Adaptive Optimal Feedback Control with Learned Internal Dynamics Models

Djordje Mitrovic, Stefan Klanke, and Sethu Vijayakumar

Abstract. Optimal Feedback Control (OFC) has been proposed as an attractive movement generation strategy in goal reaching tasks for anthropomorphic manipulator systems. Recent developments, such as the Iterative Linear Quadratic Gaussian (ILQG) algorithm, have focused on the case of non-linear, but still analytically available, dynamics. For realistic control systems, however, the dynamics may often be unknown, difficult to estimate, or subject to frequent systematic changes. In this chapter, we combine the ILQG framework with learning the forward dynamics for simulated arms, which exhibit large redundancies, both, in kinematics and in the actuation. We demonstrate how our approach can compensate for complex dynamic perturbations in an online fashion. The specific adaptive framework introduced lends itself to a computationally more efficient implementation of the ILQG optimisation without sacrificing control accuracy – allowing the method to scale to large DoF systems.

1 Introduction

The human motion apparatus is by nature a highly redundant system and modern humanoid robots, designed to mimic human behaviour and performance, typically exhibit large degrees of freedom (DoF) in the kinematics domain (joints) and in the dynamics domain (actuation). Many recent humanoid system designs are extending the classic joint torque operated designs (i.e., one motor per joint) by redundantly actuated systems based on antagonistic or pseudo-antagonistic architectures (e.g., [11, 32]). Therefore producing even the simplest movement, such as reaching towards a particular position, involves an enormous amount of information processing and a controller has to make a choice from a very large space of possible movements

Djordje Mitrovic, Stefan Klanke, and Sethu Vijayakumar
School of Informatics, University of Edinburgh,
10 Crichton Street, Edinburgh EH8 9AB, United Kingdom
e-mail: {d.mitrovic,s.klanke,sethu.vijayakumar}@ed.ac.uk

O. Sigaud, J. Peters (Eds.): From Mot. Learn. to Inter. Learn. in Rob., SCI 264, pp. 65–84.
springerlink.com

to achieve a task. An important question to answer therefore is how to resolve this redundancy?

Optimal control theory [23] answers this question by establishing a certain cost function, and selecting the solution with minimal cost (e.g., minimum jerk [10], minimum torque change [29]). Quite often these control schemes are only concerned with trajectory *planning* and an open loop optimisation of the control commands, while the correction of errors during *execution* is left to simple PID controllers. As an alternative, closed loop optimisation models are aimed at providing a control law which is explicitly based on feedback from the system. In the ideal case, the system state is directly mapped to control signals during execution, and the form of this mapping is again governed by a cost function [25].

Another characteristic property of anthropomorphic systems, besides the high redundancies, is a lightweight and flexible-joint construction which is a key ingredient for achieving compliant human-like motion. However such a morphology complicates analytic dynamics calculations, which usually are based on unrealistic rigid body assumptions. Moreover, even if the different links of a manipulator could be modelled as a rigid body, the required parameters such as mass and inertia may be unknown or hard to estimate. Finally, unforeseen changes in the plant dynamics are hard to model based purely on analytic dynamics. In order to overcome these shortcomings we can employ online supervised learning methods to extract dynamics models driven by data from the movement system itself. This enables the controller to adapt *on the fly* to changes in dynamics conditions due to wear and tear or external perturbations. Applying such methods has previously been studied in robot control [8, 6, 18, 31] but has not found much attention in the perspective of the optimal control framework. Indeed the ability to adapt to perturbations is a key feature of biological motion systems and enabling optimal control to be adaptive is a valuable theoretical test-bed for human adaptation experiments.

By combining optimal control with dynamics learning we can create a powerful framework for the realisation of efficient control for high dimensional systems. This will provide a viable and principled control strategy for the biomorphic based highly redundant actuation systems that are currently being developed. Furthermore, we would like to exploit this framework for understanding optimal control and its link to biological motor control.

2 Optimal Feedback Control

In the past, although many control problems have been described within the framework of optimality, most optimal motor control models have focused on open loop (feed-forward) optimisation [10, 29]. Assuming deterministic dynamics (i.e., no perturbations or noise), open-loop control will produce a sequence of optimal motor signals or limb states. However if the system leaves the optimal path due to inevitable modelling imperfections, it must be corrected for example with a hand-tuned PID controller. This will often lead to suboptimal behaviour, because the error feedback has not been incorporated into the optimisation process.

Stable optimal performance can only be achieved by constructing an *optimal feedback law* that produces a mapping from states to actions by making use of all available sensory data. In such a scheme, which is referred to as *optimal feedback control* (OFC) [26], there is no separation anymore between trajectory planning and trajectory execution for the completion of a given task. Rather, one directly seeks to obtain the gains of a feedback controller which produce an optimal mapping from state to control signals (control law). A key property of OFC is that errors are only corrected by the controller if they adversely affect the task performance, otherwise they are neglected (minimum intervention principle [27]). This is an important property especially in systems that suffer from control dependent noise, since task-irrelevant correction could destabilise the system beside expending additional control effort.

In this work, we focus on the investigation of OFC in limb reaching movements for highly nonlinear and redundant systems. Let $\mathbf{x}(t)$ denote the state of a plant and $\mathbf{u}(t)$ the applied control signal at time t. The state consists of the joint angles $\mathbf{q}$ and velocities $\dot{\mathbf{q}}$ of a robot, and the actuator control signals $\mathbf{u}$. If the system would be deterministic, we could express its dynamics as $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u})$, whereas in the presence of noise we write the dynamics as a stochastic differential equation

$$d\mathbf{x} = \mathbf{f}(\mathbf{x},\mathbf{u})dt + \mathbf{F}(\mathbf{x},\mathbf{u})d\boldsymbol{\omega}. \tag{1}$$

Here, $d\boldsymbol{\omega}$ is assumed to be Brownian motion noise, which is transformed by a possibly state- and control-dependent matrix $\mathbf{F}(\mathbf{x},\mathbf{u})$. We formally specify the problem of carrying out a (reaching) movement as follows: Given an initial state $\mathbf{x}_0$ at time $t = 0$, we seek a control sequence $\mathbf{u}(t)$ such that the system's state is $\mathbf{x}^*$ at time $t = T$. Stochastic optimal control theory approaches the problem by first specifying a cost function which is composed of (i) some evaluation $h(\mathbf{x}(T))$ of the final state, usually penalising deviations from the desired state $\mathbf{x}^*$, and (ii) the accumulated cost $c(t,\mathbf{x},\mathbf{u})$ of sending a control signal $\mathbf{u}$ at time t in state $\mathbf{x}$, typically penalising large motor commands. Introducing a policy $\boldsymbol{\pi}(t,\mathbf{x})$ for selecting $\mathbf{u}(t)$, we can write the expected cost of following that policy from time t as [28]

$$v^{\boldsymbol{\pi}}(t,\mathbf{x}(t)) = \left\langle h(\mathbf{x}(T)) + \int_t^T c(s,\mathbf{x}(s),\boldsymbol{\pi}(s,\mathbf{x}(s)))ds \right\rangle. \tag{2}$$

In OFC one then aims to find the policy $\boldsymbol{\pi}$ that minimises the total expected cost $v^{\boldsymbol{\pi}}(0,\mathbf{x}_0)$. Thus, in contrast to classical control, calculation of the trajectory (planning) and the control signal (execution) is handled in one go. Notably, optimal control provides a principled approach to resolve redundancy: Whereas redundant degrees of freedom are often a nuisance for kinematic path planning, in OFC redundancy can actually be exploited in order to decrease the cost.

