In: Proc. Fifth Intl. Conf. on Informatics in Control, Automation and Robotics (ICINCO '08), Madeira, Portugal (2008).

OPTIMAL CONTROL WITH ADAPTIVE INTERNAL DYNAMICS MODELS

Djordje Mitrovic, Stefan Klanke, and Sethu Vijayakumar
Institute of Perception, Action and Behavior, School of Informatics, University of Edinburgh, Edinburgh, UK
d.mitrovic@ed.ac.uk, s.klanke@ed.ac.uk, sethu.vijayakumar@ed.ac.uk

Keywords: Learning dynamics, optimal control, adaptive control, robot simulation

Abstract: Optimal feedback control has been proposed as an attractive movement generation strategy in goal reaching tasks for anthropomorphic manipulator systems. The optimal feedback control law for systems with non-linear dynamics and non-quadratic costs can be found by iterative methods, such as the iterative Linear Quadratic Gaussian (iLQG) algorithm. So far this framework relied on an analytic form of the system dynamics, which may often be unknown, difficult to estimate for more realistic control systems or may be subject to frequent systematic changes. In this paper, we present a novel combination of learning a forward dynamics model within the iLQG framework. Utilising such adaptive internal models can compensate for complex dynamic perturbations of the controlled system 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

We address the problem related to control of movement in large degree of freedom (DoF) anthropomorphic manipulators, with specific emphasis on (target) reaching tasks. This is challenging mainly due to the large redundancies that such systems exhibit. For example, a controller has to make a choice between many different possible trajectories (kinematics) and a multitude of applicable motor commands (dynamics) for achieving a particular task. How do we resolve this redundancy?

Optimal control theory (Stengel, 1994) answers this question by postulating that a particular choice is made because it is the optimal solution to the task. Most optimal motor control models so far have focused on open loop optimisation in which the sequence of motor commands or the trajectory is directly optimised with respect to some cost function, for example, minimum jerk (Flash and Hogan, 1985), minimum torque change (Uno et al., 1989), minimum end point variance (Harris and Wolpert, 1998), etc. Trajectory *planning* and *execution* steps are separated and errors during execution are compensated for by using a feedback component (e.g., PID controller). However, these corrections are not taken into account in the optimisation process.

A suggested alternative to open loop models are closed loop optimisation models, namely optimal feedback controllers (OFC) (Todorov, 2004). In contrast to open loop optimisation that just produces a desired optimal trajectory, in OFC, the gains of a feedback controller are optimised to 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 (Todorov and Jordan, 2003)). 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. Another interesting feature of OFC is that desired trajectories do not need to be planned explicitly but they simply fall out from the feedback control laws. Empirically, OFC also accounts for many motion patterns that have been observed in natural, redundant systems and human experiments (Shadmehr and Wise, 2005) including the confounding trial-to-trial variability in individual degrees of freedom that, remarkably, manages to not compromise task optimality (Li, 2006; Scott, 2004). Therefore, this paradigm is potentially a very attractive control strategy for artificial anthropomorphic systems (i.e., large DoF, redundant actuation, flexible lightweight construction, variable stiffness).

Finding an optimal control policy for nonlinear systems is a big challenge because they do not follow the well explained Linear-Quadratic-Gaussian formalism (Stengel, 1994). Global solutions could be found in theory by applying dynamic programming methods (Bertsekas, 1995) 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 (Thrun, 2000), and it has been suggested that sampling can avoid the curse of dimensionality if the underlying problem is simple enough (Atkeson, 2007), as is the case if the dynamics and cost functions are very smooth.

As an alternative, one may use iterative approaches that are based on local approximations of the cost and dynamics functions, such as differential dynamic programming (Dyer and McReynolds, 1970; Jacobson and Mayne, 1970), iterative linear-quadratic regulator designs (Li and Todorov, 2004), or the recent iterative Linear Quadratic Gaussian (iLQG) framework (Todorov and Li, 2005; Li and Todorov, 2007), which will form the basis of our work.

