0.0001)$  and  $R_2 = 0.01$  for the cost function in (20). The optimised movement duration was  $T = 0.806$  s.

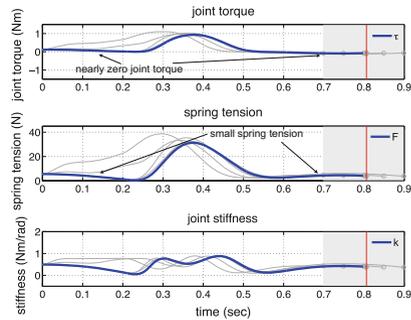
Figure 3 shows the simulation result of (a) the optimised robot movement, (b) joint trajectories and servo motor positions, and (c) joint torque, spring tension and joint stiffness. In the plots, trajectories of the fixed time horizon ranging  $T \in [0.7, 0.75, \dots, 0.9]$  s are overlaid for comparison in addition to the case of the optimal movement duration  $T = 0.806$  s. In the movement with temporal optimisation, the spring tension and the joint stiffness are kept small at the beginning and end of the movement resulting in nearly zero joint torque. This allows the joint to swing passively. The joint torque is exerted only during the middle of the swing by increasing the spring tension as necessary. In contrast, with non-optimal time horizon, larger joint torque and spring tension as well as higher joint stiffness can be observed resulting in the requirement of increased control effort. This result suggests that the natural plant dynamics are fully exploited for the desirable task execution with simultaneous stiffness and temporal optimisation.



(a) Movement of the robot (simulation) with optimal variable stiffness control (optimised duration  $T=0.806$  s).



(b) Joint trajectories and servo motor positions



(c) Joint torque, spring tension and joint stiffness

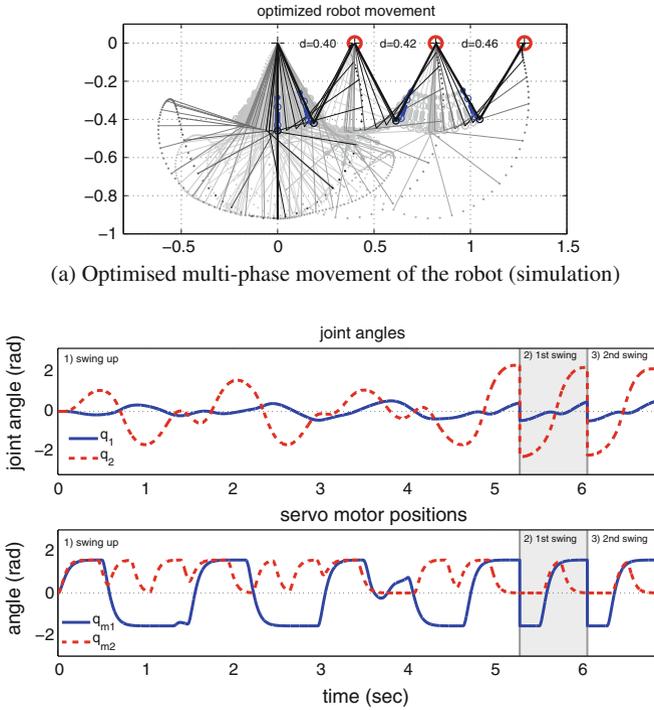
**Fig. 3** Simulation result of the single phase brachiation task with temporal optimisation. In **b** and **c**, *grey thin lines* show the plots for non-optimised  $T$  in the range of  $T = [0.7, 0.75, \dots, 0.9]$  s and *blue thick lines* show the plots for optimised  $T = 0.806$  s. With temporal optimisation, at the beginning and the end of the movement, joint torque, spring tension and joint stiffness are kept small allowing the joint to swing passively in comparison to the non-optimal time cases

## 2.5 Spatio-Temporal Optimisation of Multiple Swings in Robot Brachiation

To demonstrate the effectiveness of our proposed approach in multi-phase movement optimisation, we consider the following brachiation task with multiple phases of the movement: The robot initially swings up from the suspended posture to the target at  $d_1 = 0.40$  m and subsequently moves to the target located at  $d_2 = 0.42$  m and  $d_3 = 0.46$  m. The composite cost function to encode this task is designed as:

$$\begin{aligned}
 J = & (\mathbf{y}(T_f) - \mathbf{y}_f^*)^T \mathbf{Q}_{T_f} (\mathbf{y}(T_f) - \mathbf{y}_f^*) + \sum_{j=1}^K (\mathbf{y}(T_j^-) - \mathbf{y}_j^*)^T \mathbf{Q}_{T_j} (\mathbf{y}(T_j^-) - \mathbf{y}_j^*) \\
 & + \int_0^{T_f} (\mathbf{u}^T \mathbf{R}_1 \mathbf{u} + R_2 F^2) dt + w_T T_1
 \end{aligned} \quad (21)$$

where  $K = 3$  is the number of phases,  $\mathbf{y} = [\mathbf{r}, \dot{\mathbf{r}}]^T \in \mathbb{R}^4$  are the position and the velocity of the gripper in the Cartesian coordinates measured from the origin at



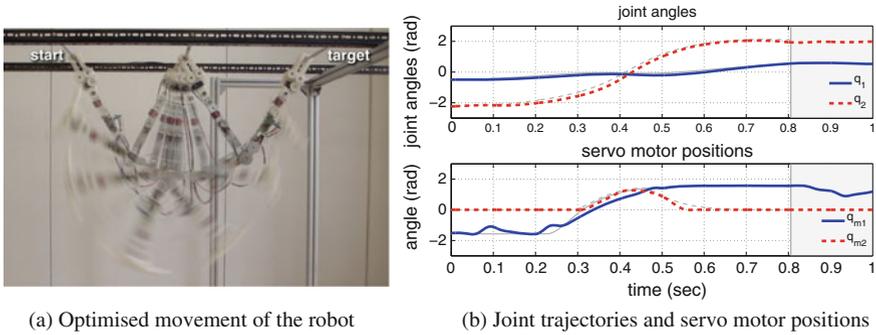
(b) Joint trajectories and servo motor positions with temporal optimisation.