If the dynamics $\mathbf{f}$ is linear in $\mathbf{x}$ and $\mathbf{u}$, the cost is quadratic, and the noise is Gaussian, the resulting so-called LQG or LQR[1] problem is convex and can be solved

[1] LQR stands for linear quadratic regulator and describes the optimal controller for linear systems and quadratic costs. LQG also includes an optimal state estimator (under the assumption of Gaussian noise), but because for linear systems estimation and control are independent of each other, LQR and LQG essentially compute the same control law.

analytically [23]. Finding an optimal control policy for nonlinear systems, in contrast, is a much harder challenge. Global solutions could be found in theory by applying dynamic programming methods [5] that are based on the Hamilton-Jacobi-Bellman equations. However, in their basic form these methods rely on a discretisation of the state and action space, an approach that is not viable for large DoF systems. Some research has been carried out on random sampling in a continuous state and action space [24], and it has been suggested that sampling can avoid the curse of dimensionality if the underlying problem is simple enough [2], as is the case if the dynamics and cost functions are very smooth.

A promising alternative to global OFC methods are approaches that compromise between open loop and closed loop optimisation and iteratively compute an optimal trajectory together with a locally valid feedback law. These trajectory-based methods are not directly subject to the curse of dimensionality but still yield locally optimal controllers. Differential dynamic programming (DDP) [9, 12] is a well-known successive approximation technique for solving nonlinear dynamic optimisation problems. This method uses second order approximations of the system dynamics and cost function to perform dynamic programming in the neighbourhood of a nominal trajectory. A more recent algorithm is the Iterative Linear Quadratic Regulator (ILQR) [16]. This algorithm uses iterative linearisation of the nonlinear dynamics around the nominal trajectory, and solves a locally valid LQR problem to iteratively improve the trajectory. However, ILQR is still deterministic and cannot deal with control constraints. A recent extension to ILQR, the Iterative Linear Quadratic Gaussian (ILQG) framework [28], allows to model nondeterministic dynamics by incorporating a Gaussian noise model. Furthermore it supports control constraints like non-negative muscle activations or upper control boundaries and therefore is well suited for the investigation of biologically inspired systems. The ILQG framework has been shown to be computationally significantly more efficient than DDP [16] and also has been previously tested on biologically inspired movement systems and therefore is the favourite approach for us to investigate further.

The ILQG algorithm starts with a time-discretised initial guess of an optimal control sequence and then iteratively improves it w.r.t. the performance criteria in v (eq. 2). From the initial control sequence $\bar{\mathbf{u}}^i$ at the i-iteration, the corresponding state sequence $\bar{\mathbf{x}}^i$ is retrieved using the deterministic forward dynamics $\mathbf{f}$ with a standard Euler integration $\bar{\mathbf{x}}^i_{k+1} = \bar{\mathbf{x}}^i_k + \Delta t\, \mathbf{f}(\bar{\mathbf{x}}^i_k, \bar{\mathbf{u}}^i_k)$. In a next step the discretised dynamics (eq. 1) are linearly approximated around $\bar{\mathbf{x}}^i_k$ and $\bar{\mathbf{u}}^i_k$.:

$$\delta\mathbf{x}_{k+1} = \left(\mathbf{I} + \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big|_{\bar{\mathbf{x}}_k}\right)\delta\mathbf{x}_k + \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\Big|_{\bar{\mathbf{u}}_k}\delta\mathbf{u}_k + \sqrt{\Delta t}\left(\mathbf{F}(\mathbf{u}_k) + \frac{\partial \mathbf{F}}{\partial \mathbf{u}}\Big|_{\bar{\mathbf{u}}_k}\delta\mathbf{u}_k\right)\boldsymbol{\xi}_k. \qquad (3)$$

Similarly to the linearised dynamics in (3) one can derive an approximate cost function which is quadratic in $\delta\mathbf{u}$ and $\delta\mathbf{x}$ (for details please see [28]). Both approximations are formulated as deviations of the current optimal trajectory $\delta\mathbf{x}^i_k = \mathbf{x}^i_k - \bar{\mathbf{x}}^i_k$ and $\delta\mathbf{u}^i_k = \mathbf{u}^i_k - \bar{\mathbf{u}}^i_k$ and therefore form a *local* LQG problem. This linear quadratic problem can be solved efficiently via a modified Ricatti-like set of equations. The

optimisation supports constraints for the control variable $\mathbf{u}$, such as lower and upper bounds. After the optimal control signal correction $\delta\bar{\mathbf{u}}^i$ has been obtained, it can be used to improve the current optimal control sequence for the next iteration using $\bar{\mathbf{u}}_k^{i+1} = \bar{\mathbf{u}}_k^i + \delta\bar{\mathbf{u}}^i$. At last $\bar{\mathbf{u}}_k^{i+1}$ is applied to the system dynamics (eq. 1) and the new total cost along the trajectory is computed. The algorithm stops once the cost v cannot be significantly decreased anymore. After convergence, ILQG returns an optimal control sequence $\bar{\mathbf{u}}$ and a corresponding optimal state sequence $\bar{\mathbf{x}}$ (i.e., trajectory). Along with the optimal open loop parameters $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$, ILQG produces a feedback matrix $\mathbf{L}$ which may serve as optimal feedback gains for correcting local deviations from the optimal trajectory on the plant.

Since the focus of this work is on utilising dynamics learning within ILQG, and its implications to adaptivity, we do not utilise an explicit noise model $\mathbf{F}$ for the sake of clarity of results. In fact it has been shown that a matching feedback control law is only marginally superior to one that is optimised for a deterministic system [28]. We also do not include any model for estimating the state, that is, we assume that noise-free measurements of the system are available (full observability). However an ILQG implementation for systems with partial observability has been been developed recently [17].

3 Adaptive Optimal Feedback Control

As mentioned earlier a major shortcoming of ILQG (and other OFC methods) is the dependence on an analytic form of the system dynamics, which often may be unknown or subject to change. We overcome this limitation by learning an adaptive internal model of the system dynamics using an online, supervised learning method. We consequently use the learned model to derive an ILQG formulation that is computationally efficient, reacts optimally to transient perturbations, and most notably adapts to systematic changes in plant dynamics. We name this algorithm *ILQG with learned dynamics (ILQG–LD)*.

The idea of learning the system dynamics in combination with iterative optimisations of trajectory or policy has been explored previously in the literature, e.g., for learning to swing up a pendulum [4] using some prior knowledge about the form of the dynamics. Similarly, Abeel et al. [1] proposed a hybrid reinforcement learning algorithm, where a policy and an internal model get subsequently updated from "real life" trials. In contrast to their method, however, we employ a second-order optimisation method, and we refine the control law solely from the internal model. To our knowledge, learning dynamics in conjunction with control optimisation has not been studied in the light of adaptability to changing plant dynamics.

From a biological point of view, enabling OFC to be adaptive would allow us to investigate the role of optimal control in human adaptation scenarios. Indeed, adaptation in humans, for example towards external perturbations, is a key property of human motion and is a very active area of research since nearly two decades [21, 22].

