Optimal Control of Multi-phase Movements with Learned Dynamics

Andreea Radulescu, Jun Nakanishi and Sethu Vijayakumar

Abstract In this paper, we extend our work on movement optimisation for variable stiffness actuation (VSA) with multiple phases and switching dynamics to incorporate scenarios with incomplete, complex or hard to model robot dynamics. By incorporating a locally weighted nonparametric learning method to model the discrepancies in the system dynamics, we formulate an online adaptation scheme capable of systematically improving the multi-phase plans (stiffness modulation and torques) and switching instances while improving the dynamics model on the fly. This is demonstrated on a realistic model of a VSA brachiating system with excellent adaptation results.

Keywords Optimal control · Variable stiffness actuators · Adaptive dynamics learning · Adaptive control

1 Introduction

The accuracy of model-based control is significantly dependent on that of the models themselves. Traditional robotics employs models obtained from mechanical engineering insights. Kinematic equations will provide accurate information about the evolution of a rigid body configuration, given a precise knowledge of its geometry. Similarly, the dynamics equation can incorporate well modelled factors such as inertia, Coriolis and centrifugal effect or external forces.

However, there are certain elements that cannot be fully captured by these models, such as friction from the joints or resulting from cable movement [22], which can vary in time. The introduction of flexible elements in the structure of a system increases

A. Radulescu · S. Vijayakumar (✉)
Institute of Perception, Action and Behaviour, University of Edinburgh, Edinburgh, UK
e-mail: sethu.vijayakumar@ed.ac.uk

J. Nakanishi
Institute for Cognitive Systems, Technical University of Munich, Munich, Germany

A. Gruca et al. (eds.), *Man–Machine Interactions 4*, Advances in Intelligent Systems and Computing 391, DOI 10.1007/978-3-319-23437-3_5

the complexity of the model and makes the identification of accurate dynamics significantly more difficult. Additionally, during operation, the robot can suffer changes in its mechanical structure due to wear and tear or due to the use of a tool (which modifies the mechanical chain structure) [23].

On-line adaptation of models can provide a solution for capturing all these properties. Early approaches, such as on-line *parameter identification* [21], which tunes the parameters of a predefined model (dictated by the mechanical structure) using data collected during operation, proved sufficient for accurate control and remained a popular approach for a long time [1, 7]. The increased complexity of latest robotic systems demands novel approaches capable of accommodating significant non-linear and unmodelled robot dynamics. Successful *non-parametric model learning* methods use supervised learning to perform system identification with only limited prior information about its structure—removing the restriction to a fixed model structure, allowing the model complexity to adapt in a data driven manner.

In this work, we will build on our significant prior efforts to engage this techniques in the context of robot control [11, 13, 17] and apply this in the context of multiphase variable impedance movements. Indeed, adaptive model learning has been used successfully in a wide range of scenarios such as inverse dynamics control [18, 26], inverse kinematics [5, 24], robot manipulation and locomotion [6, 20].

1.1 Adaptive Learning for Optimal Control

Classical OC (Optimal Control) is formulated using an analytic dynamics model, but recent work [2, 10] has shown that combining OC with dynamics learning can produce a powerful and principled control strategy for complex systems with redundant actuation.

In [10], using online (non-parametric) supervised learning methods, an adaptive internal model of the system dynamics is learned. The model is afterwards used to derive an optimal control law. This approach, named *iterative Linear Quadratic-Gaussian (iLQG) method with learned dynamics (iLQG-LD)*, proved efficient in a variety of realistic scenarios, including problems where the analytic dynamics model is difficult to estimate accurately or subject to changes and the system is affected by noise [10, 12].

In iLQG-LD the update of the dynamics model takes place on a trial-by-trial basis [12]. The operating principle (depicted in Fig. 1) is to (i) compute the iLQG solution, (ii) run the obtained control law on the plant and collect data, (iii) use the plant data to update the dynamics model.

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

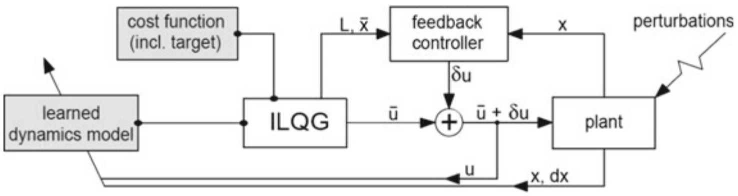

Fig. 1 The iLQG-LD learning and control scheme as first introduced in [12]

$\delta\mathbf{u} = \mathbf{L}(\mathbf{x} - \bar{\mathbf{x}})$, 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 $\mathbf{x}$.

This methodology employs the *Locally Weighted Projection Regression (LWPR)* [8] as the nonparametric learning scheme of choice to train a model of the dynamics in an incremental fashion. In LWPR, the regression function is constructed by combining local linear models. During training the parameters of the local models (locality and fit) are updated using incremental *partial least squares (PLS)*. PLS projects the input on to a small number of directions in the input space along the directions of maximal correlation with the output and then performs linear regression on the projected inputs. This makes LWPR suitable for high dimensional input spaces. Local models can be pruned or added on an as-need basis (e.g., when training data is generated in previously unexplored regions). The areas of validity (receptive fields) of each local model are modelled by Gaussian kernels. LWPR keeps a number of variables that hold sufficient statistics for the algorithm to perform the required calculations incrementally.

We incorporate the iLQG-LD scheme into our approach involving learning the dynamics of a brachiation system with VSA (variable stiffness actuator) capabilities and employing it in planning for locomotion tasks.