**Fig. 4** Simulation result of the multi-phase brachiation task with temporal optimisation

current handhold,  $\mathbf{y}^*$  is the target values when reaching the target  $\mathbf{y}^* = [\mathbf{r}^*, \mathbf{0}]^T$  and  $F$  is the spring tension in the VSA. Note that this cost function includes the additional time cost  $w_T T_1$  for the swing up manoeuvre in order to regulate the duration of the swing up movement. We use  $\mathbf{Q}_{T_f} = \mathbf{Q}_{T_j} = \text{diag}(10000, 10000, 10, 10)$ ,  $\mathbf{R}_1 = \text{diag}(0.0001, 0.0001)$  and  $R_2 = 0.01$  and  $w_T = 1$ . In addition, we impose constraints on the range of the angle of the second joint during the course of the swing up manoeuvre as  $q_{2_{min}} \leq q_2 \leq q_{2_{max}}$ , where  $[q_{2_{min}}, q_{2_{max}}] = [-1.745, 1.745]$  rad by adding a penalty term to the cost (21). This is empirically introduced and adjusted considering the physical joint limit of the hardware platform used in the experiments.

Figure 4a shows the sequence of the optimised multi-phase movement of the robot using the proposed algorithm including temporal optimisation in numerical simulations. The optimised switching instances and the total movement duration are  $T_1 = 5.259$  s,  $T_2 = 6.033$  s and  $T_f = 6.835$  s, respectively, and the total cost is  $J = 37.815$ . Figure 4b shows the optimised joint trajectories and servo motor positions.

To illustrate the benefit of the proposed multi-phase formulation, we performed movement optimisation with a pre-specified nominal (fixed, non-optimal)



**Fig. 5** Experimental result of the single phase locomotion task on the brachiating robot hardware. In **b**, red and blue think lines show the experimental data, and grey thin lines show the corresponding simulation result with the optimised planned movement duration  $T = 0.806$  s presented in Sect. 2.4.2

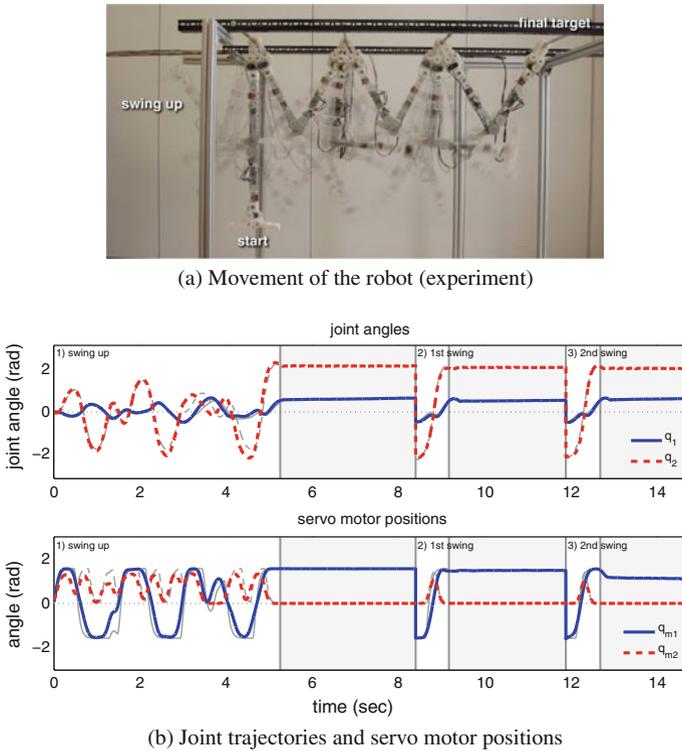
time horizon with  $T_1 = 5.2$  s,  $T_2 = 5.9$  s and  $T_f = 6.7$  s using the same cost parameters both in sequential and multi-phase optimisation. With sequential individual movement optimisation, large overshoot was observed at the end of the final phase movement (distance from the target at  $t = T_f$  was 0.0697 m) incurring a significantly large total cost of  $J = 101.053$ . In contrast, for the same pre-specified time horizon, by employing multi-phase movement optimisation, it was possible to find a feasible solution to reach the target bars. The error at the final swing at  $t = T_f$  was 0.0020 m, which was significant improvement compared to the case of individual optimisation. The largest error observed in this sequence was 0.0109 m at the end of the first swing up phase. In this case, the total cost was  $J = 50.228$ . These results demonstrate the benefit of the multi-phase movement optimisation in finding optimal control commands and temporal aspect of the movement using the proposed method resulting in lower cost.

## 2.6 Evaluation on Hardware Platform

This section presents experimental implementation of our proposed algorithm on a two-link brachiating robot hardware developed in our laboratory [21]. The robot is equipped with a MACCEPA variable stiffness actuator and the parameters of the robot are given in Table 1.

Figure 5 shows the experimental result of a swing locomotion corresponding to the simulation in Sect. 2.4.2 with the optimal movement duration. In the experiments, we only use the open-loop optimal control command to the servo motors without state feedback as in [3].

Figure 6 shows the experimental result of multi-phase movements consisting of swing-up followed by two additional swings, which corresponds to the simulation



**Fig. 6** **a** Experimental results of multi-phase locomotion task with the brachiating robot hardware. In **b**, *red* and *blue* lines show the actual hardware behaviour, and *grey* lines show the corresponding simulation results presented in Sect. 2.5

in Fig. 4 (Sect. 2.5). Note that at the end of each phase of the movement, switching to the next phase is manually done by confirming firm grasping of the bar in order to avoid falling off from the ladder at run-time.

The joint trajectories in the experiment closely match the planned movement in the simulation. The observed discrepancy is mainly due to the inevitable difference between the analytical nominal model and the hardware system. In the next section, we introduce an adaptive learning algorithm to improve the accuracy of the dynamics model used in optimisation.