3.1 ILQG with Learned Dynamics (ILQG–LD)

In order to eliminate the need for an analytic dynamics model and to make ILQG adaptive, we wish to learn an approximation $\tilde{\mathbf{f}}$ of the real plant forward dynamics $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$. Assuming our model $\tilde{\mathbf{f}}$ has been coarsely pre-trained, for example by motor babbling, we can refine that model in an online fashion as shown in Fig. 1. For optimising and carrying out a movement, we have to define a cost function (where also the desired final state is encoded), the start state, and the number of discrete time steps because the ILQG algorithm in its current form requires a specified final time. Given an initial torque sequence $\bar{\mathbf{u}}_k^0$, the ILQG iterations can be carried out as described in the Section 2, but utilising the learned model $\tilde{\mathbf{f}}$. This yields a locally optimal control sequence $\bar{\mathbf{u}}_k$, a corresponding desired state sequence $\bar{\mathbf{x}}_k$, and feedback correction gain matrices $\mathbf{L}_k$. Denoting the plant's true state by $\mathbf{x}$, at each time step k, the feedback controller calculates the required correction to the control signal as $\delta\mathbf{u}_k = \mathbf{L}_k(\mathbf{x}_k - \bar{\mathbf{x}}_k)$. We then use the final control signal $\mathbf{u}_k = \bar{\mathbf{u}}_k + \delta\mathbf{u}_k$, the plant's state $\mathbf{x}_k$ and its change $d\mathbf{x}_k$ to update our internal forward model $\tilde{\mathbf{f}}$. As we show in Section 4, we can thus account for (systematic) perturbations and also bootstrap a dynamics model from scratch.

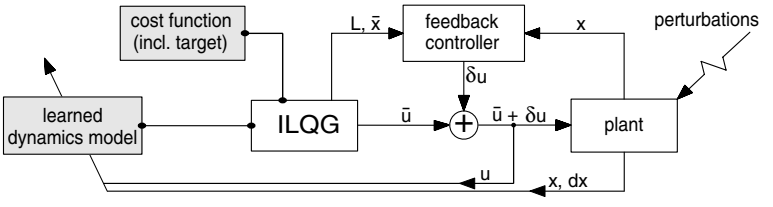


Fig. 1. Illustration of our ILQG–LD learning and control scheme.

3.2 Learning the Dynamics

Various machine learning algorithms could be applied to the robot control learning problem just mentioned. Global learning methods like sigmoid neural networks often suffer from the problem of negative interference, i.e., interference between learning in different parts of the input space when input data distributions are not uniform [20]. Local learning methods, in contrast, represent a function by using small simplistic patches - e.g. first order polynomials. The range of these local patches is determined by weighting kernels, and the number and parameters of the local kernels are adapted during learning to represent the non-linear function. Because any given training sample activates only a few patches, local learning algorithms are robust against global negative interference. This ensures the flexibility of the learned model towards changes in the dynamics properties of the arm (e.g. load, material wear, and different motion). Furthermore the domain of real-time robot control demands certain properties of a learning algorithm, namely fast learning rates and high computational efficiency for predictions and updates if the model is trained

incrementally. Locally Weighted Projection Regression (LWPR) has been shown to exhibit these properties, and to be very efficient for incremental learning of non-linear models in high dimensions [30].

During LWPR training, the parameters of the local models (locality and fit) are updated using incremental Partial Least Squares, and local models can be pruned or added on an as-need basis, for example, when training data is generated in previously unexplored regions. Usually the areas of validity (also termed its receptive field) of each local model are modelled by Gaussian kernels, so their activation or response to a query vector $\mathbf{z} = (\mathbf{x}^T, \mathbf{u}^T)^T$ (combining the *state* and *control* inputs of the forward dynamics $\mathbf{f}$) is given by

$$w_k(\mathbf{z}) = \exp\left(-\frac{1}{2}(\mathbf{z}-\mathbf{c}_k)^T \mathbf{D}_k (\mathbf{z}-\mathbf{c}_k)\right), \tag{4}$$

where $\mathbf{c}_k$ is the centre of the k^{th} linear model and $\mathbf{D}_k$ is its distance metric. Treating each output dimension[2] separately for notational convenience, and ignoring the details about the underlying PLS computations [14], the regression function can be written as

$$\tilde{f}(\mathbf{z}) = \frac{1}{W}\sum_{k=1}^{K} w_k(\mathbf{z})\psi_k(\mathbf{z}), \quad W = \sum_{k=1}^{K} w_k(\mathbf{z}), \tag{5}$$

$$\psi_k(\mathbf{z}) = b_k^0 + \mathbf{b}_k^T(\mathbf{z}-\mathbf{c}_k), \tag{6}$$

where b_k^0 and $\mathbf{b}_k$ denote the offset and slope of the k-th model, respectively.

LWPR learning has the desirable property that it can be carried out online, and moreover, the learned model can be adapted to changes in the dynamics in real-time. A forgetting factor λ [30], which balances the trade-off between preserving what has been learned and quickly adapting to the non-stationarity, can be tuned to the expected rate of external changes. In order to provide some insight, LWPR internally uses update rules within each receptive field of the form $E_{new} = \lambda \cdot E_{old} + w \cdot e_{cur}$. In this example, E is the sufficient statistics for the squared prediction error, and e_{cur} is the error from the current training sample alone, but the same principle applies for other quantities such as the correlation between input and output data. In this way, after N updates to a receptive field, the original value of the sufficient statistics has been down-weighted (or forgotten) by a factor of λ^N. As we will see later, the factor λ can be used to model biologically realistic adaptive behaviour to external force-fields.

3.3 Reducing the Computational Cost

So far, we have shown how the problem of unknown or changing system dynamics can be addressed within ILQG–LD. Another important issue to discuss is the

[2] In the case of learning forward dynamics, the target values are the joint accelerations. We effectively learn a separate model for each joint.

computational complexity. The ILQG framework has been shown to be the most effective locally optimal control method in terms of convergence speed and accuracy [15]. Nevertheless the computational cost of ILQG remains daunting even for simple movement systems, preventing their application to real-time optimal motion planning for large DoF systems. A large part of the computational cost arises from the linearisation of the system dynamics, which involves repetitive calculation of the system dynamics' derivatives $\partial\mathbf{f}/\partial\mathbf{x}$ and $\partial\mathbf{f}/\partial\mathbf{u}$. When the analytical form of these derivatives is not available, they must be approximated using finite differences. The computational cost of such an approximation scales linearly with the sum of the dimensionalities of $\mathbf{x} = (\mathbf{q};\dot{\mathbf{q}})$ and $\mathbf{u} = \boldsymbol{\tau}$ (i.e., $3N$ for an N DoF joint torque controlled robot). In simulations, our analysis show that for the 2 DoF manipulator, 60% of the total ILQG computations can be attributed to finite differences calculations. For a 6 DoF arm, this rises to 80%.