2 Problem Formulation

In our previous work [14], we introduced a general formulation of optimal control problems for tasks with multiple phase movements including switching dynamics and discrete state transition arising from iterations with an environment. Given a rigid body dynamics formulation of a robot with a VSA model, a hybrid dynamics representation with a composite cost function is introduced to describe such a task. In this section we briefly describe this approach, for details we refer the interested reader to [14]. We also introduce the changes dictated by the use of the LWPR method in the context of iLQG-LD, for integration within our approach.

2.1 Hybrid Dynamics with Time-Based Switching and Discrete State Transition

We employ the following hybrid dynamics representation to model multi-phase movements having interactions with an environment [4]:

$$\dot{\mathbf{x}} = \mathbf{f}_{i_j}(\mathbf{x}, \mathbf{u}), \quad T_j \le t < T_{j+1} \tag{1}$$

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

with $j = 0, \ldots, K$ for (1) and $j = 1, \ldots, K$ for (2) and where $\mathbf{f}_i : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the ith subsystem, $\mathbf{x} \in \mathbb{R}^n$ is a state vector, $\mathbf{u} \in \mathbb{R}^m$ is a control input vector.

When the dynamics switch from subsystem i_{j-1} to i_j at $t = T_j$, we assume that instantaneous discrete (discontinuous) state transition is introduced, which is denoted by a map $\boldsymbol{\Delta}^{i_{j-1}, i_j}$ in (2). The terms $\mathbf{x}(T_j^+)$ and $\mathbf{x}(T_j^-)$ denote the post- and pre-transition states, respectively. In this case, the sequence of switching is assumed to be given, e.g., $(1, 2, \ldots, K, K+1)$ or $(1, 2, 1, 2, \ldots)$. Figure 2 depicts a schematic diagram of a hybrid system we consider in this work.

2.2 Robot Dynamics with Variable Stiffness Actuation

To describe multi-phase movements, we consider multiple sets of robot dynamics, as described by (1). An individual rigid body dynamics model is defined for each associated phase of the movement as a subsystem. The servo motor dynamics in the VSA are modelled as a critically damped second order dynamics:

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

$$\ddot{\mathbf{q}}_m + 2\boldsymbol{\alpha}_i\dot{\mathbf{q}}_m + \boldsymbol{\alpha}_i^2\mathbf{q}_m = \boldsymbol{\alpha}_i^2\mathbf{u} \tag{4}$$

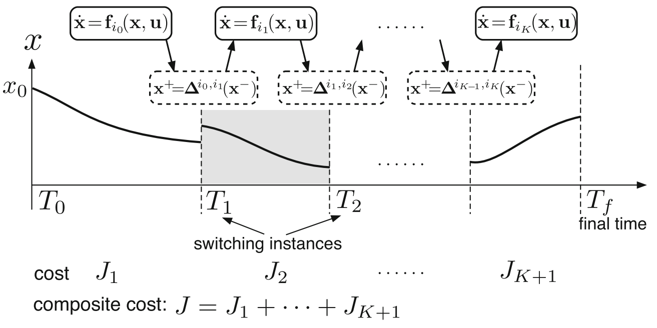
$\dot{\mathbf{x}} = \mathbf{f}_{i_0}(\mathbf{x}, \mathbf{u})$
$\dot{\mathbf{x}} = \mathbf{f}_{i_1}(\mathbf{x}, \mathbf{u})$
......
$\dot{\mathbf{x}} = \mathbf{f}_{i_K}(\mathbf{x}, \mathbf{u})$
x
x_0
$\mathbf{x}^+ = \boldsymbol{\Delta}^{i_0, i_1}(\mathbf{x}^-)$
$\mathbf{x}^+ = \boldsymbol{\Delta}^{i_1, i_2}(\mathbf{x}^-)$
$\mathbf{x}^+ = \boldsymbol{\Delta}^{i_{K-1}, i_K}(\mathbf{x}^-)$
T_0
T_1
T_2
T_f
final time
switching instances
cost J_1 J_2 ...... J_{K+1}
composite cost: $J = J_1 + \cdots + J_{K+1}$

Fig. 2 A hybrid system with time-based switching dynamics and discrete state transition with a known sequence. The objective is to find an optimal control command $\mathbf{u}$, switching instances T_i and final time T_f which minimises the composite cost J

where i denotes the ith subsystem, $\mathbf{q} \in \mathbb{R}^n$ is the joint angle vector, $\mathbf{q}_m \in \mathbb{R}^m$ is the motor position vector of the VSA, $\mathbf{M} \in \mathbb{R}^{n\times n}$ is the inertia matrix, $\mathbf{C} \in \mathbb{R}^n$ is the Coriolis term, $\mathbf{g} \in \mathbb{R}^n$ is the gravity vector, $\mathbf{D} \in \mathbb{R}^{n\times n}$ is the viscous damping matrix, and $\boldsymbol{\tau} \in \mathbb{R}^n$ are the joint torques from the variable stiffness mechanism. In the equations above, (3) denotes the rigid body dynamics of the robot and (4) denotes the servo motor dynamics in the variable stiffness actuator. In (4), $\boldsymbol{\alpha}$ determines the bandwidth of the servo motors[1] and $\mathbf{u}$ is the motor position command [3]. We assume that the range of control command $\mathbf{u}$ is limited between $\mathbf{u}_{min}$ and $\mathbf{u}_{max}$.

In this work, we consider a VSA model in which the joint torques are given in the form

$$\boldsymbol{\tau}(\mathbf{q}, \mathbf{q}_m) = \mathbf{A}^T(\mathbf{q}, \mathbf{q}_m)\mathbf{F}(\mathbf{q}, \mathbf{q}_m) \tag{5}$$

where $\mathbf{A}$ is the moment arm matrix and $\mathbf{F}$ is the forces by the elastic elements [3] and the joint stiffness is defined as $\mathbf{K} = -\frac{\partial \boldsymbol{\tau}}{\partial \mathbf{q}}$.