These experimental results demonstrate the effectiveness and feasibility of the proposed framework in achieving highly dynamic tasks in compliantly actuated robots with variable stiffness capabilities under real conditions.

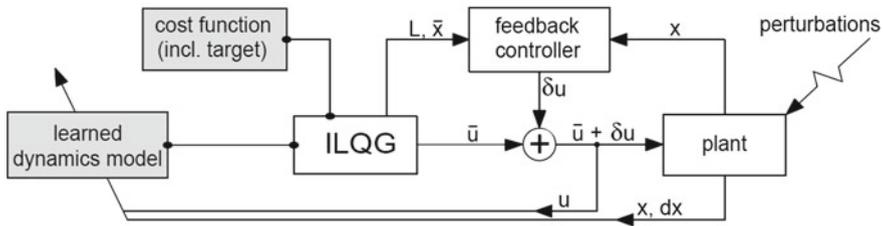


Fig. 7 The iLQG-LD learning and control scheme [16]

### 3 Optimal Control with Learned Dynamics

Classical optimal control is formulated using an analytical dynamics model, however, recent work [1, 17] has shown that combining optimal control with dynamics learning can produce an effective and principled control strategy for complex systems affected by model uncertainties.

In [17], using online (non-parametric) supervised learning methods, an adaptive internal model of the system dynamics is learned. The learned model is used afterwards to derive an optimal control law. This approach, named *iLQG with learned dynamics (iLQG-LD)*,<sup>5</sup> proved efficient in a variety of realistic scenarios including problems where the analytical dynamics model is difficult to estimate accurately or subject to changes and the system is affected by noise [16, 17]. The initial state and the cost function (which includes the desired final state) are provided to the iLQG planner alongside a preliminary model of the dynamics. An initial (locally optimal) command sequence  $\bar{\mathbf{u}}$  is generated together with the corresponding state sequence  $\bar{\mathbf{x}}$  and feedback correction gains  $\mathbf{L}$ . Applying the feedback controller scheme, at each time step the control command is corrected by  $\delta \mathbf{u} = \mathbf{L}(\mathbf{x} - \bar{\mathbf{x}})$ , where  $\mathbf{x}$  is the true state of the plant. The model of the dynamics is updated using the information provided by the applied command  $\mathbf{u} + \delta \mathbf{u}$  and observed state  $\mathbf{x}$  (Fig. 7).

This iLQG-LD methodology employs a *Locally Weighted Learning (LWL)* method, or more specifically, the *Locally Weighted Projection Regression (LWPR)*, to train a model of the dynamics in an incremental fashion. LWL algorithms are non-parametric local learning methods that proved successful in the context of (online) motor learning scenarios [1]. Incremental LWL was proposed in [33] (*Receptive Field Weighted Regression (RFWR)* method) in order to achieve fast learning and computational efficiency. RFWR works by allocating resources in a data driven fashion, allowing online adaptation to changes in the behaviour. The LWPR [41] extends the RFWR method by projecting the input information into a lower dimensional manifold along selected directions before performing the fitting. Thus, it proves effective in high dimensionality scenarios where the data lies in a lower dimensional space. Consequently, iLQG-LD proved to be a robust and efficient technique for incremental learning of nonlinear models in high dimensions [17].

<sup>5</sup>Hereafter, we use the term iLQG for the optimisation algorithm of our concern.

We incorporate the iLQG-LD scheme into our approach involving learning the dynamics of a brachiating robot with a VSA and employing it in planning for multi-phase locomotion tasks [29]. The method proved capable of adapting to changes in the system's properties and provided a better accuracy performance than the optimisation without model adaptation. Based on these results, iLQG-LD could be a strong candidate for optimal control strategy for more complex hardware systems. We demonstrate the effectiveness of our adaptive learning approach in numerical simulations (Sect. 3.2) and hardware experiments (Sect. 3.3).

### 3.1 Multi-Phase Optimisation with Adaptive Dynamics Learning

In this section, we introduce the changes by the use of the LWPR method in the context of iLQG-LD for integration within the multi-phase optimisation approach described in Sect. 2.2. We assume that initially we have a nominal analytical dynamics model that takes the form presented in (3) and (6) which has inaccuracies. We use the LWPR method to model the error between the true behaviour of the system and the initial nominal model provided. For this purpose, we replace the dynamics  $\mathbf{f}_i$  in (7) with a composite dynamics model  $\mathbf{f}_{c_i}$ :

$$\dot{\mathbf{x}} = \mathbf{f}_{c_i}(\mathbf{x}, \mathbf{u}) = \tilde{\mathbf{f}}_i(\mathbf{x}, \mathbf{u}) + \bar{\mathbf{f}}_i(\mathbf{x}, \mathbf{u}), \quad \mathbf{f}_{c_i} \in \mathbb{R}^{2(n+m)}, \quad (22)$$

where  $\tilde{\mathbf{f}}_i$  is the initial nominal model and  $\bar{\mathbf{f}}_i$  is the LWPR model to learn the discrepancy between  $\tilde{\mathbf{f}}_i$  and the behaviour of the system.<sup>6</sup>

When using the composite model of the dynamics  $\mathbf{f}_c$  introduced in (22), the linearisation of the dynamics is provided in two parts. The linearisation of  $\tilde{\mathbf{f}}$  is obtained by replacing  $\mathbf{f}$  with  $\tilde{\mathbf{f}}$  in (9) and (11). The derivatives of the learned model  $\bar{\mathbf{f}}$  are obtained analytically by differentiating the LWPR model with respect to the inputs  $\mathbf{z} = (\mathbf{x}; \mathbf{u})$  as suggested in [1]. With these modifications, the developed optimisation methodology is applied as described in Sect. 2.2 to obtain the locally optimal feedback control law.