Within our ILQG–LD scheme, we can avoid finite difference calculations and rather use the analytic derivatives of the learned model, as has similarly been proposed in [3]. Differentiating the LWPR predictions (5) with respect to $\mathbf{z} = (\mathbf{x};\mathbf{u})$ yields terms

$$\frac{\partial \tilde{f}(\mathbf{z})}{\partial \mathbf{z}} = \frac{1}{W}\sum_k \left(\frac{\partial w_k}{\partial \mathbf{z}}\psi_k(\mathbf{z}) + w_k\frac{\partial \psi_k}{\partial \mathbf{z}}\right) - \frac{1}{W^2}\sum_k w_k(\mathbf{z})\psi_k(\mathbf{z})\sum_l \frac{\partial w_l}{\partial \mathbf{z}} \tag{7}$$

$$= \frac{1}{W}\sum_k \left(-\psi_k w_k \mathbf{D}_k(\mathbf{z}-\mathbf{c}_k) + w_k\mathbf{b}_k\right) + \frac{\tilde{f}(\mathbf{z})}{W}\sum_k w_k\mathbf{D}_k(\mathbf{z}-\mathbf{c}_k) \tag{8}$$

for the different rows of the Jacobian matrix $\begin{pmatrix}\partial\tilde{\mathbf{f}}/\partial\mathbf{x}\\ \partial\tilde{\mathbf{f}}/\partial\mathbf{u}\end{pmatrix} = \frac{\partial}{\partial\mathbf{z}}(\tilde{f}_1,\tilde{f}_2,\ldots\tilde{f}_N)^T$.

Table 1 illustrates the computational gain (mean CPU time per ILQG iteration) across 3 test manipulators – highlighting added benefits for more complex systems. On a notebook running at 1.6 GHz, the average CPU times for a *complete* ILQG trajectory using the analytic method are 0.8 sec (2 DoF), 1.9 sec (6 DoF), and 9.8 sec (12 DoF), respectively. Note that LWPR is a highly parallelisable algorithm: Since the local models learn independently of each other, the respective computations can be distributed across multiple processors or processor cores, which can yield a further significant performance gain [14].

Table 1. CPU time for one ILQG–LD iteration (sec).

	finite differences	**analytic Jacobian**	**improvement factor**
2 DoF	0.438	0.193	2.269
6 DoF	4.511	0.469	9.618
12 DoF	29.726	1.569	18.946

4 Evaluation

In this section we evaluate ILQG–LD in several setups with increasing complexity. We start with joint torque controlled manipulators setups first, which will be analysed under stationary and non-stationary conditions. We then present ILQG–LD results from an antagonistic humanoid arm model which embodies the challenge of large redundancies in the dynamics domain.

All simulations are performed with the Matlab Robotics Toolbox [7]. This simulation model computes the non-linear plant dynamics using standard equations of motion. For an N-DoF manipulator the joint torques $\boldsymbol{\tau}$ are given by

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{b}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}), \tag{9}$$

where $\mathbf{q}$ and $\dot{\mathbf{q}}$ are the joint angles and joint velocities respectively; $\mathbf{M}(\mathbf{q})$ is the N-dimensional symmetric joint space inertia matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ accounts for Coriolis and centripetal effects, $\mathbf{b}(\dot{\mathbf{q}})$ describes the viscous and Coulomb friction in the joints, and $\mathbf{g}(\mathbf{q})$ defines the gravity loading depending on the joint angles $\mathbf{q}$ of the manipulator.

We study movements for a fixed motion duration of one second, which we discretise into $K = 100$ steps ($\Delta t = 0.01$s). The manipulator starts at an initial position $\mathbf{q}_0$ and reaches towards a target $\mathbf{q}_{tar}$. During movement we wish to minimise the energy consumption of the system. We therefore use the cost function

$$v = w_p \,|\, \mathbf{q}_K - \mathbf{q}_{tar} \,|^2 + w_v \,|\, \dot{\mathbf{q}}_K \,|^2 + w_e \sum_{k=0}^{K} |\, \mathbf{u}_k \,|^2 \Delta t, \tag{10}$$

where the factors for the target position accuracy (w_p), for the zero end-point velocity (w_v), and for the energy term (w_e) weight the importance of each component. We compare the control results of ILQG–LD and ILQG with respect to the number of iterations, the end point accuracy and the generated costs. In this paper we will refer to *cost* as total cost defined in (10) and to *running cost* to the energy consumption only, i.e., the summation term in (10).

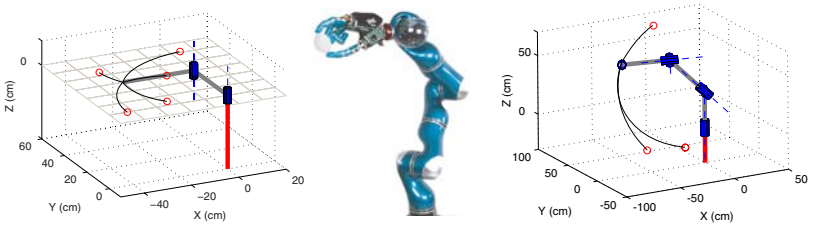


Fig. 2. Two different joint-torque controlled manipulator models with selected targets (circles) and ILQG generated trajectories as benchmark data. All models are simulated using the Matlab Robotics Toolbox. Left: 2 DoF planar manipulator model; Middle: picture of the Kuka Light-Weight Robot arm (LWR); Right: Simulated 6 DoF LWR model (without hand).

4.1 Planar Arm with 2 Torque-Controlled Joints

The first setup (Fig. 2 left) is a horizontally planar 2 DoF manipulator similar to the one used in [28]. The arm is controlled by directly commanding joint torques. This low DoF system is ideal for performing extensive (quantitative) comparison studies and to test the manipulator under controlled perturbations and force fields during planar motion.

4.1.1 Stationary Dynamics

First, we compared the characteristics of ILQG–LD and ILQG (both operated in open loop mode) in the case of stationary dynamics without any noise in the 2 DoF plant. Fig. 3 shows three trajectories generated by learned models of different predictive quality, which is reflected by the different normalised means square errors (nMSE) on test data. 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 and $\tilde{y}$ represents the LWPR predictions. The nMSE takes into account the output distribution of the data (variance σ_y^2 in the data) and therefore produces a "dimensionless" error measure. As one would expect, the quality of the model plays an important role for the final cost, the number of ILQG–LD iterations, and the final target distances (cf. the table within Fig. 3). For the final learned model, we observe a striking resemblance with the analytic ILQG performance.

Next, we carried out a reaching task to 5 reference targets covering a wide operating area of the planar arm. To simulate control dependent noise, we contaminated the commands $\mathbf{u}$ just before feeding them into the plant, using Gaussian noise with 50% of the variance of the signal $\mathbf{u}$. We then generated motor commands to move the system towards the targets, both with and without the feedback controller. As expected, closed loop control (utilising gain matrices $\mathbf{L}_k$) is superior to open loop operation regarding reaching accuracy. Fig. 4 depicts the performance of ILQG–LD and ILQG under both control schemes. Averaged over all trials, both methods show similar endpoint variances and behaviour which is statistically indistinguishable.