We consider the state space representation as the combined plant dynamics consisting of the rigid body dynamics (3) and the servo motor dynamics (4):

$$\dot{\mathbf{x}} = \mathbf{f}_i(\mathbf{x}, \mathbf{u}) \tag{6}$$

where

$$\mathbf{f}_i = \begin{bmatrix} \mathbf{x}_2 \\ \mathbf{M}_i^{-1}(\mathbf{x}_1)\left(-\mathbf{C}_i(\mathbf{x}_1, \mathbf{x}_2)\mathbf{x}_2 - \mathbf{g}_i(\mathbf{x}_1) - \mathbf{D}_i\dot{\mathbf{x}}_2 + \boldsymbol{\tau}_i(\mathbf{x}_1, \mathbf{x}_3)\right) \\ \mathbf{x}_4 \\ -\boldsymbol{\alpha}_i^2\mathbf{x}_3 - 2\boldsymbol{\alpha}_i\mathbf{x}_4 + \boldsymbol{\alpha}_i^2\mathbf{u} \end{bmatrix} \tag{7}$$

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

Employing the iLQG-LD framework we aim to create an accurate model of the dynamics model of the real hardware using supervised learning. We assume the existence of a preliminary analytic dynamics model which takes the form presented in (3), (4), which is inaccurate (due to various factors such as: the inability of the rigid body dynamics to incorporate all the elements of the system's behaviour or changes suffered during operation).

We use the LWPR method to model the error between the true behaviour of the system and the initial model provided. Thus we replace the dynamics $\mathbf{f}_i$ in (6) with the composite dynamics model $\mathbf{f}_{\mathbf{c}i}$:

$$\dot{\mathbf{x}} = \mathbf{f}_{\mathbf{c}i}(\mathbf{x}, \mathbf{u}) = \tilde{\mathbf{f}}_i(\mathbf{x}, \mathbf{u}) + \bar{\mathbf{f}}_i(\mathbf{x}, \mathbf{u}) \tag{8}$$

where $\tilde{\mathbf{f}}_i \in \mathbb{R}^{2(n+n)}$ is the initial inaccurate model and $\bar{\mathbf{f}} \in \mathbb{R}^{2(n+n)}$ is the LWPR model mapping the discrepancy between $\tilde{\mathbf{f}}_i \in \mathbb{R}^{2(n+n)}$ and the behaviour of the

[1] $\boldsymbol{\alpha} = \text{diag}\{a_1, \ldots, a_m\}$ and $\boldsymbol{\alpha}^2 = \text{diag}\{a_1^2, \ldots, a_m^2\}$ for notational convenience.

system. We note that the changes introduced by iLQG-LD only affect the dynamics modelling in (1), while the instantaneous state transition mapped by $\mathbf{\Delta}$ in (2) remains unchanged.

2.3 Movement Optimisation of Multiple Phases

For the given hybrid dynamics, in order to describe the full movement with multiple phases, we consider the following composite cost function:

$$J = \phi(\mathbf{x}(T_f)) + \sum_{j=1}^{K} \psi^j(\mathbf{x}(T_j^-)) + \int_{T_0}^{T_f} h(\mathbf{x}, \mathbf{u})dt \tag{9}$$

where $\phi(\mathbf{x}(T_f))$ is the terminal cost, $\psi^j(\mathbf{x}(T_j^-))$ is the via-point cost at the jth switching instance and $h(\mathbf{x}, \mathbf{u})$ is the running cost.

2.4 Optimal Control of Switching Dynamics and Discrete State Transition

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

$$\delta\mathbf{x}_{k+1} = \mathbf{A}_k\delta\mathbf{x}_k + \mathbf{B}_k\delta\mathbf{u}_k \tag{10}$$

$$\delta\mathbf{x}_{k_j}^+ = \mathbf{\Gamma}_{k_j}\delta\mathbf{x}_{k_j}^- \tag{11}$$

$$\mathbf{A}_k = \mathbf{I} + \Delta t_j \left.\frac{\partial \mathbf{f}_{i_j}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_k}, \quad \mathbf{B}_k = \Delta t_j \left.\frac{\partial \mathbf{f}_{i_j}}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u}_k} \tag{12}$$

$$\mathbf{\Gamma}_{k_j} = \left.\frac{\partial \mathbf{\Delta}^{i_{j-1}, i_j}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_{k_j}^-} \tag{13}$$

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, Δt_j is the sampling time for the time interval $T_j \le t < T_{j+1}$, and k_j is the jth switching instance in the discretised time step.

When using the composite model of the dynamics ($\mathbf{f_c}$) introduced in (8) 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 (10) and (12). The derivatives of the learned model ($\bar{\mathbf{f}}$) are obtained analytically by differentiating with respect to the inputs $\mathbf{z} = (\mathbf{x}; \mathbf{u})$ as suggested in [2]:

$$\frac{\partial \bar{\mathbf{f}}(\mathbf{z})}{\partial \mathbf{z}} = \frac{1}{W}\sum_k \left(\frac{\partial w_k}{\partial \mathbf{z}}\psi_k(\mathbf{z}) + w_k\frac{\partial \psi_k}{\partial \mathbf{z}}\right) - \frac{1}{W^2}\sum_k w_k(\mathbf{z})\psi_k(\mathbf{z})\sum_l \frac{\partial w_l}{\partial \mathbf{z}}$$

$$= \frac{1}{W}\sum_k (-\psi_k w_k \mathbf{D}_k(\mathbf{z}-\mathbf{c}_k) + w_k \mathbf{b}_k) + \frac{\bar{\mathbf{f}}(\mathbf{z})}{W}\sum_k w_k \mathbf{D}_k(\mathbf{z}-\mathbf{c}_k) \quad (14)$$

where

$$\frac{\partial \bar{\mathbf{f}}(\mathbf{z})}{\partial \mathbf{z}} = \begin{pmatrix} \partial \bar{\mathbf{f}}/\partial \mathbf{x} \\ \partial \bar{\mathbf{f}}/\partial \mathbf{u} \end{pmatrix}. \quad (15)$$

Since there are no changes on the encoding of the instantaneous state transition (2) the equations in (11) and (13) remain unchanged for the iLQG-LD framework.

