

# Optimal Feedback Control for Anthropomorphic Manipulators

Djordje Mitrovic, Sho Nagashima, Stefan Klanke, Takamitsu Matsubara, Sethu Vijayakumar

**Abstract**—We study target reaching tasks of redundant anthropomorphic manipulators under the premise of minimal energy consumption and compliance during motion. We formulate this motor control problem in the framework of Optimal Feedback Control (OFC) by introducing a specific cost function that accounts for the physical constraints of the controlled plant. Using an approximative computational optimal control method we can optimally control a high-dimensional anthropomorphic robot without having to specify an explicit inverse kinematics, inverse dynamics or feedback control law. We highlight the benefits of this biologically plausible motor control strategy over traditional (open loop) optimal controllers: The presented approach proves to be significantly more energy efficient and compliant, while being accurate with respect to the task at hand. These properties are crucial for the control of mobile anthropomorphic robots, that are designed to interact safely in a human environment. To the best of our knowledge this is the first OFC implementation on a high-dimensional (redundant) manipulator.

## I. 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 [1], [2], [7], [14] 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 hereby have focused on *open loop optimisation* in which the sequence of motor commands or the kinematic trajectory itself is directly optimised with respect to some cost function. The solution to an open loop problem gives us an optimal control trajectory, which can usually be obtained by solving a two point boundary difference/differential equation derived by applying *Pontryagin's minimum principle* [14]. Therefore trajectory *planning* and *execution* steps are separated and errors during execution are compensated for by using a feedback component (e.g., PID controller). However, the required corrections are not taken into account in the optimisation process.

\*D. Mitrovic, S. Klanke, and S. Vijayakumar are with the School of Informatics, University of Edinburgh, United Kingdom. d.mitrovic@ed.ac.uk, s.klanke@ed.ac.uk, sethu.vijayakumar@ed.ac.uk; S. Nagashima and T. Matsubara are with the Department of Information Processing, Nara Institute of Science and Technology, Japan. sho-n@is.naist.jp, takam-m@is.naist.jp

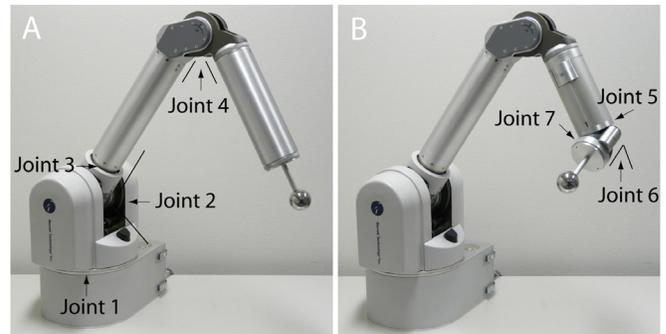


Fig. 1. The antropomorphic manipulator (Barrett WAM) used for our experiments. (A): 4 DoF setup; (B): 7 DoF setup (with wrist attached).

A suggested alternative to open loop models are *closed loop* optimisation models. The solution to a closed loop problem is an *optimal feedback controller (OFC)*, i.e. a function mapping from the state to the optimal control value<sup>1</sup>. Thus, in contrast to classical control, calculation of the trajectory (planning) and the control signal (execution) is handled in one go.

Recently OFC has found large attention in the study of biological motor control [16] where it has shown to be a powerful theory for interpreting human motor behaviour [8], [13]. A key property of such OFC, that typically minimise for task error and energy consumption, is that errors are only corrected by the controller if they adversely affect the task performance, otherwise they are neglected (minimum intervention principle [17]). Therefore redundant degrees of freedom, often a nuisance for kinematic path planning, in OFC are actually exploited in order to decrease the cost. This is an important property especially for systems that demand low energy consumption and compliant motion, such as mobile humanoid robots interacting safely in a human environment. Furthermore OFC solutions, unlike classic robot controllers, neither require a separate inverse kinematics nor inverse dynamics computations nor the determination of the feedback control law.

Despite the appeal of closed loop optimal control, its application on realistic high dimensional and non-linear systems<sup>2</sup> is problematic: It is very complicated to solve a closed loop problem since the information represented by the optimal value function is essentially equal to the information obtained by solving a two point boundary or-

<sup>1</sup>Closed loop models are often referred as global methods since each point in the state space is considered.