ILQG–LD	**(L)**	**(M)**	**(H)**	**ILQG**
No. of training points	111	146	276	–
Prediction error (nMSE)	0.80	0.50	0.001	–
Iterations	19	17	5	4
Cost	2777.36	1810.20	191.91	192.07
Eucl. target distance (cm)	19.50	7.20	0.40	0.01

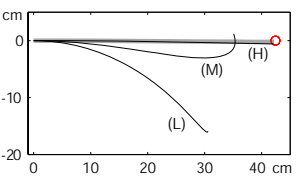
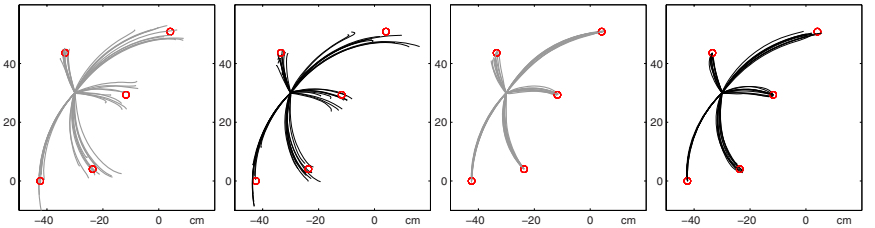


Fig. 3. Behaviour of ILQG–LD for learned models of different quality: (L)-Low, (M)-Medium, (H)-High. Right: Trajectories in task space produced by ILQG–LD (black lines) and ILQG (grey line).



open loop ILQG
$d = 2.71 \pm 1.84$ cm

open loop ILQG–LD
$d = 3.20 \pm 1.60$ cm

closed loop ILQG
$d = 0.19 \pm 0.001$ cm

closed loop ILQG–LD
$d = 0.18 \pm 0.001$ cm

Fig. 4. Illustration of the target reaching performances for the planar 2 DoF in the presence of strong control dependent noise, where d represents the average Euclidean distance to the five reference targets.

4.1.2 Non-stationary Dynamics

A major advantage of ILQG–LD is that it does not rely on an accurate analytic dynamics model; consequently, it can adapt *on the fly* to external perturbations and to changes in the plant dynamics that may result from altered morphology or wear and tear. We carried out adaptive reaching experiments in our simulation similar to the human manipulandum experiments in [21]. First, we generated a constant unidirectional force field (FF) acting perpendicular to the reaching movement (see Fig. 5). Using the ILQG–LD models from the previous experiments, the manipulator gets strongly deflected when reaching for the target because the learned dynamics model cannot account for the *spurious* forces. However, using the resultant deflected trajectory (100 data points) as training data, updating the dynamics model online brings the manipulator nearer to the target with each new trial. We repeated this procedure until the ILQG–LD performance converged successfully. At that point, the internal model successfully accounts for the change in dynamics caused by the FF. Then, removing the FF results in the manipulator overshooting to the other side, compensating for a non-existing FF. Just as before, we re-adapted the dynamics online over repeated trials.

Fig. 5 summarises the results of the sequential adaptation process just described. The closed loop control scheme clearly converges faster than the open loop scheme, which is mainly due to the OFC's desirable property of always correcting the system towards the target. Therefore, it produces more relevant dynamics training data. Furthermore, we can accelerate the adaptation process significantly by tuning the forgetting factor λ, allowing the learner to weight the importance of new data more strongly [30]. A value of $\lambda = 0.95$ produces significantly faster adaptation results than the default of $\lambda = 0.999$. As a follow-up experiment, we made the force field dependent on the velocity $\mathbf{v}$ of the end-effector, i.e. we applied a force

$$\mathbf{F} = \mathbf{B}\mathbf{v}, \quad \text{with} \quad \mathbf{B} = \begin{pmatrix} 0 & 50 \\ -50 & 0 \end{pmatrix} Nm^{-1}s \tag{11}$$

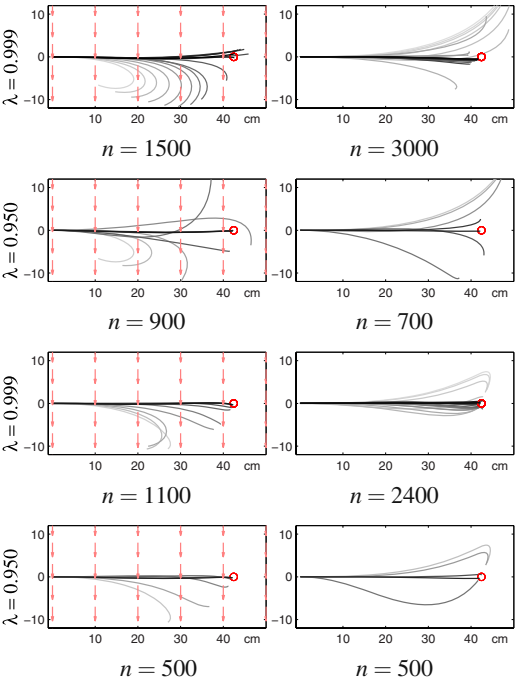


Fig. 5. Illustration of adaptation experiments for open loop (rows 1,2) and closed loop (rows 3,4) ILQG–LD. Arrows depict the presence of a (constant) force field; n represents the number of training points required to successfully update the internal LWPR dynamics model. Darker lines indicate better trained models, corresponding to later trials in the adaption process.

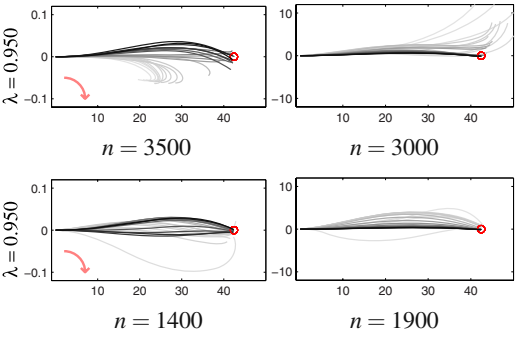


Fig. 6. Adaptation to a velocity-dependent force field (as indicated by the bent arrow) and re-adaptation after the force field is switched off (right column). Top: open loop. Bottom: closed loop.

to the end-effector. The results are illustrated in Fig. 6: For the more complex FF, more iterations are needed in order to adapt the model, but otherwise ILQG–LD shows a similar behaviour as for the constant FF. Interestingly, the overshoot behaviour depicted in Fig. 5 and 6 has been observed similarly in human adaptation experiments where it was referred to as "after effects" [21]. We believe this to be an interesting insight for future investigation of ILQG–LD and its role in modeling sensorimotor adaptation data in the (now extensive) human reach experimental paradigm [22].

4.2 Anthropomorphic 6 DoF Robot Arm

Our next experimental setup is a 6 DoF manipulator (Fig. 2, right), the physical parameters (i.e., link inertia, mass, etc.) of which are a faithful model of the first 6 links of the *Kuka Light-Weight Robot* (LWR).

Using this arm, we studied reaching targets specified in *Cartesian* coordinates $\mathbf{r} \in \mathbb{R}^3$ in order to highlight the redundancy resolution capability and trial-to-trial variability in large DoF systems. We set up the cost function (cf. eq. 10) as

$$v = w_p \,|\, \mathbf{r}(\mathbf{q}_K) - \mathbf{r}_{tar} \,|^2 + w_v \,|\, \dot{\mathbf{q}}_K \,|^2 + w_e \sum_{k=0}^{K} |\, \mathbf{u}_k \,|^2 \Delta t, \tag{12}$$

where $\mathbf{r}(\mathbf{q})$ denotes the end-effector position as calculated from forward kinematics. It should be noted that for the specific kinematic structure of this arm, this 3D position depends only on the first 4 joint angles. Joints 5 and 6 only change the orientation of the end-effector[3], which does not play a role in our reaching task and correspondingly in the cost function. In summary, our arm has *one redundant* and further *two irrelevant* degrees of freedom for this task.