The composite cost function (9) is locally approximated in a quadratic form as

$$\begin{aligned} \Delta J &= \delta\mathbf{x}_N^T \phi_\mathbf{x} + \frac{1}{2}\delta\mathbf{x}_N^T\phi_{\mathbf{xx}}\delta\mathbf{x}_N + \sum_{j=1}^{K}\left((\delta\mathbf{x}_{k_j}^-)^T\psi_\mathbf{x}^j + \frac{1}{2}(\delta\mathbf{x}_{k_j}^-)^T\psi_{\mathbf{xx}}^j\delta\mathbf{x}_{k_j^-}\right) \\ &+ \sum_{k=1}^{N}\left(\delta\mathbf{x}_k^T h_\mathbf{x} + \delta\mathbf{u}_k^T h_\mathbf{u} + \frac{1}{2}\delta\mathbf{x}_k^T h_{\mathbf{xx}}\delta\mathbf{x}_k + \frac{1}{2}\delta\mathbf{u}_k^T h_{\mathbf{uu}}\delta\mathbf{u}_k + \delta\mathbf{u}_k h_{\mathbf{ux}}\delta\mathbf{x}_k\right)\Delta t_j \end{aligned} \quad (16)$$

and a local quadratic approximation of the optimal cost-to-go function is

$$v_k(\delta\mathbf{x}_k) = \frac{1}{2}\delta\mathbf{x}_k^T\mathbf{S}_k\delta\mathbf{x}_k + \delta\mathbf{x}_k^T\mathbf{s}_k. \quad (17)$$

For notational convenience, note that in (16), $\phi_\mathbf{x}$ and $\phi_{\mathbf{xx}}$ denote $\phi_\mathbf{x} = \frac{\partial\phi}{\partial\mathbf{x}}$ and $\phi_{\mathbf{xx}} = \frac{\partial^2\phi}{\partial\mathbf{x}^2}$, respectively. Similar definitions apply to other partial derivatives.

The local control law $\delta\mathbf{u}_k$ of the form

$$\delta\mathbf{u}_k = \mathbf{l}_k + \mathbf{L}_k\delta\mathbf{x}_k \quad (18)$$

is obtained from the Bellman equation

$$v_k(\delta\mathbf{x}_k) = min_{\delta\mathbf{u}}\{h_k(\delta\mathbf{x}_k, \delta\mathbf{u}_k) + v_{k+1}(\delta\mathbf{x}_{k+1})\} \quad (19)$$

by substituting (10) and (17) into the Eq. (19), where h_k is the local approximation of the running cost in (16) (see [9] for details).

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 $\bar{\mathbf{x}}$ is computed by running the obtained control $\bar{\mathbf{u}}$ and the above process is iterated until convergence.

In order to optimise the switching instances and the total movement duration, we introduce a scaling parameter and sampling time for each duration between switching as (cf. (12) and (16)):

$$\Delta t'_j = \frac{1}{\beta_j}\Delta t_j \quad \text{for} \quad T_j \le t < T_{j+1}, \quad \text{where} j = 0, \dots, K. \tag{20}$$

By optimising the vector of temporal scaling factors $\boldsymbol{\beta} = [\ \beta_0,\ \dots,\ \beta_K\ ]^T$ via gradient descent [19] we obtain each switching instance T_{j+1} and the total movement duration T_f. This approach was applied previously [15, 16] to optimise the frequency of the periodic movement and the movement duration of swing locomotion in a brachiation task.

In the complete optimisation, computation of optimal feedback control law and temporal scaling parameter update are iteratively performed until convergence. A pseudocode of the complete algorithm is summarised in Algorithm 1.

3 Brachiation System Dynamics

We evaluate the effectiveness of the approach on a robot brachiation task which incorporates switching dynamics and multiple phases of the movement in a realistic VSA actuator model. We consider a two-link underactuated brachiating robot with a MACCEPA [25] variable stiffness actuator. The equation of motion of the system used takes the standard form of rigid body dynamics where only the second joint is actuated[2]:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{D}\dot{\mathbf{q}} = \begin{bmatrix} 0 \\ \tau(\mathbf{q}, \mathbf{q}_m) \end{bmatrix} \tag{21}$$

where $\mathbf{q} = [q_1,\ q_2]^T$ is the joint angle vector, $\mathbf{M}$ is the inertia matrix, $\mathbf{C}$ is the Coriolis term, $\mathbf{g}$ is the gravity vector, $\mathbf{D}$ is the viscous damping matrix, τ is the joint torque acting on the second joint given by the VSA, and $\mathbf{q}_m$ is the motor positions vector in the VSA as described below.

The MACCEPA actuator is equipped with two position controlled servo motors, $\mathbf{q}_m = [q_{m1},\ q_{m2}]^T$, which control the equilibrium position and the spring

[2]For notational simplicity, the subscript i is omitted.