A major shortcoming of iLQG is its 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 models to derive an iLQG formulation that is computationally less expensive (especially for large DoF systems), reacts optimally to transient perturbations as well as adapts to systematic changes in plant dynamics.

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 (Atkeson and Schaal, 1997) using some prior knowledge about the form of the dynamics. Similarly, Abeel et al. (Abbeel et al., 2006) 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 (or rather iLQG) employ second-order optimisation methods, and we iteratively refine the control policy 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.

The remainder of this paper is organised as follows: In the next section, we recall some basic concepts of optimal control theory and we briefly describe the iLQG framework. In Section 3, we introduce our extension to that framework, and we explain how we include a learned internal model of the dynamics. We demonstrate the benefits of our method experimentally in Section 4, and we conclude the paper with a discussion of our work and future research directions in Section 5.

2 LOCALLY-OPTIMAL FEEDBACK CONTROL

Let $\mathbf{x}(t)$ denote the state of a plant and $\mathbf{u}(t)$ the applied control signal at time t. In this paper, the state consists of the joint angles $\mathbf{q}$ and velocities $\dot{\mathbf{q}}$ of a robot, and the control signals $\mathbf{u}$ are torques. 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

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

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 state our problem 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 (Todorov and Li, 2005)

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

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 not separated anymore, and for example, redundancy can actually be exploited in order to decrease the cost.

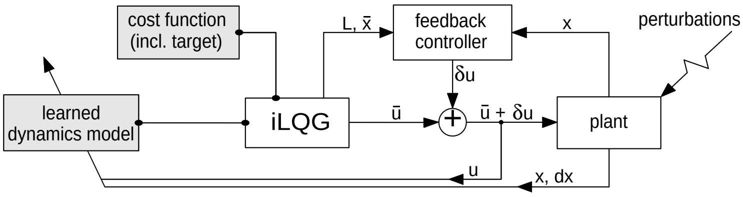


Figure 1: Illustration of our iLQG–LD learning and control scheme.

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 problem is convex and can be solved analytically (Stengel, 1994).

In the more realistic case of non-linear dynamics and non-quadratic cost, one can make use of time-varying linear approximations and apply a similar formalism to iteratively improve a policy, until at least a local minimum of the cost function is found. The resulting iLQG algorithm has only recently been introduced (Todorov and Li, 2005), so we give a brief summary in the following.

One starts with an initial time-discretised control sequence $\bar{\mathbf{u}}_k \equiv \bar{\mathbf{u}}(k\Delta t)$ and applies the deterministic forward dynamics to retrieve an initial trajectory $\bar{\mathbf{x}}_k$, where

$$\bar{\mathbf{x}}_{k+1} = \bar{\mathbf{x}}_k + \Delta t\, \mathbf{f}(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k). \qquad (3)$$

Linearising the discretised dynamics (1) around $\bar{\mathbf{x}}_k$ and $\bar{\mathbf{u}}_k$ and subtracting (3), one gets a dynamics equation for the deviations $\delta\mathbf{x}_k = \mathbf{x}_k - \bar{\mathbf{x}}_k$ and $\delta\mathbf{u}_k = \mathbf{u}_k - \bar{\mathbf{u}}_k$:

$$\begin{aligned}\delta\mathbf{x}_{k+1} \quad &= \quad \left(\mathbf{I} + \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big|_{\bar{\mathbf{x}}_k}\right)\delta\mathbf{x}_k + \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\Big|_{\bar{\mathbf{u}}_k}\delta\mathbf{u}_k \\ & \quad + \sqrt{\Delta t}\left(\mathbf{F}(\mathbf{u}_k) + \frac{\partial \mathbf{F}}{\partial \mathbf{u}}\Big|_{\bar{\mathbf{u}}_k}\delta\mathbf{u}_k\right)\boldsymbol{\xi}_k. \qquad (4)\end{aligned}$$