Table 2. Comparison of the performance of ILQG–LD and ILQG for controlling a 6 DoF robot arm. We report the number of iterations required to compute the control law, the average running cost, and the average Euclidean distance **d** to the three reference targets.

	ILQG			ILQG–LD		
Targets	**Iter.**	**Run. cost**	**d (cm)**	**Iter.**	**Run. cost**	**d (cm)**
(a)	51	**18.50** $\pm$ 0.13	**2.63** $\pm$ 1.63	51	**18.32** $\pm$ 0.55	**1.92** $\pm$ 1.03
(b)	61	**18.77** $\pm$ 0.25	**1.32** $\pm$ 0.69	99	**18.65** $\pm$ 1.61	**0.53** $\pm$ 0.20
(c)	132	**12.92** $\pm$ 0.04	**1.75** $\pm$ 1.30	153	**12.18** $\pm$ 0.03	**2.00** $\pm$ 1.02

Similar to the 2 DoF experiments, we bootstrapped a forward dynamics model through extensive data collection (i.e., motor babbling). Next, we used ILQG–LD (closed loop, with noise) to train our dynamics model online until it converged to stable reaching behaviour. Fig. 7 depicts reaching trials, 20 for each reference target,

[3] The same holds true for the 7th joint of the original LWR arm.

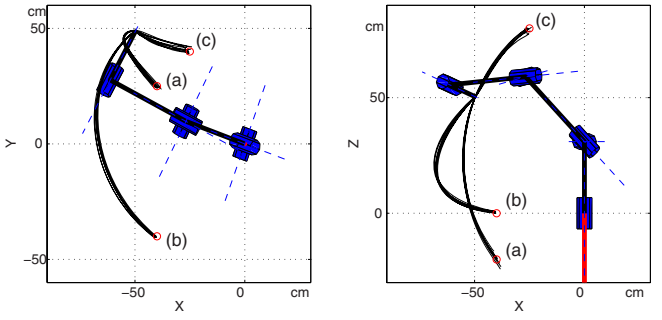


Fig. 7. Illustration of the trial-to-trial variability of the 6-DoF arm when reaching towards target (a,b,c). Left: top-view, right: side-view.

using ILQG–LD with the final learned model. Table 2 quantifies the performance. The targets are reached reliably and no statistically significant differences can be spotted between ILQG–LD and ILQG. An investigation of the trials in *joint angle* space also shows similarities. Fig. 8 depicts the 6 joint angle trajectories for the 20 reaching trials towards target (c). Please note the high variance of the joint angles especially for the irrelevant joints 5 and 6, which nicely show that task irrelevant errors are not corrected unless they adversely affect the task (minimum intervention principle of OFC). Moreover, the joint angle variances (trial-to-trial variability) between the ILQG–LD and ILQG trials are in a similar range, indicating an equivalent corrective behaviour – the shift of the absolute variances can be explained by the slight mismatch between the learned and analytical dynamics. We can conclude from our results that ILQG–LD scales up very well to 6 DoF, not suffering from any losses in terms of accuracy, cost or convergence behaviour. Furthermore, its computational cost is significantly lower than the one of ILQG.

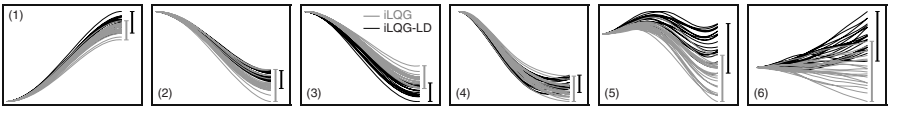


Fig. 8. Illustration of the trial-to-trial variability in the joint angles (1–6) over time when reaching towards target (c). Grey lines indicate ILQG, black lines stem from ILQG–LD.

4.3 Antagonistic Planar Arm

In order to analyse ILQG–LD in a dynamically redundant scenario, we studied a two DoF planar human arm model, which is actuated by four single-joint and two double-joint antagonistic muscles (Fig. 9 left). The arm model described in this section is based on [13]. Although kinematically simple, the system is over-actuated and therefore an interesting testbed for our control scheme, because large

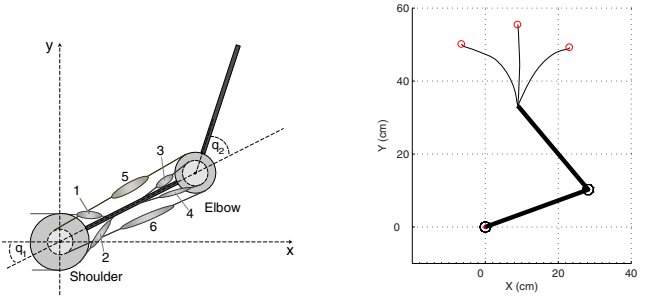


Fig. 9. Left: Human arm model with 6 muscles (adapted from [13]). Right: Same arm model with selected targets (circles) and ILQG generated trajectories as benchmark data. The physics of the model is simulated using the Matlab Robotics Toolbox [7].

redundancies in the dynamics have to be resolved. The dimensionality of the control signals makes adaptation processes (e.g., to external force fields) quite demanding. Indeed this arm poses a harder learning problem than the 6-DoF manipulator of the previous section, because the muscle-based actuation makes the dynamics less linear.

As before the dynamics of the arm is in part based on standard equations of motion, given by

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}. \tag{13}$$

Given the antagonistic muscle-based actuation, we cannot command joint torques directly, but rather we have to calculate effective torques from the muscle activations $\mathbf{u}$. For the present model the corresponding transfer function is given by

$$\boldsymbol{\tau}(\mathbf{q},\dot{\mathbf{q}},\mathbf{u}) = -\mathbf{A}(\mathbf{q})^{\mathbf{T}}\mathbf{T}(\mathbf{l},\dot{\mathbf{l}},\mathbf{u}), \tag{14}$$

where $\mathbf{A}$ represents the moment arm. For simplicity, we assume $\mathbf{A}$ to be constant and independent of the joint angles $\mathbf{q}$:

$$\mathbf{A}(\mathbf{q}) = \mathbf{A} = \begin{pmatrix} a_1 & a_2 & 0 & 0 & a_5 & a_6 \\ 0 & 0 & a_3 & a_4 & a_7 & a_8 \end{pmatrix}^T. \tag{15}$$

The muscle lengths $\mathbf{l}$ depend on the joint angles $\mathbf{q}$ through the affine relationship $\mathbf{l} = \mathbf{l}_m - \mathbf{A}\mathbf{q}$, which also implies $\dot{\mathbf{l}} = -\mathbf{A}\dot{\mathbf{q}}$. The term $\mathbf{T}(\mathbf{l},\dot{\mathbf{l}},\mathbf{u})$ in (14) denotes the muscle tension, for which we follow the Kelvin-Voight model [19] and define:

$$\mathbf{T}(\mathbf{l},\dot{\mathbf{l}},\mathbf{u}) = \mathbf{K}(\mathbf{u})\big(\mathbf{l}_r(\mathbf{u}) - \mathbf{l}\big) - \mathbf{B}(\mathbf{u})\dot{\mathbf{l}}. \tag{16}$$

Here, $\mathbf{K}(\mathbf{u})$, $\mathbf{B}(\mathbf{u})$, and $\mathbf{l}_r(\mathbf{u})$ denote the muscle stiffness, the muscle viscosity and the muscle rest length, respectively. Each of these terms depends linearly on the motor commands $\mathbf{u}$, as given by

$$\mathbf{K}(\mathbf{u}) = diag(\mathbf{k}_0 + k\mathbf{u}), \qquad \mathbf{B}(\mathbf{u}) = diag(\mathbf{b}_0 + b\mathbf{u}), \qquad \mathbf{l}_r(\mathbf{u}) = \mathbf{l}_0 + r\mathbf{u}. \tag{17}$$

The elasticity coefficient k, the viscosity coefficient b, and the constant r are given from the muscle model. The same holds true for $\mathbf{k}_0$, $\mathbf{b}_0$, and $\mathbf{l}_0$, which are the intrinsic elasticity, viscosity and rest length for $\mathbf{u} = \mathbf{0}$, respectively. For the exact values of these coefficients please refer to [13]. ILQG has been applied previously to similar antagonistic arm models, that are slightly more complex. Most notably, non-constant moment arms $\mathbf{A}(\mathbf{q})$, stochastic control signals, and a muscle activation dynamics which increase the dimensionality of the state space have been used [15].