Algorithm 1 Complete optimisation algorithm for hybrid dynamics with temporal optimisation

```
1: Input:
   – Timed switching plant dynamics $\mathbf{f}_i$ (1 or 8), discrete state transition $\mathbf{\Delta}^{i_{j-1},i_j}$ (2) and switching sequence
   – Composite cost function $J$ (9)
2: Initialise:
   – Nominal switching instance and final time $T_1, \cdots, T_K$ and $T_f$
   – Nominal control sequence $\bar{\mathbf{u}}$ and corresponding $\bar{\mathbf{x}}$
3: repeat
4:   repeat
5:     Optimise control sequence $\bar{\mathbf{u}}$:
       – Obtain linearised time-based switching dynamics (10 or 14) and state transition (11) around $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$ in discrete time with current $\Delta t_j$
       – Compute quadratic approximation of the composite cost (16)
       – Solve local optimal control problem to obtain $\delta\mathbf{u}$ (18)
       – Apply $\delta\mathbf{u}$ to the linearised hybrid dynamics (10) and (16)
       – Update nominal control sequence $\bar{\mathbf{u}} \leftarrow \bar{\mathbf{u}} + \delta\mathbf{u}$, trajectory $\bar{\mathbf{x}}$ and cost $J$
6:   until convergence
7:   Temporal optimisation: update $\Delta t_j$:
     – Update the vector of temporal scaling factor $\boldsymbol{\beta}$ and corresponding sampling time $\Delta t_0, \cdots, \Delta t_K$ in (20) via gradient descent [19].
8: until convergence
9: Output:
   – Optimal feedback control law $\mathbf{u}(\mathbf{x}, t)$: forward optimal control sequence $\mathbf{u}_{opt}$, optimal trajectory $\mathbf{x}_{opt}(t)$ and optimal gain matrix $\mathbf{L}_{opt}(t)$:
     $\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_{opt}(t) + \mathbf{L}_{opt}(t)(\mathbf{x}(t) - \mathbf{x}_{opt}(t))$
   – Optimal switching instance $T_1, \cdots, T_K$ and final time $T_f$
   – Optimal composite cost $J$
```

pre-tension,[3] respectively. The servo motor dynamics are approximated by a second order system with a PD feedback control, as mentioned in (4):

$$\ddot{\mathbf{q}}_m + 2\boldsymbol{\alpha}\dot{\mathbf{q}}_m + \boldsymbol{\alpha}^2\mathbf{q}_m = \boldsymbol{\alpha}^2\mathbf{u} \tag{22}$$

where $\mathbf{u} = [u_1,\ u_2]^T$ is the motor position command, $\boldsymbol{\alpha}$ determines the bandwidth of the actuator. In this study, we use $\boldsymbol{\alpha} = \mathrm{diag}(20, 25)$. The range of the commands of the servo motors are limited as $u_1 \in [-\pi/2, \pi/2]$ and $u_2 \in [0, \pi/2]$.

We use the model parameters shown in Table 1 and the MACCEPA parameters with the spring constant $\kappa = 771$ N/m, the lever length $B = 0.03$ m, the pin displacement $C = 0.125$ m and the drum radius $r_d = 0.01$ m (Fig. 3).

[3] Which is used to modulate the stiffness of the joint, for details see [25].

Table 1 Model parameters of the two-link brachiating robot

Robot parameters		i = 1	i = 2	i = 1
Mass	m_i (kg)	1.390	0.527	*1.240*
Moment of inertia	I_i (kg m^2)	0.0297	0.0104	*0.0278*
Link length	l_i (m)	0.46	0.46	*0.46*
COM location	l_{ci} (m)	0.362	0.233	*0.350*
Viscous friction	d_i (Nm/s)	0.03	0.035	*0.03*

The final column shows the change of parameters of the first link of the system under the changed mass distribution described in Sect. 4

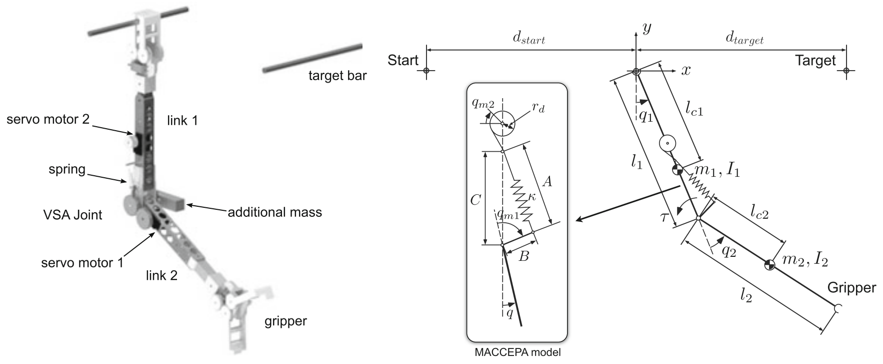

Fig. 3 Two-link brachiating robot model with the VSA joint with the inertial and geometric parameters. The parameters of the robot are given in Table 1, where the indices i denote the link number in this figure and Table 1

4 Experimental Setup

To test the efficiency of our approach we create a scenario where the difference between the true and the assumed model is caused by a change in the mass (and implicitly mass distribution) on one the links (i.e. the mass of the true model is smaller by 150 g (located at the joint) on link $i = 1$). The changed model parameters are shown in the right column of Table 1.[4]

Due to the nature of the discrepancy introduced, the error in the dynamics manifests itself only in the joint accelerations. Thus, we require to map the error just in those two dimensions, reducing the dimension of the output of the LWPR model $\bar{\mathbf{f}}$ from $n = 8$ to 2, where the predictions are added on the corresponding dimension of $\tilde{\mathbf{f}}$ within $\mathbf{f_c}$ (8). Note that different discrepancies will necessitate estimation of the full 8-dim state error.

In line with previous work, we will demonstrate the effectiveness of the proposed approach on a multi-phase, asymmetric swing-up and brachiation task with a VSA

[4]The MACCEPA parameters are the same as described in the previous section.

while incorporating *continuous, online model learning*. Specifically, in the multi-phase task, the robot 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, respectively.