Similarly, one can derive an approximate cost function which is quadratic in $\delta\mathbf{u}$ and $\delta\mathbf{x}$. Thus, in the vicinity of the current trajectory $\bar{\mathbf{x}}$, the two approximations form a "local" LQG problem, which can be solved analytically and yields an affine control law $\delta\mathbf{u}_k = \mathbf{l}_k + \mathbf{L}_k\delta\mathbf{x}_k$ (for details please see (Todorov and Li, 2005)). This control law is fed into the linearised dynamics (eq. 4 without the noise term) and the resulting $\delta\mathbf{x}$ are used to update the trajectory $\bar{\mathbf{x}}$. In the same way, the control sequence $\bar{\mathbf{u}}$ is updated from $\delta\mathbf{u}$. This process is repeated until the total cost cannot be reduced anymore. The resultant control sequence $\bar{\mathbf{u}}$ can than be applied to the system, whereas the matrices $\mathbf{L}_k$ from the final iteration may serve as feedback gains.

In our current implementation[1], we do not utilise an explicit noise model $\mathbf{F}$ for the sake of clarity of results; in any case, a matching feedback control law is only marginally superior to one that is optimised for a deterministic system (Todorov and Li, 2005).

3 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. Given an initial torque sequence $\bar{\mathbf{u}}^0$, the iLQG iterations can be carried out as described in the previous section, 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.

The domain of real-time robot control demands certain properties of a learning algorithm, namely fast learning rates, high prediction speeds at run-time, and robustness towards negative interference 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 (Vijayakumar et al., 2005). In LWPR, the regression function is constructed by blending local linear models, each of which is endowed with a locality kernel that defines the area of its validity (also termed its receptive field). During training, the parameters of the local models (locality and fit) are updated using incremental Partial Least Squares, and models can be

[1] We used an adapted version of the iLQG implementation at: www.cogsci.ucsd.edu/∼todorov/software.htm

pruned or added on an as-need basis, for example, when training data is generated in previously unexplored regions.

Usually the receptive fields of LWPR are modelled by Gaussian kernels, so their activation or response to a query vector $\mathbf{z}$ (combined inputs $\mathbf{x}$ and $\mathbf{u}$ of the forward dynamics $\tilde{\mathbf{f}}$) is given by

$$w_k(\mathbf{z}) = \exp\left(-\frac{1}{2}(\mathbf{z}-\mathbf{c}_k)^T\mathbf{D}_k(\mathbf{z}-\mathbf{c}_k)\right), \qquad (5)$$

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 separately for notational convenience, the regression function can be written as

$$\tilde{f}(\mathbf{z}) = \frac{1}{W}\sum_{k=1}^{K} w_k(\mathbf{z})\psi_k(\mathbf{z}), \quad W = \sum_{k=1}^{K} w_k(\mathbf{z}), \qquad (6)$$

$$\psi_k(\mathbf{z}) = b_k^0 + \mathbf{b}_k^T(\mathbf{z}-\mathbf{c}_k), \qquad (7)$$

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 (Vijayakumar et al., 2005), 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. As we will see later, the factor λ can be used to model biologically realistic adaptive behaviour to external force-fields.

So far, we have shown how the problem of unknown or changing system dynamics can be solved within iLQG–LD. Another important issue to address is the computational complexity. The iLQG framework has been shown to be the most effective locally optimal control method in terms of convergence speed and accuracy (Li, 2006). Nevertheless the computational cost of iLQG remains daunting even for simple movement systems, preventing their application to real-time, large DoF systems. A major component of the computational cost is due to 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 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 also been proposed in (Atkeson et al., 1997). Differentiating the LWPR predictions (6) with respect to $\mathbf{z} = (\mathbf{x}; \mathbf{u})$ yields terms