Please note that in contrast to standard torque-controlled robots, in our arm model the dynamics (13) is *not* linear in the control signals, since $\mathbf{u}$ enters (16) quadratically. We follow the same cost function as before (eq. 10) and the same fixed motion duration of one second. Here we discretise the time into $K = 50$ steps ($\Delta t = 0.02$s).

4.3.1 Stationary Dynamics

In order to make ILQG–LD converge for our three reference targets we coarsely pretrained our LWPR model with a focus on a wide coverage of the workspace. The training data are given as tuples consisting of $(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u})$ as inputs (10 dimensions in total), and the observed joint accelerations $\ddot{\mathbf{q}}$ as the desired two-dimensional output. We stopped training once the normalised mean squared error (nMSE) in the predictions reached ≤ 0.005. At this point LWPR had seen $1.2 \cdot 10^6$ training data points and had acquired 852 receptive fields, which is in accordance with the previously discussed high non-linearity of the plant dynamics.

We carried out a reaching task to the 3 reference targets (Fig. 9, right) using the feedback controller (feedback gain matrix $\mathbf{L}$) that falls out of ILQG(-LD). To compare the stability of the control solution, we simulated control dependent noise by contaminating the muscle commands $\mathbf{u}$ just before feeding them into the plant. We applied Gaussian noise with 50% of the variance of the signal $\mathbf{u}$.

Fig. 10 depicts the generated control signals and the resulting performance of ILQG–LD and ILQG over 20 reaching trials per target. Both methods show similar

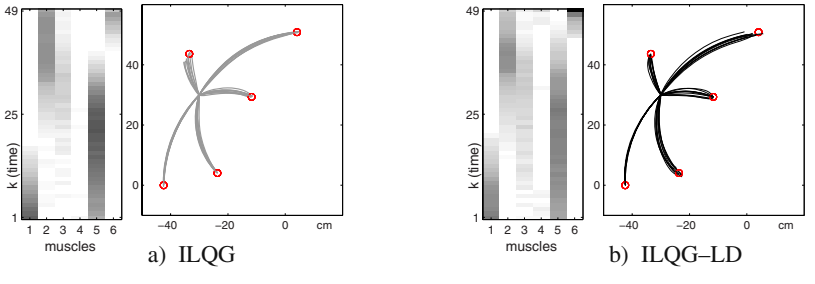


Fig. 10. Illustration of an optimised control sequence (left) and resulting trajectories (right) when using a) the known analytic dynamics model and b) the LWPR model learned from data. The control sequences (left target only) for each muscle (1–6) are drawn from bottom to top, with darker grey levels indicating stronger muscle activation.

Table 3. Comparison of the performance of ILQG–LD and ILQG with respect to the number of iterations required to compute the control law, the average running cost, and the average Euclidean distance to the three reference targets (left, center, right).

	ILQG			ILQG–LD		
Targets	**Iter.**	**Run. cost**	**d (cm)**	**Iter.**	**Run. cost**	**d (cm)**
Center	19	**0.0345** $\pm$ 0.0060	**0.11** $\pm$ 0.07	14	**0.0427** $\pm$ 0.0069	**0.38** $\pm$ 0.22
Left	40	**0.1873** $\pm$ 0.0204	**0.10** $\pm$ 0.06	36	**0.1670** $\pm$ 0.0136	**0.21** $\pm$ 0.16
Right	41	**0.1858** $\pm$ 0.0202	**0.57** $\pm$ 0.49	36	**0.1534** $\pm$ 0.0273	**0.19** $\pm$ 0.12

endpoint variances and trajectories which are in close match. As can be seen from the visualisation of the control sequences, antagonistic muscles (i.e., muscle pairs 1/2, 3/4, and 5/6 in Fig. 9, left) are never activated at the same time. This is a direct consequence of the cost function, which penalises co-contraction as a waste of energy. Table 3 quantifies the control results of ILQG–LD and ILQG for each target with respect to the number of iterations, the generated running costs and the end point accuracy.

4.3.2 Adaptation Results

As before we carried out adaptive reaching experiments (towards the center target) and we generated a constant unidirectional force field (FF) acting perpendicular to the reaching movement (see Fig. 11). Using the ILQG–LD model from the previous experiment, the manipulator gets strongly deflected when reaching for the target because the learned dynamics model cannot yet account for the "spurious" forces. However, using the resultant deflected trajectory as training data, updating the dynamics model online brings the manipulator nearer to the target with each new trial. In order to produce enough training data, as is required for a successful adaptation, we generated 20 slightly jittered versions of the optimised control sequences, ran these on the plant, and trained the LWPR model with the resulting 50 samples each. We repeated this procedure until the ILQG–LD performance converged successfully, which was the case after 27000 training samples. At that point, the internal model successfully accounted for the change in dynamics caused by the FF. Then, we switched off the FF while continuing to use the adapted LWPR model. This resulted in an overshooting of the manipulator to the other side, trying to compensate for non-existing forces. Just as before, we re-adapted the dynamics online over repeated trials. The arm reached the target again after 7000 training points. One should note that compared to the initial *global* motor babbling, where we required $1.2 \cdot 10^6$ training data points, for the *local* (re-)adaptation we need only a fraction of the data points.

Fig. 11 summarises the results of the sequential adaptation process just described. Please note how the optimised *adapted* control sequence contains considerably stronger activations of the extensor muscles responsible for pulling the arm to the right (denoted by "2" and "6" in Fig. 9), while still exhibiting practically no co-contraction.

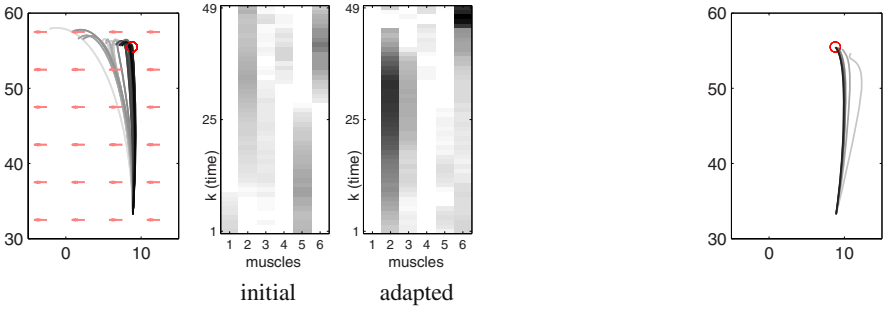


Fig. 11. Left: Adaptation to a unidirectional constant force field (indicated by the arrows). Darker lines indicate better trained models. In particular, the left-most trajectory corresponds to the "initial" control sequence, which was calculated using the LWPR model (from motor babbling) *before* the adaptation process. The fully "adapted" control sequence results in a nearly straight line reaching movement. Right: Resulting trajectories during re-adaptation after the force field has been switched off (i.e., after effects).