### 3.2 Numerical Evaluations

In this section, we numerically demonstrate the effectiveness of the proposed model learning approach on a brachiating robot model with a VSA used in Sect. 2.3. In the nominal model, we introduce a mass (and implicitly mass distribution) discrepancy on one of the links (i.e., the mass of the true model is smaller by 150g located

---

<sup>6</sup>Note that the changes introduced by iLQG-LD only affect the dynamics modelling in (1), while the instantaneous state transition map in (2) remains unchanged.

at the joint on link  $i = 1$ ). The changed model parameters are shown in the right column of Table 1 (the numbers *in italic*). The introduced discrepancy affects the rigid body dynamics in the joint accelerations ( $\ddot{q}_1, \ddot{q}_2$ ). Thus, in the composite model  $\mathbf{f}_c$ , the information from the nominal model  $\tilde{\mathbf{f}}$  requires correction only in those two dimensions (i.e., the required dimensionality of the LWPR model output  $\bar{\mathbf{f}}$  is 2, the remaining dimensions can be filled with zeros).

We demonstrate the effectiveness of the proposed approach on a multi-phase swing-up and brachiation task with a VSA while incorporating *continuous, online model learning*. In the multi-phase task, we consider the same task of swing-up and multi-swing locomotion presented in Sect. 2.5. Since the system has an asymmetric configuration and the state space of the swing-up task is significantly different from that of a brachiation movement, we proceed by first learning a separate error model for each phase. This procedure contains two steps. The initial exploration phase is performed in order to pre-train the LWPR model  $\tilde{\mathbf{f}}_i$  (as an alternative to applying motor babbling), while the second phase is using iLQG-LD to refine the model in an online fashion. In our evaluations, the training data are obtained by using a simulated version of the true dynamics, which is an analytical model incorporating the discrepancy.

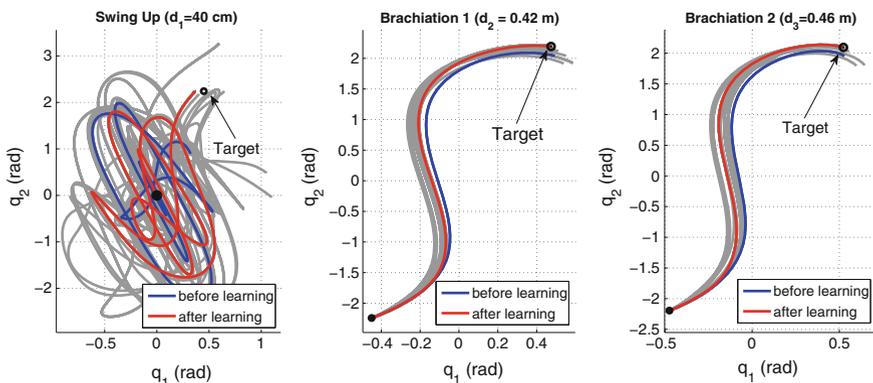
### 3.2.1 Individual Phase Learning

As presented in Sect. 2.5, using our multi-phase spatio-temporal optimisation framework with the correct dynamics model, we successfully achieved a multi-phase brachiation task with a position error of as small as 0.002 m at the target bar in numerical simulations. However, once the discrepancy is introduced to the nominal model as described in Sect. 3.2, the planned solution is no longer valid and the final position deviates from the desired target in each individual swing-up and locomotion movement as illustrated in simulations (Fig. 8, blue line). We demonstrate the effectiveness of the iLQG-LD framework in order to learn the characteristics of the system dynamics and recover the task performance.

As a measure of the model accuracy, we use the *normalised mean squared error* (nMSE) of the model prediction on the true optimal trajectory (if given access to the analytical form of the true dynamics of the system). The evolution of the nMSE at the stage of training for each movement phase is shown in Fig. 9.

In the first part (pre-training phase in Fig. 9), we generate ( $p = 7$ ) random targets around the desired  $x_T$ . A movement is planned for these targets using the nominal model  $\tilde{\mathbf{f}}$ . The obtained optimal feedback controller for the nominal model is then applied to the simulated version of the true dynamics. We repeat this procedure for a set of 10 noise added versions of the commands. The collected data are used to train the model.

This pre-training phase seeds the model with information within the region of interest prior to its use for planning. The aim is to reduce the number of iterations required for convergence in iLQG-LD. For each movement, at the end of the procedure,



**Fig. 8** Individual phase learning with model adaptation in simulation: Comparison of the final position achieved (for each individual phase) when using the initial planning (inaccurate nominal model in *blue*) and the final planning (composite model after learning in *red*). Intermediary solutions obtained at each step of the iLQG-LD run are depicted in *grey*

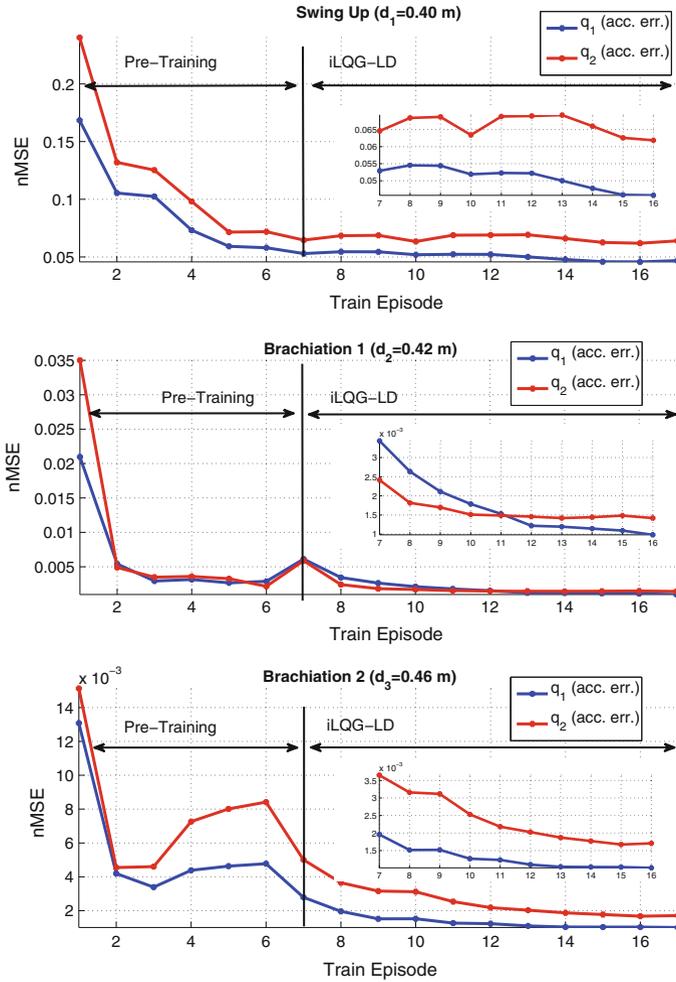
the planned trajectory matched the behaviour obtained from running the command solution on the real plant (the final nMSE has an order of magnitude of  $10^{-4}$ ).