$$\begin{aligned}
\frac{\partial \tilde{f}(\mathbf{z})}{\partial \mathbf{z}} &= \frac{1}{W}\sum_k \left(\frac{\partial w_k}{\partial \mathbf{z}}\psi_k(\mathbf{z}) + w_k\frac{\partial \psi_k}{\partial \mathbf{z}}\right) \\
&\quad - \frac{1}{W^2}\sum_k w_k(\mathbf{z})\psi_k(\mathbf{z})\sum_l \frac{\partial w_l}{\partial \mathbf{z}} \qquad (8) \\
&= \frac{1}{W}\sum_k \left(-\psi_k w_k \mathbf{D}_k(\mathbf{z}-\mathbf{c}_k) + w_k\mathbf{b}_k\right) \\
&\quad + \frac{\tilde{f}(\mathbf{z})}{W}\sum_k w_k\mathbf{D}_k(\mathbf{z}-\mathbf{c}_k) \qquad (9)
\end{aligned}$$

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 4 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 could be distributed across multiple processors, which would yield a further significant performance gain.

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

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

4 EXPERIMENTS

We studied iLQG–LD on two different joint torque controlled manipulators. The first (Fig. 2, left) is a horizontally planar 2 DoF manipulator similar to the one used in (Todorov and Li, 2005). 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. The second experimental setup is a 6 DoF manipulator, the physical parameters (i.e., inertia tensors, mass, gear ratios etc.) of which are a faithful model of the actual *DLR Light-Weight Robot (LWR)* from the German Aerospace Centre[2] (Fig. 2, right). This setup is used to evaluate iLQG–LD on a realistic, redundant anthropomorphic system.

[2] http://www.dlr.de

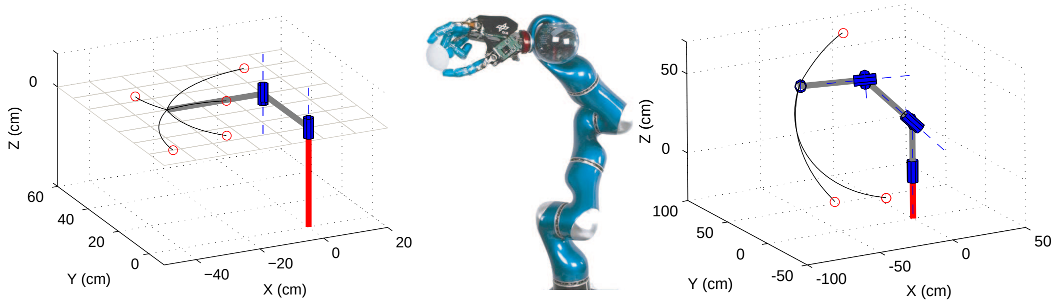


Figure 2: Two different manipulator models with selected targets (circles) and iLQG generated trajectories as benchmark data. All models are simulated using the Matlab Robotics Toolbox. Left: 2 DoF planar manipulator model; Middle: picture of real LWR; Right: 6 DoF LWR model (without hand).

Our 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

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{b}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}), \qquad (10)$$

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. It is important to note that while the above analytical dynamics perfectly match the system dynamics in simulation, they are at best, an extremely crude approximation to the dynamics of the real hardware arm.

We study movements for a fixed motion duration of one second, which we discretise into $K = 100$ steps (Δt = 0.01s). 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[3]

$$v = w_p\,|\,\mathbf{q}_K - \mathbf{q}_{tar}\,|^2 + w_v\,|\,\dot{\mathbf{q}}_K\,|^2 + w_e \sum_{k=0}^{K} |\,\mathbf{u}_k\,|^2 \Delta t, \qquad (11)$$

where the factors for the target position accuracy (w_p), the final target velocity accuracy (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. We first present results for the 2 DoF planar arm in order to test whether our theoretical assumptions hold and iLQG–LD works in practice (Sections 4.1 and 4.2). In a final experiment, we present qualitative and quantitative results to show that iLQG–LD scales up to the redundant 6 DoF anthropomorphic system (Section 4.3).

4.1 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 (reflected by the different test nMSE). 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.

Real world systems usually suffer from control dependent noise, so in order to be practicable, iLQG–LD has to be able to cope with this. 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 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.