5 Discussion

In this work we introduced ILQG–LD, a method that realises adaptive optimal feedback control by incorporating a learned dynamics model into the ILQG framework. Most importantly, we carried over the favourable properties of ILQG to more realistic control problems where the analytic dynamics model is often unknown, difficult to estimate accurately or subject to changes. As with ILQG control, redundancies are implicitly resolved by the OFC framework through a cost function, eliminating the need for a separate trajectory planner and inverse kinematics/dynamics computation.

Utilising the derivatives (8) of the learned dynamics model $\tilde{\mathbf{f}}$ avoids expensive finite difference calculations during the dynamics linearisation step of ILQG. This significantly reduces the computational complexity, allowing the framework to scale to larger DoF systems. We empirically showed that ILQG–LD performs reliably in the presence of noise and that it is adaptive with respect to systematic changes in the dynamics; hence, the framework has the potential to provide a unifying tool for modelling (and informing) non-linear sensorimotor adaptation experiments even under complex dynamic perturbations. As with ILQG control, redundancies are implicitly resolved by the OFC framework through a cost function, eliminating the need for a separate trajectory planner and inverse kinematics/dynamics computation.

Our future work will concentrate on implementing the ILQG–LD framework on an anthropomorphic hardware – this will not only explore an alternative control paradigm, but will also provide the only viable and principled control strategy for the biomorphic *variable stiffness* based highly redundant actuation system that we are currently developing. Indeed, exploiting this framework for understanding OFC and its link to biological motor control is another very important strand.

References

1. Abbeel, P., Quigley, M., Ng, A.Y.: Using inaccurate models in reinforcement learning. In: Proc. Int. Conf. on Machine Learning (ICML), pp. 1–8 (2006)
2. Atkeson, C.G.: Randomly sampling actions in dynamic programming. In: Proc. Int. Symp. on Approximate Dynamic Programming and Reinforcement Learning, pp. 185–192 (2007)
3. Atkeson, C.G., Moore, A., Schaal, S.: Locally weighted learning for control. AI Review 11, 75–113 (1997)
4. Atkeson, C.G., Schaal, S.: Learning tasks from a single demonstration. In: Proc. Int. Conf. on Robotics and Automation (ICRA), Albuquerque, New Mexico, pp. 1706–1712 (1997)
5. Bertsekas, D.P.: Dynamic programming and optimal control. Athena Scientific, Belmont (1995)
6. Conradt, J., Tevatia, G., Vijayakumar, S., Schaal, S.: On-line learning for humanoid robot systems. In: Proc. Int. Conf. on Machine Learning (ICML), pp. 191–198 (2000)
7. Corke, P.I.: A robotics toolbox for MATLAB. IEEE Robotics and Automation Magazine 3(1), 24–32 (1996)
8. D'Souza, A., Vijayakumar, S., Schaal, S.: Learning inverse kinematics. In: Proc. Int. Conf. on Intelligence in Robotics and Autonomous Systems (IROS), Hawaii, pp. 298–303 (2001)
9. Dyer, P., McReynolds, S.: The Computational Theory of Optimal Control. Academic Press, New York (1970)
10. Flash, T., Hogan, N.: The coordination of arm movements: an experimentally confirmed mathematical model. Journal of Neuroscience 5, 1688–1703 (1985)
11. Grebenstein, M., van der Smagt, P.: Antagonism for a highly anthropomorphic hand-arm system. Advanced Robotics 22(1), 39–55 (2008)
12. Jacobson, D.H., Mayne, D.Q.: Differential Dynamic Programming. Elsevier, New York (1970)
13. Katayama, M., Kawato, M.: Virtual trajectory and stiffness ellipse during multijoint arm movement predicted by neural inverse model. Biological Cybernetics 69, 353–362 (1993)
14. Klanke, S., Vijayakumar, S., Schaal, S.: A library for locally weighted projection regression. Journal of Machine Learning Research 9, 623–626 (2008)
15. Li, W.: Optimal Control for Biological Movement Systems. PhD dissertation, University of California, San Diego (2006)
16. Li, W., Todorov, E.: Iterative linear-quadratic regulator design for nonlinear biological movement systems. In: Proc. 1st Int. Conf. Informatics in Control, Automation and Robotics (2004)
17. Li, W., Todorov, E.: Iterative linearization methods for approximately optimal control and estimation of non-linear stochastic system. International Journal of Control 80(9), 14391–14453 (2007)
18. Nguyen-Tuong, D., Peters, J., Seeger, M., Schoelkopf, B.: Computed torque control with nonparametric regressions techniques. In: American Control Conference (2008)
19. Özkaya, N., Nordin, M.: Fundamentals of biomechanics: equilibrium, motion, and deformation. Van Nostrand Reinhold, New York (1991)
20. Schaal, S.: Learning Robot Control. In: The handbook of brain theory and neural networks, pp. 983–987. MIT Press, Cambridge (2002)
21. Shadmehr, R., Mussa-Ivaldi, F.A.: Adaptive representation of dynamics during learning of a motor task. The Journal of Neurosciene 14(5), 3208–3224 (1994)

22. Shadmehr, R., Wise, S.P.: The Computational Neurobiology of Reaching and Ponting. MIT Press, Cambridge (2005)
23. Stengel, R.F.: Optimal control and estimation. Dover Publications, New York (1994)
24. Thrun, S.: Monte carlo POMDPs. In: Advances in Neural Information Processing Systems (NIPS), pp. 1064–1070 (2000)
25. Todorov, E.: Optimality principles in sensorimotor control. Nature Neuroscience 7(9), 907–915 (2004)
26. Todorov, E., Jordan, M.: Optimal feedback control as a theory of motor coordination. Nature Neuroscience 5, 1226–1235 (2002)
27. Todorov, E., Jordan, M.: A minimal intervention principle for coordinated movement. In: Advances in Neural Information Processing Systems (NIPS), pp. 27–34. MIT Press, Cambridge (2003)
28. Todorov, E., Li, W.: A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems. In: Proc. of the American Control Conference (2005)
29. Uno, Y., Kawato, M., Suzuki, R.: Formation and control of optimal trajectories in human multijoint arm movements: minimum torque-change model. Biological Cybernetics 61, 89–101 (1989)
30. Vijayakumar, S., D'Souza, A., Schaal, S.: Incremental online learning in high dimensions. Neural Computation 17, 2602–2634 (2005)
31. Vijayakumar, S., D'Souza, A., Shibata, T., Conradt, J., Schaal, S.: Statistical learning for humanoid robots. Autonomous Robots 12(1), 55–69 (2002)
32. Wolf, S., Hirzinger, G.: A new variable stiffness design: Matching requirements of the next robot generation. In: Proc. Int. Conf. on Robotics and Automation (ICRA), pp. 1741–1746 (2008)