Overall, the discrepancy is found to be small enough to reach the desired end-effector position within a tolerance of  $\epsilon_T = 0.040$  m. Figure 8 shows the effect of the learning by comparing the performance of the planning with the inaccurate nominal model and with the composite model obtained after training.

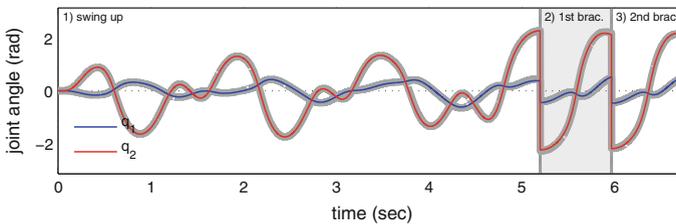
### 3.2.2 Multi-phase Performance

In order to evaluate the validity of the learned model, we optimise the multi-phase brachiation task with the composite cost function given in (21) using the obtained model from the individual phase learning procedure in Sect. 3.2.1. We use the optimal solutions obtained for each individual phase above as an initial command sequence for the multi-phase optimisation. The simulation result is shown in Fig. 10. The planner is able to successfully achieve the intermediate and final goals, while the expected behaviour provides a reliable match to the actual system’s behaviour.<sup>7</sup> The cost of multi-phase optimised solution ( $J = 35.13$ ) is significantly lower than the sum of the costs of the individual phase solutions ( $J = 44.45$ ).

<sup>7</sup>We assume that if the position at the end of each phase is within a threshold  $\epsilon_T = 0.040$  m from the desired target, the system is able to start the next phase movement from the ideal location considering the effect of the gripper on the hardware.



**Fig. 9** Individual phase learning with model adaptation in simulation: Evolution of the nMSE for each phase of the movement at each episode



**Fig. 10** Performance of the fully optimised multi-phase brachiation task using the composite learned model in simulation. *Thick grey lines*: planned movement. *Red and blue lines*: actual system movement

### 3.3 Hardware Experiments: Individual Phase Learning for a Brachiating Robot

In this section, we perform hardware experiments to evaluate the simulation results obtained in Sect. 3.2 on our two-link brachiating robot with a VSA. In the hardware experimental set-up, we only have access to sensory readings from two IMU units attached to each link, a potentiometer mounted on the main joint and the internal potentiometers of the servo motors in the VSA. The outputs of the IMU units are fairly accurate and adequate filtering provides reliable readings for estimating the positions and velocities of the robot links. However, the internal potentiometers of the servo motors suffer from significant amount of noise; with filtering, their readings can be used as an estimation of the motor position, but they are not reliable enough to derive the servo motor accelerations, which are needed for model learning. For this reason, we reduce the dimensionality of the input for the model approximation from 10 ( $[\mathbf{q}^T, \dot{\mathbf{q}}^T, \mathbf{q}_m^T, \dot{\mathbf{q}}_m^T, \mathbf{u}^T]^T$ ) to 8 ( $[\mathbf{q}^T, \dot{\mathbf{q}}^T, \mathbf{q}_m^T, \mathbf{u}^T]^T$ ).<sup>8</sup>

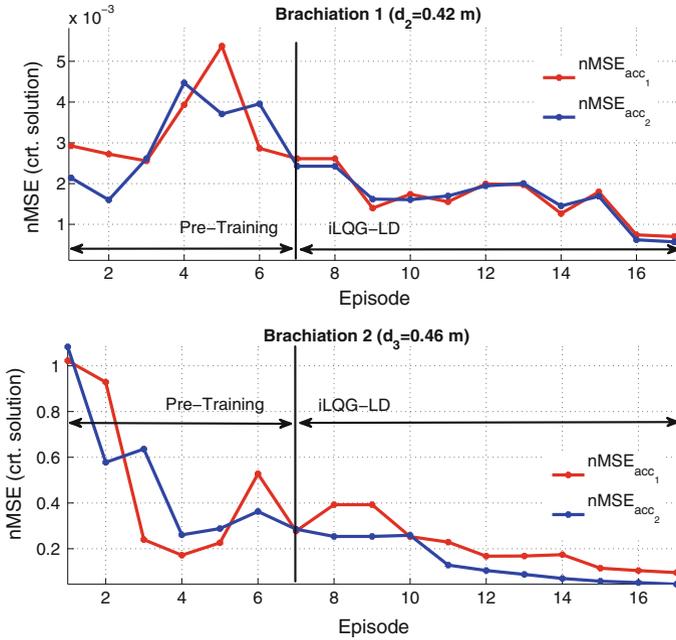
In the pre-training phase, we generate random targets around the desired  $x_T$  and plan movements for those targets using the nominal model  $\tilde{\mathbf{f}}$  as in the numerical simulations described in Sect. 3.2.1 (with 150 g of mass discrepancy introduced). We apply the obtained solution to the hardware with a set of 10 noise added versions of the commands. The collected data are used to train the model. In the second phase, we apply iLQG-LD as described in Sect. 3.2. The evolution of the nMSE at the stage of training for each phase is shown in Fig. 11.