4.2 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

[3] We specify the target in joint space only for the 2-DoF arm.

iLQG–LD	(L)	(M)	(H)	iLQG
Train. 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

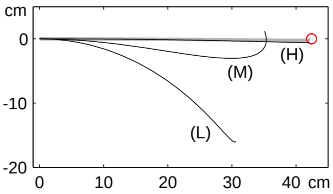
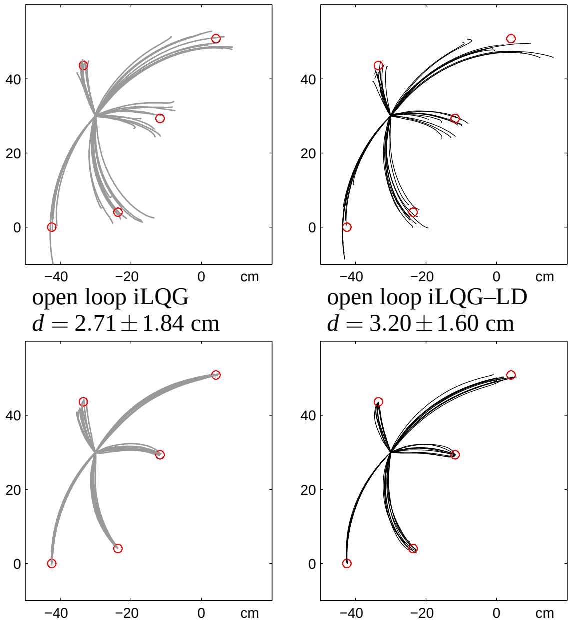


Figure 3: Behaviour of iLQG–LD for learned models of different quality. Right: Trajectories in task space produced by iLQG–LD (black lines) and iLQG (grey line).



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

Figure 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.

result from altered morphology or wear and tear. We carried out adaptive reaching experiments in our simulation similar to the human manipulandum experiments in (Shadmehr and Mussa-Ivaldi, 1994). 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 relevant dynamics training data in a way that could be termed "active learning". 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 (Vijayakumar et al., 2005). 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

$$\mathbf{F} = \mathbf{B}\mathbf{v}, \quad \text{with} \quad \mathbf{B} = \begin{pmatrix} 0 & 50 \\ -50 & 0 \end{pmatrix} Nm^{-1}s \quad (12)$$

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 overshooting behaviour depicted in Fig. 5 and 6 has been observed in human adaptation experiments (Shadmehr and Mussa-Ivaldi, 1994). 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 (Shadmehr and Wise, 2005).

4.3 iLQG–LD for 6 DoF

In the 6 DoF LWR, we studied reaching targets specified in *task space* coordinates $\mathbf{r} \in \mathcal{R}^3$ in order to highlight the redundancy resolution capability and trial-to-trial variability in large DoF systems. Therefore, we replaced the term $|\mathbf{q}_K - \mathbf{q}_{tar}|^2$ by $|\mathbf{r}(\mathbf{q}_K) - \mathbf{r}_{tar}|^2$ in (11), where $\mathbf{r}(\mathbf{q})$ denotes the forward kinematics.

Similar to the 2 DoF, we bootstrapped a forward dynamics model through 'motor babbling'. Next, we