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. The procedure used is briefly described in Algorithm 2. The initial exploration loop is performed in order to pre-train the LWPR model $\bar{\mathbf{f}}_i$ (as an alternative to applying motor babbling), the later loop is using iLQG-LD to refine the model in an online fashion. In our experiments the training data is obtained by using a simulated version of the true dynamics, which is an analytic model incorporating the discrepancy.

Algorithm 2 Description of the learning and exploration procedure

```
Given:
– analytic dynamics for one configuration $\tilde{\mathbf{f}}_i$ and start state $\mathbf{x}_0$
– thresholds for target reaching $\varepsilon_T$ and model accuracy $\varepsilon_M$
– the associated cost function $J$ (including desired target $\mathbf{x}_T$)
– $p$ number of initial exploration training episodes
Initialise
– $\bar{\mathbf{f}}_i(\mathbf{x}, \mathbf{u})$ ; $\mathbf{f_c}_i(\mathbf{x}, \mathbf{u})$=$\tilde{\mathbf{f}}_i(\mathbf{x}, \mathbf{u}) + \bar{\mathbf{f}}_i(\mathbf{x}, \mathbf{u})$
repeat
  generate $\bar{\mathbf{u}}, \bar{\mathbf{x}}, \mathbf{L}$ using $\tilde{\mathbf{f}}_i(\mathbf{x}, \mathbf{u})$ for an artificial target (a new target at each iteration obtained by sampling around $\mathbf{x}_T$)
  apply the solution to the true dynamics and train the model on the collected data
until $p$ training episodes have been performed

repeat
  apply iLQG-LD for target $\mathbf{x}_T$
until $\varepsilon_T$ and $\varepsilon_M$ conditions are met
```

4.1 Individual Phase Learning

Using the traditional OC framework in the presence of an accurate dynamics model, the multi-phase task described previously was achieved with a position error of just 0.002 m. Once the discrepancy detailed in Sect. 4 is introduced, the planned solution is no longer valid and the final position deviates from the desired target (Fig. 5, blue line). We deploy the iLQG-LD framework in order to learn the new behaviour of the system and recover the task performance.

As a measure of the model accuracy we use the *normalised mean square error* (nMSE) of the model prediction on the true optimal trajectory (if given access to the analytic form of the true dynamics of the system). The nMSE is defined as $nMSE(y, \tilde{y}) = \frac{1}{n\sigma_y^2} \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2$ where y is the desired output data set of size n

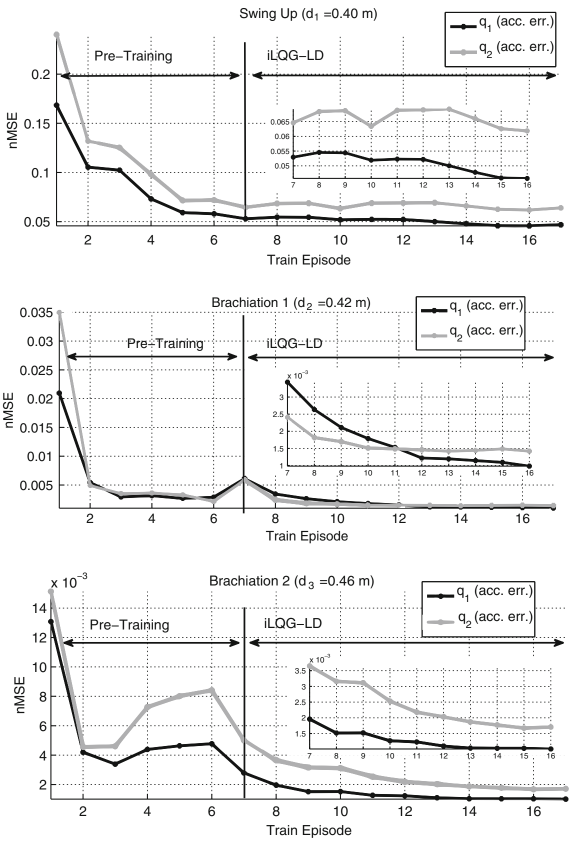

Fig. 4 Evolution of the nMSE for each phase of the movement, at each episode

and $\tilde{y}$ represents the LWPR predictions. The evolution of the nMSE at each stage of the training for every phase is shown in Fig. 4.

In the first part (pre-training phase in Fig. 4) we generate random targets around the desired x_T. A movement is planned for these targets using the assumed model ($\tilde{\mathbf{f}}$). The obtained command solution is then applied to the simulated version of the true dynamics, using a closed loop control scheme. We repeat the procedure for a set of 10 noise contaminated versions of the commands. The collected data is used to train the model.

This pre-training phase seeds the model with information within the region of interest, prior to using it for planning. This reduces the load on the iLQG-LD by lowering the number of iterations required for convergence. 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}).

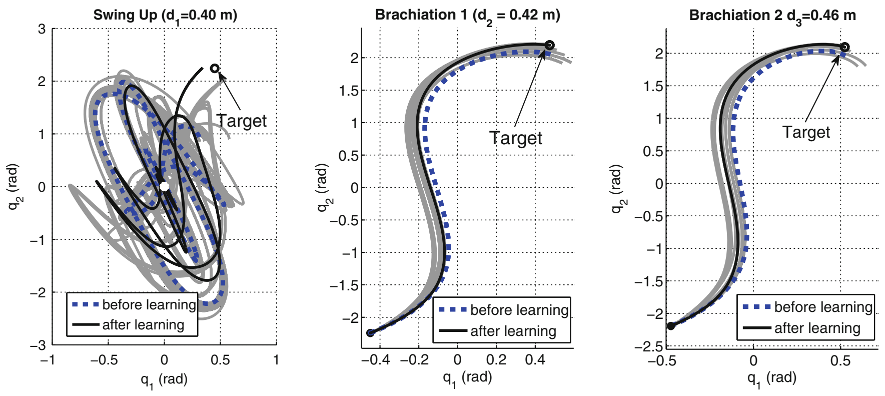

Fig. 5 Phase plot: comparison of the final position achieved (for each individual phase) when using the initial planning (erroneous model—*blue*) and the final planning (composite model—*black*). Intermediary solutions obtained at each step of the iLQG-LD run are depicted in *grey*

Overall the discrepancy is small enough to allow reaching the desired end effector position within a threshold of $\varepsilon_T = 0.040$ m accuracy.[5] Figure 5 shows the effect of the learning by comparing the performance of the planning with the erroneous model and with the composite model obtained after training.