For each phase of the movement, at the end of the procedure, the planned trajectory matched the behaviour obtained from running the command solution on the real plant (the final nMSE has an order of magnitude of  $10^{-4}$  and  $10^{-2}$ , respectively).

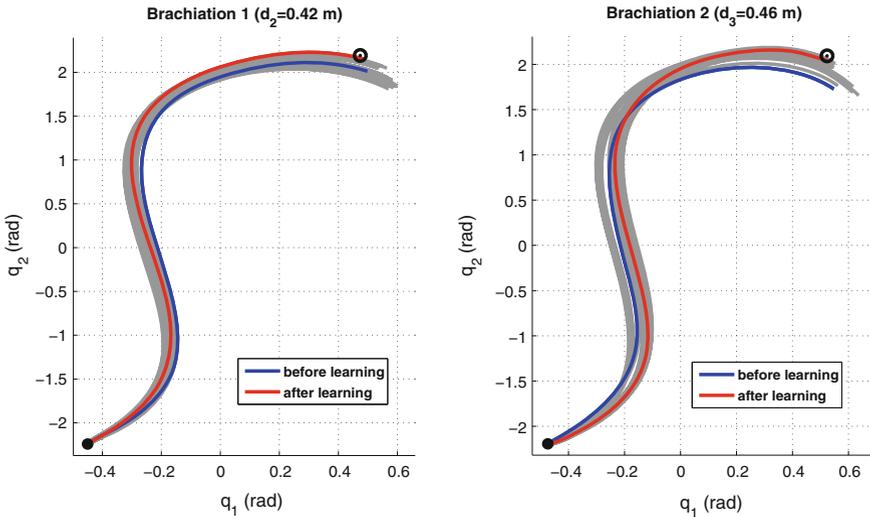
In Fig. 12, we compare the performance of the system under the (i) solution obtained from the nominal (incorrect) analytical model (blue) and (ii) solution obtained after training the LWPR model (red). We can observe that the error in the position of the end-effector (i.e., open gripper) at the end of the allocated time improved significantly in both brachiation tasks from (i) 0.0867 m and 0.1605 m to (ii) 0.0161 m and 0.0233 m, respectively. The final positions are close enough to allow the gripper to compensate for the rest of the error by grasping it, thus resulting in the final error of 0.004 m. Note that the true positions of the gripper are actually at the target as it is securely locked on the target bar. The error comes from the variability of the sensor readings. The experimental conditions for the individual phase learning in hardware presented in this section correspond to the second and third phases of the movement from the multi-phase brachiation task considered in this chapter.

Besides the mass change, we perform an additional experiment in which we also modify the stiffness of the spring in the MACCEPA actuator from  $\kappa_s = 771$  N/m to  $\kappa_s = 650$  N/m in the nominal model. As before, at the end of the procedure,

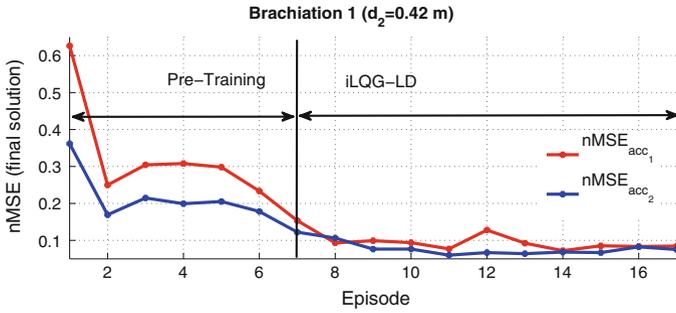
<sup>8</sup>With the reduced input dimensionality, practically, there could be the case that it is not possible to predict the full state of the system particularly in the swing-up motion due to unobserved input dimensions. Thus, we only considered the swing locomotion task in the hardware experiment with model learning.



**Fig. 11** Individual phase learning in hardware (mass discrepancy): Evolution of the nMSE for each phase of the movement at each episode

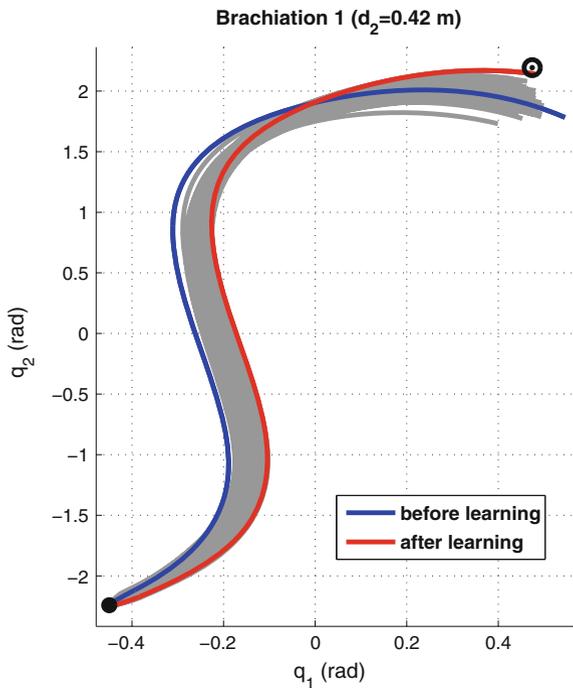


**Fig. 12** Individual phase learning in hardware (mass discrepancy). Final position: initial planning (inaccurate nominal model in *blue*) and final planning (composite model after learning in *red*). Intermediary iLQG-LD solutions (*grey*)



**Fig. 13** Individual phase learning in hardware (mass and stiffness discrepancy): Evolution of the nMSE for each phase of the movement at each episode

**Fig. 14** Individual phase learning in hardware (mass and stiffness discrepancy): Final position achieved when using the initial planning (inaccurate nominal model in *blue*) and the final planning (composite model after learning in *red*). Intermediary solutions obtained at each step of the iLQG-LD run and exploration trajectories are depicted in *grey*



the planned trajectory matched the behaviour obtained from running the command solution on the real plant (the final nMSE  $\approx 10^{-2}$ , Fig. 13).