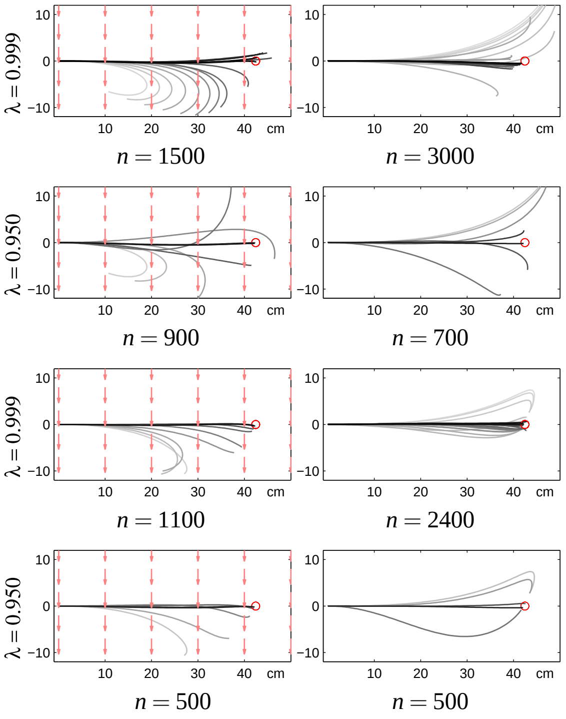


Figure 5: Illustration of adaptation experiments for open loop (rows 1,2) and closed loop (rows 3,4) iLQG–LD. Arrows depict the presence of a (constant) force field; *n* represents the number of training points required to successfully update the internal LWPR dynamics model. Darker lines indicate better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

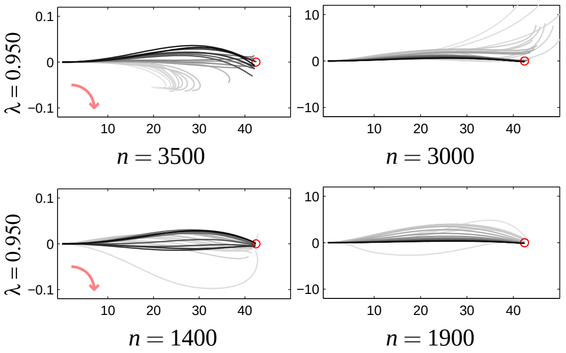


Figure 6: Adaptation to a velocity-dependent force field (as indicated by the bent arrow) and re-adaptation after the force field is switched off (right column). Top: open loop. Bottom: closed loop.

used iLQG–LD (closed loop, with noise) to train our dynamics model online until it converged to stable reaching behaviour. Fig. 7 (left) depicts reaching trials, 20 for each reference target, 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. 7 (right) depicts the 6 joint angle trajectories for the 20 reaching trials towards target (c). The joint angle variances are much higher compare d to the variances of the task space trajectories, meaning 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 in 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.

Table 2: Comparison of iLQG–LD and iLQG for controlling a 6 DOF arm to reach for three targets.

iLQG–LD	**Iter.**	**Run. cost**	**d (cm)**
(a)	51	**18.32** ± 0.55	**1.92** ± 1.03
(b)	99	**18.65** ± 1.61	**0.53** ± 0.20
(c)	153	**12.18** ± 0.03	**2.00** ± 1.02
iLQG	**Iter.**	**Run. cost**	**d (cm)**
(a)	58	**18.50** ± 0.13	**2.63** ± 1.63
(b)	61	**18.77** ± 0.25	**1.32** ± 0.69
(c)	132	**12.92** ± 0.04	**1.75** ± 1.30

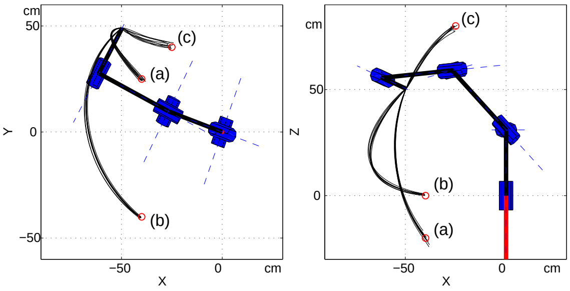


Figure 7: Illustration of the 6-DoF manipulator and the trajectories for reaching towards the targets (a,b,c). Left: top-view, right: side-view.

5 CONCLUSION

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.