4.2 *Multi-phase Performance*

In the previous section we showed that our approach to iLQG-LD is able to cope with the requirements of the task in each phase of the movement. For a full solution we use the newly learned models from each phase to obtain the global solution for the multi-phase task wrt. the composite cost function J (9). We use the phase optimal solutions obtained at the previous stage as the initial command sequence, the resulting behaviour is displayed in Fig. 6. The planner is able to use the learned model to achieve the intermediary and final goals, while the expected behaviour provides a reliable match to the actual system's behaviour.[6] The cost of multi-phase optimised solution ($J = 39.17$) is significantly lower than the sum of the costs of the individual phase solutions ($J = 58.23$).

[5]The error in the swing up task is 0.033 m, while for brachiations the value is 0.004 m. In future work we aim to bring the former value to the same magnitude.

[6]We consider that if the position at the end of each phase is within our prescribed threshold $\varepsilon_T = 0.040$ of the desired target the system is able to start the next phase from the ideal location, thus resembling the effect of the grabber on the hardware.

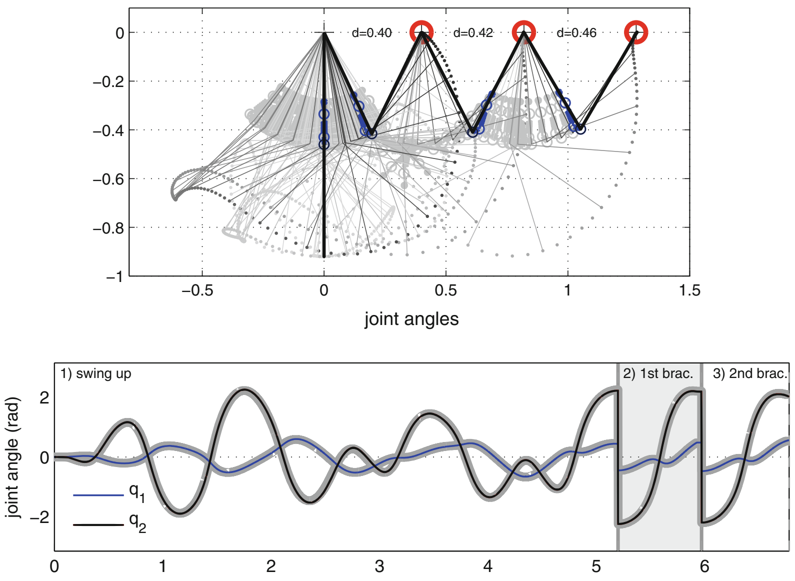

Fig. 6 Performance of the fully optimised multi-phase task using the composite model. *Thick grey lines* planned movement. *Black* and *blue lines* actual system movement

4.3 *Performance of Learning*

In the previous experiment, we investigated a single (arbitrarily chosen) mass distribution discrepancy. Next we investigate the capacity of our approach to cope with a wider range of mismatched dynamics. For this, we consider the magnitude of the change that is bounded by the capability of the altered system to achieve all the phases of the movement presented in Fig. 6. We define these bounds as the limit values of mass change that allow the same accuracy in task execution, under the same cost function (9). The corresponding values for these limits, found empirically, are -0.200 kg and $+0.300$ kg, respectively.[7]

We apply our framework to the scenarios where the mass has been altered to these boundary values and demonstrate the result on just one of the phases, namely the first brachiation move, to study the relative effects.

Figure 7a shows the effect of the alteration introduced, when executing the commands resulting from the initial planning (using the erroneous model). After training and re-planning the model starts approximating the true behaviour of the system, such that in less than 10 episodes, the system is once again able to reach the desired target, as depicted in Fig. 7b, c.

[7]We note that in our experiments, we assume that modulating the mass distribution does not affect the motor dynamics—this represents a simplified scenario. In the real hardware, the speed of the motor dynamics (4) is indeed a function of the overall load distribution.

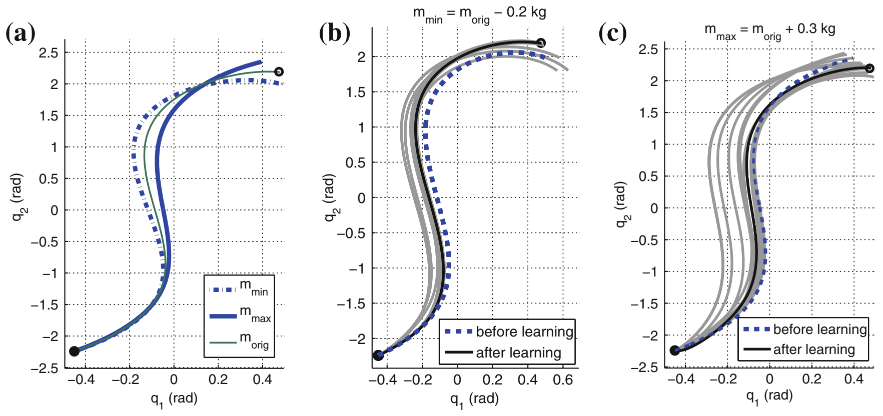

Fig. 7 Phase plot: **a** Effect of the discrepancy introduced by the limit values on the mass modulation (*blue lines*). The behaviour when the model match is correct is depicted for comparison (*green line*). **b–c** Comparison of the final position achieved (for each individual phase) when using the initial planning (erroneous model—*blue*) and the final planning (composite model—*black*). Intermediary solutions obtained at each step of the iLQG-LD run are depicted in *grey*. Results for the boundary value discrepancies on the mass distribution

5 Conclusion

In this work we have presented an extension of our methodology for movement optimisation with multiple phases and switching dynamics, including variable impedance actuators. We broaden the 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 capture by a rigid body dynamics formulation. In future work we aim to investigate a wider range of model discrepancies and show the performance of the extended approach on a hardware implementation.