In Fig. 14, we observe the improvement in performance from an initial reaching error (at the end of the planned time) of (i) 0.1265 m (blue line) to (ii) 0.0167m (red line). The robot was able to grab the bar located at the desired target with the optimised control command using the improved model. The experimental results

demonstrate the feasibility of the developed adaptive learning framework for the application to real-world systems.

## 4 Summary

In this chapter, we addressed the optimal control problem of robotic systems with VSAs including switching dynamics and discontinuous state transitions.

First, we presented a systematic methodology for movement optimisation with multiple phases and switching dynamics in robotic systems with variable stiffness actuation with the focus on exploiting intrinsic dynamics of the system. Tasks including switching dynamics and interaction with an environment are approximately modelled as a hybrid dynamical system with time-based switching. We have demonstrated the benefit of simultaneous temporal and variable stiffness optimisation leading to reduction in control effort and improved performance. With an appropriate choice of the composite cost function to encode the task, we have demonstrated the effectiveness of the proposed approach in various brachiation tasks in numerical simulations and hardware implementation in a brachiating robot with VSA. In [21], we have presented additional numerical evaluations of the proposed approach on the control of a hopping robot model with a VSA having different mode of dynamics (flight and stance) and impact with the environment. Simulation results on the hopping robot control in [21] illustrated the feasibility of our approach and the robustness of the obtained optimal feedback controller against external perturbations.

Next, we extended our approach by incorporating adaptive learning, which allows for adjustments to the dynamics model, based on changes occurred to the system's behaviour, or when the behaviour cannot be fully represented by a rigid body dynamics formulation. The method employed (in the form of the LWPR algorithm) is particularly suited for certain regression situations such as non-linear function learning with the requirement of incremental learning. We demonstrated that the augmented developed methodology was successfully applied in the case of underactuated systems such as a brachiating robot. We provided results for a range of model discrepancies in both numerical simulations and real hardware experiments.

In our previous work, we have addressed movement optimisation of variable impedance actuators including damping [28], and our framework presented in this chapter can be generalised to deal with such systems in a straightforward manner. Our future interest is in the application of our approach to a broad range of complex physical and robotic systems having interactions with environments.

## References

1. C.G. Atkeson, A.W. Moore, S. Schaal, Locally weighted learning for control. *Artif. Intell. Rev.* **11**(1–5), 75–113 (1997)
2. G. Bätz, U. Mettin, A. Schmidts, M. Scheint, D. Wollherr, A. S. Shiriaev, Ball dribbling with an underactuated continuous-time control phase: theory and experiments, in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2010), pp. 2890–2895
3. D. Braun, M. Howard, S. Vijayakumar, Optimal variable stiffness control: formulation and application to explosive movement tasks. *Auton. Robot.* **33**(3), 237–253 (2012)
4. D.J. Braun, F. Petit, F. Huber, S. Haddadin, P. van der Smagt, A. Albu-Schäffer, S. Vijayakumar, Robots driven by compliant actuators: optimal control under actuation constraints. *IEEE Trans. Robot.* **29**(5), 1085–1101 (2013)
5. A.E. Bryson, Y.-C. Ho, *Applied Optimal Control* (Taylor and Francis, United Kingdom, 1975)
6. M. Buehler, D.E. Koditschek, P.J. Kindlmann, Planning and control of robotic juggling and catching tasks. *Int. J. Robot. Res.* **13**(2), 101–118 (1994)
7. M. Buss, M. Glocker, M. Hardt, O. von Stryk, R. Bulirsch, G. Schmidt, Nonlinear hybrid dynamical systems: modeling, optimal control, and applications, in *Lecture Notes in Control and Information Science* (Springer, Heidelberg, 2002), pp. 311–335
8. T.M. Caldwell, T.D. Murphey, Switching mode generation and optimal estimation with application to skid-steering. *Automatica* **47**(1), 50–64 (2011)
9. M.G. Catalano, G. Grioli, M. Garabini, F. Bonomo, M. Mancini, N. Tsagarakis, A. Bicchi, VSA-CubeBot: A modular variable stiffness platform for multiple degrees of freedom robots, in *IEEE International Conference on Robotics and Automation* (2011), pp. 5090–5095
10. M. Gomes, A. Ruina, A five-link 2D brachiating ape model with life-like zero-energy-cost motions. *J. Theor. Biol.* **237**(3), 265–278 (2005)
11. K. Goris, J. Saldien, B. Vanderborght, D. Lefeber, Mechanical design of the huggable robot probo. *Int. J. Humanoid Robot.* **8**(3), 481–511 (2011)
12. S. S. Groothuis, G. Rusticelli, A. Zucchelli, S. Stramigioli, R. Carloni, The vsaUT-II: A novel rotational variable stiffness actuator, in *IEEE International Conference on Robotics and Automation* (2012), pp. 3355–3360
13. W. Li, E. Todorov, Iterative linearization methods for approximately optimal control and estimation of non-linear stochastic system. *Int. J. Control* **80**(9), 1439–1453 (2007)
14. A.W. Long, T.D. Murphey, K.M. Lynch, Optimal motion planning for a class of hybrid dynamical systems with impacts, in *IEEE International Conference on Robotics and Automation* (2011), pp. 4220–4226
15. D. Mitrovic, S. Klanke, M. Howard, S. Vijayakumar, Exploiting sensorimotor stochasticity for learning control of variable impedance actuators, in *IEEE-RAS International Conference on Humanoid Robots* (2010), pp. 536–541
16. D. Mitrovic, S. Klanke, S. Vijayakumar, Optimal control with adaptive internal dynamics models, in *Fifth International Conference on Informatics in Control, Automation and Robotics* (2008)
17. D. Mitrovic, S. Klanke, S. Vijayakumar, Adaptive optimal feedback control with learned internal dynamics models, in *From Motor Learning to Interaction Learning in Robots* (2010), pp. 65–84
18. K. Mombaur, Using optimization to create self-stable human-like running. *Robotica* **27**(3):321330 (2009)
19. J. Nakanishi, J.A. Farrell, S. Schaal, Composite adaptive control with locally weighted statistical learning. *Neural Netw.* **18**(1), 71–90 (2005)
20. J. Nakanishi, T. Fukuda, D. Koditschek, A brachiating robot controller. *IEEE Trans. Robot. Autom.* **16**(2), 109–123 (2000)
21. J. Nakanishi, A. Radulescu, D. J. Braun, S. Vijayakumar, Spatio-temporal stiffness optimization with switching dynamics. *Auton. Robot.* 1–19 (2016)