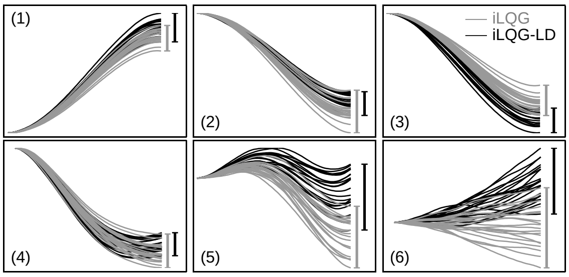


Figure 8: Illustration of the trial-to-trial variability of the 6-DoF arm when reaching towards target (c). The plots depict the joint angles (1–6) over time. Grey lines indicate iLQG, black lines stem from iLQG–LD.

Utilising the derivatives (8) of the learned dynamics model **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 the anthropomorphic LWR 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

Abbeel, P., Quigley, M., and Ng, A. Y. (2006). Using inaccurate models in reinforcement learning. In *Proc. Int. Conf. on Machine Learning*, pages 1–8.

Atkeson, C. G. (2007). Randomly sampling actions in dynamic programming. In *Proc. Int. Symp. on Approximate Dynamic Programming and Reinforcement Learning*, pages 185–192.

Atkeson, C. G., Moore, A., and Schaal, S. (1997). Locally weighted learning for control. *AI Review*, 11:75–113.

Atkeson, C. G. and Schaal, S. (1997). Learning tasks from a single demonstration. In *Proc. Int. Conf. on Robotics and Automation (ICRA)*, volume 2, pages 1706–1712, Albuquerque, New Mexico.

Bertsekas, D. P. (1995). *Dynamic programming and optimal control*. Athena Scientific, Belmont, Mass.

Dyer, P. and McReynolds, S. (1970). *The Computational Theory of Optimal Control*. Academic Press, New York.

Flash, T. and Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience*, 5:1688–1703.

Harris, C. M. and Wolpert, D. M. (1998). Signal-dependent noise determines motor planning. *Nature*, 394:780–784.

Jacobson, D. H. and Mayne, D. Q. (1970). *Differential Dynamic Programming*. Elsevier, New York.

Li, W. (2006). *Optimal Control for Biological Movement Systems*. PhD dissertation, University of California, San Diego.

Li, W. and Todorov, E. (2004). Iterative linear-quadratic regulator design for nonlinear biological movement systems. In *Proc. 1st Int. Conf. Informatics in Control, Automation and Robotics*.

Li, W. and Todorov, E. (2007). Iterative linearization methods for approximately optimal control and estimation of non-linear stochastic system. *International Journal of Control*, 80(9):14391453.

Scott, S. H. (2004). Optimal feedback control and the neural basis of volitional motor control. *Nature Reviews Neuroscience*, 5:532–546.

Shadmehr, R. and Mussa-Ivaldi, F. A. (1994). Adaptive representation of dynamics during learning of a motor task. *The Journal of Neurosciene*, 14(5):3208–3224.

Shadmehr, R. and Wise, S. P. (2005). *The Computational Neurobiology of Reaching and Ponting*. MIT Press.

Stengel, R. F. (1994). *Optimal control and estimation*. Dover Publications, New York.

Thrun, S. (2000). Monte carlo POMDPs. In Solla, S. A., Leen, T. K., and Müller, K. R., editors, *Advances in Neural Information Processing Systems 12*, pages 1064–1070. MIT Press.

Todorov, E. (2004). Optimality principles in sensorimotor control. *Nature Neuroscience*, 7(9):907–915.

Todorov, E. and Jordan, M. (2003). A minimal intervention principle for coordinated movement. In *Advances in Neural Information Processing Systems*, volume 15, pages 27–34. MIT Press.

Todorov, E. and Li, W. (2005). A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *Proc. of the American Control Conference*.

Uno, Y., Kawato, M., and Suzuki, R. (1989). Formation and control of optimal trajectories in human multijoint arm movements: minimum torque-change model. *Biological Cybernetics*, 61:89–101.

Vijayakumar, S., D'Souza, A., and Schaal, S. (2005). Incremental online learning in high dimensions. *Neural Computation*, 17:2602–2634.