References

1. Atkeson, C.G., An, C.H., Hollerbach, J.M.: Estimation of inertial parameters of manipulator loads and links. Int. J. Robot. Res. **5**(3), 101–119 (1986)
2. Atkeson, C., Moore, A., Schaal, S.: Locally weighted learning for control. Artif. Intell. Rev. **11**(1–5), 75–113 (1997)
3. Braun, D., Howard, M., Vijayakumar, S.: Optimal variable stiffness control: formulation and application to explosive movement tasks. Auton. Robots **33**(3), 237–253 (2012)
4. Caldwell, T.M., Murphey, T.D.: Switching mode generation and optimal estimation with application to skid-steering. Automatica **47**(1), 50–64 (2011)
5. Herbort, O., Butz, M., Pedersen, G.: The sure_reach model for motor learning and control of a redundant ARM: from modeling human behavior to applications in robotics. In: Sigaud, O.,

Peters, J. (eds.) From Motor Learning to Interaction Learning in Robots, Studies in Computational Intelligence. vol. 264, pp. 85–106. Springer, Heidelberg (2010)

6. Kalakrishnan, M., Buchli, J., Pastor, P., Schaal, S.: Learning locomotion over rough terrain using terrain templates. In: IROS 2009, pp. 167–172. St. Louis, USA (2009)
7. Khalil, W., Dombre, É.: Modeling, Identification and Control of Robots. Kogan Page Science, London (2004)
8. Klanke, S., Vijayakumar, S., Schaal, S.: A library for locally weighted projection regression. J. Mach. Learn. Res. **9**, 623–626 (2008)
9. Li, W., Todorov, E.: Iterative linearization methods for approximately optimal control and estimation of non-linear stochastic system. Int. J. Control **80**(9), 1439–1453 (2007)
10. Mitrovic, D., Klanke, S., Vijayakumar, S.: Adaptive optimal feedback control with learned internal dynamics models. In: Sigaud, O., Peters, J. (eds.) From Motor Learning to Interaction Learning in Robots, Studies in Computational Intelligence, vol 264, pp. 65–84 Springer, Berlin, Heidelberg (2010)
11. Mitrovic, D., Klanke, S., Howard, M., Vijayakumar, S.: Exploiting sensorimotor stochasticity for learning control of variable impedance actuators. In: Humanoids, pp. 536–541. Nashville, USA (2010)
12. Mitrovic, D., Klanke, S., Vijayakumar, S.: Optimal control with adaptive internal dynamics models. In: ICINCO 2008, Madeira, Portugal (2008)
13. Nakanishi, J., Farrell, J.A., Schaal, S.: Composite adaptive control with locally weighted statistical learning. Neural Netw. **18**(1), 71–90 (2005)
14. Nakanishi, J., Radulescu, A., Vijayakumar, S.: Spatio-temporal optimization of multi-phase movements: Dealing with contacts and switching dynamics. In: IROS 2013, pp. 5100–5107. Tokyo, Japan (2013)
15. Nakanishi, J., Rawlik, K., Vijayakumar, S.: Stiffness and temporal optimization in periodic movements: an optimal control approach. In: IROS, pp. 718–724. San Francisco, USA (2011)
16. Nakanishi, J., Vijayakumar, S.: Exploiting passive dynamics with variable stiffness actuation in robot brachiation. In: Robotics: Science and Systems, Sydney, Australia, July 2012
17. Nguyen-Tuong, D., Peters, J.: Model learning for robot control: a survey. Cogn. Process. **12**(4), 319–340 (2011)
18. Nguyen-Tuong, D., Seeger, M., Peters, J.: Model learning with local gaussian process regression. Adv. Robot. **23**(15), 2015–2034 (2009)
19. Rawlik, K., Toussaint, M., Vijayakumar, S.: An approximate inference approach to temporal optimization in optimal control. In: NIPS 2010, pp. 2011–2019. Vancouver, Canada (2010)
20. Schaal, S., Atkeson, C.G., Vijayakumar, S.: Scalable techniques from nonparametric statistics for real time robot learning. Appl. Intell. **17**(1), 49–60 (2002)
21. Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G.: Robotics: Modelling, Planning and Control. In: Springer Science & Business Media. Berlin, Germany (2009)
22. Siciliano, B., Khatib, O.: Springer Handbook of Robotics. Springer, Berlin (2008)
23. Sigaud, O., Salaün, C., Padois, V.: On-line regression algorithms for learning mechanical models of robots: a survey. Robot. Auton. Syst. **59**(12), 1115–1129 (2011)
24. Ting, J.A., Kalakrishnan, M., Vijayakumar, S., Schaal, S.: Bayesian kernel shaping for learning control. In: NIPS 2009, pp. 1673–1680. Vancouver, Canada (2009)
25. Van Ham, R., Vanderborght, B., Van Damme, M., Verrelst, B., Lefeber, D.: 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)
26. Vijayakumar, S., D'souza, A., Shibata, T., Conradt, J., Schaal, S.: Statistical learning for humanoid robots. Auton. Robots **12**(1), 55–69 (2002)