22. J. Nakanishi, K. Rawlik, S. Vijayakumar, Stiffness and temporal optimization in periodic movements: an optimal control approach, in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2011), pp. 718–724
23. D. Nguyen-Tuong, J. Peters, Model learning for robot control: a survey. *Cogn. Process.* **12**(4), 319–340 (2011)
24. F. Petit, M. Chalon, W. Friedl, M. Grebenstein, A. Albu-Schäffer, G. Hirzinger, Bidirectional antagonistic variable stiffness actuation: analysis, design and implementation, in *IEEE International Conference on Robotics and Automation* (2010), pp. 4189–4196
25. B. Piccoli, Hybrid systems and optimal control, in *IEEE Conference on Decision and Control* (1998), pp. 13–18
26. M. Posa, C. Cantu, R. Tedrake, A direct method for trajectory optimization of rigid bodies through contact. *Int. J. Robot. Res.* **33**(1), 69–81 (2014)
27. M. Posa, S. Kuindersma, R. Tedrake, Optimization and stabilization of trajectories for constrained dynamical systems, in *IEEE International Conference on Robotics and Automation* (2016), pp. 1366–1373
28. A. Radulescu, M. Howard, D. J. Braun, S. Vijayakumar, Exploiting variable physical damping in rapid movement tasks, in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (2012), pp. 141–148
29. A. Radulescu, J. Nakanishi, S. Vijayakumar, Optimal control of multi-phase movements with learned dynamics, in *Man-Machine Interactions 4* (Springer, Heidelberg, 2016), pp. 61–76
30. K. Rawlik, M. Toussaint, S. Vijayakumar, An approximate inference approach to temporal optimization in optimal control, in *Advances in Neural Information Processing Systems*, vol. 23 (MIT Press, Cambridge, 2010), pp. 2011–2019
31. N. Rosa Jr., A. Barber, R.D. Gregg, K.M. Lynch, Stable open-loop brachiation on a vertical wall, in *IEEE International Conference on Robotics and Automation* (2012), pp. 1193–1199
32. F. Saito, T. Fukuda, F. Arai, Swing and locomotion control for a two-link brachiation robot. *IEEE Control Syst. Mag.* **14**(1), 5–12 (1994)
33. S. Schaal, C.G. Atkeson, Constructive incremental learning from only local information. *Neural Comput.* **10**(8), 2047–2084 (1998)
34. M.S. Shaikh, P.E. Caines, On the hybrid optimal control problem: theory and algorithms. *IEEE Trans. Autom. Control* **52**(9), 1587–1603 (2007)
35. B. Siciliano, O. Khatib, *Springer Handbook of Robotics* (Springer, Heidelberg, 2008)
36. O. Sigaud, C. Salaün, V. Padois, On-line regression algorithms for learning mechanical models of robots: a survey. *Robot. Auton. Syst.* **59**, 1115–1129 (2011)
37. Y. Tassa, T. Erez, E. Todorov, Synthesis and stabilization of complex behaviors through on-line trajectory optimization, in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2012), pp. 2144–2151
38. M. Van Damme, B. Vanderborght, B. Verrelst, R. Van Ham, F. Daerden, D. Lefeber, Proxy-based sliding mode control of a planar pneumatic manipulator. *Int. J. Robot. Res.* **28**(2), 266–284 (2009)
39. R. Van Ham, B. Vanderborght, M. Van Damme, B. Verrelst, D. Lefeber, MACCEPA, the mechanically adjustable compliance and controllable equilibrium position actuator: design and implementation in a biped robot. *Robot. Auton. Syst.* **55**(10), 761–768 (2007)
40. B. Vanderborght, B. Verrelst, R. Van Ham, M. Van Damme, D. Lefeber, B.M.Y. Duran, P. Beyl, Exploiting natural dynamics to reduce energy consumption by controlling the compliance of soft actuators. *Int. J. Robot. Res.* **25**(4), 343–358 (2006)
41. S. Vijayakumar, S. Schaal, Locally weighted projection regression: An  $o(n)$  algorithm for incremental real time learning in high dimensional space, in *International Conference on Machine Learning, Proceedings of the Sixteenth Conference* (2000)
42. L.C. Visser, R. Carloni, S. Stramigioli, Energy-efficient variable stiffness actuators. *IEEE Trans. Robot.* **27**(5), 865–875 (2011)
43. W. Xi, C.D. Remy, Optimal gaits and motions for legged robots, in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2014), pp. 3259–3265

44. X. Xu, P.J. Antsaklis, Quadratic optimal control problems for hybrid linear autonomous systems with state jumps, in *American Control Conference (2003)*, pp. 3393–3398
45. X. Xu, P.J. Antsaklis, Results and perspectives on computational methods for optimal control of switched systems, in *International Workshop on Hybrid Systems: Computation and Control* (Springer, Heidelberg, 2003), pp. 540–555
46. X. Xu, P.J. Antsaklis, Optimal control of switched systems based on parameterization of the switching instants. *IEEE Trans. Autom. Control* **49**(1), 2–16 (2004)
47. C. Yang, G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schäeffler, E. Burdet, Human-like adaptation of force and impedance in stable and unstable interactions. *IEEE Trans. Robot.* **27**(5), 918–930 (2011)