<sup>2</sup>If the plant dynamics is linear and the cost function is quadratic the optimisation problem is convex and can be solved analytically.

dinary differential equation from *each* point in state space. Global solutions could be found in theory by applying dynamic programming methods [1] that are based on the Hamilton-Jacobi-Bellman equations, which in their basic form rely on a discretisation of the state and action space, an approach that is not viable for large DoF systems (curse of dimensionality). Approximative optimal control methods like *Differential Dynamic Programming (DDP)* [2], [6] and the *iterative Linear Quadratic Gaussian (ILQG)* [18] can be employed to overcome computational intractability. These methods iteratively compute an optimal trajectory together with a locally valid feedback law and therefore are not directly subject to the curse of dimensionality. Previous work largely has focused on the theoretical aspects in idealised simulated scenarios [8], [15], [11] whereas we demonstrate that OFC indeed is a practical approach for controlling real robotic systems [5].

In this paper we develop an optimal feedback controller for a high-dimensional anthropomorphic manipulator. We investigate finite-time target reaching tasks under the premise of minimal energy consumption and compliance during motion as it is required for safe interaction with the robot. After a brief explanation of OFC and approximative methods in Section II we introduce an optimisation criterion that allows us to incorporate the required physical constraints of our plant into the optimisation process and we formulate a forward dynamics function that avoids discontinuities arising from joint friction in the robot (Section III). In Section IV we apply the local OFC law to the Barrett WAM manipulator and we highlight the benefits of this motor control strategy for anthropomorphic robots, that are designed to interact in a human environment. To the best of our knowledge this is the first documented OFC investigation on real (high-dimensional) hardware.

## II. OPTIMAL FEEDBACK CONTROL

Let  $\mathbf{x}_t$  denote the state of a plant and  $\mathbf{u}_t$  the applied control signal (i.e., joint torque) at time  $t$ . The state consists of the joint angles  $\mathbf{q}$  and velocities  $\dot{\mathbf{q}}$  of the robot. We can express the system dynamics in deterministic form as

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u})dt. \quad (1)$$

OFC has been formalised for stochastic dynamics with partial observability [9]. However we will ignore stochasticity in this work as we would require a system noise and estimation noise model of the real hardware.

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  $t = T$ . 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* or *running 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  $\pi(t, \mathbf{x})$  for

selecting  $\mathbf{u}_t$ , we can write the expected cost of following that policy from time  $t$  as [18]

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

In OFC one then aims to find the policy  $\pi$  that minimises the total expected cost  $v^\pi(0, \mathbf{x}_0)$ .

The choice of the cost function determines the desired behaviour of the system and encodes the task. In the past, many performance criteria have been proposed (e.g., [3], [4], [19]). Inspired by the study of biological systems where metabolic cost is crucial, the minimum energy criterion [12] is also a very appealing strategy for mobile robots since battery life is limited. Furthermore minimum energy implies smooth trajectories, reduced stress on the actuators, and joint compliance through low corrective gains, which is also desired in our application.

### A. Iterative Linear Quadratic Gaussian (ILQG)

The fundamental idea behind ILQG is that, when the dynamics process is non-linear and costs are non-quadratic, one can still apply the LQG solution approximately around a nominal trajectory and use local solutions to iteratively improve the nominal solution.

We study reaching movements of a manipulator as a finite time horizon problems of length  $K = k\Delta t$  seconds. Typical values are  $k = 100$  discretisation steps with a simulation rate of  $\Delta t = 0.01$ . We assume that we have identified an accurate forward dynamics model  $\mathbf{f}(\mathbf{x}, \mathbf{u})$  of our plant. We define a discrete cost function  $v$  encoding for a task, where the manipulator has to move and stop at the target using a minimal amount of energy [18]:

$$v = w_p |\mathbf{r}(\mathbf{q}_K) - \mathbf{r}_{tar}|^2 + w_v |\dot{\mathbf{q}}_K|^2 + \Delta t \sum_{k=0}^K \underbrace{w_e |\mathbf{u}_k|^2}_{=v_k}. \quad (3)$$

The factors for the target position accuracy ( $w_p$ ), the stopping condition ( $w_v$ ), and the energy term<sup>3</sup> ( $w_e$ ) weight the importance of each component. Further  $\mathbf{r}(\mathbf{q})$  denotes the forward kinematics and  $\mathbf{r}_{tar}$  the task space coordinates of our reaching target.

The ILQG algorithm starts with a time-discretised initial guess (e.g., gravity compensation, or zero sequence) of a control sequence and then iteratively improves it w.r.t. the performance criteria in  $v$  (eq. 3). From the initial control sequence  $\bar{\mathbf{u}}^i$  at the  $i^{th}$ -iteration, the corresponding state sequence  $\bar{\mathbf{x}}^i$  is retrieved using the deterministic forward dynamics  $\mathbf{f}$  with a standard Euler integration  $\bar{\mathbf{x}}_{k+1}^i = \bar{\mathbf{x}}_k^i + \Delta t \mathbf{f}(\bar{\mathbf{x}}_k^i, \bar{\mathbf{u}}_k^i)$ . In a next step the discretised dynamics (eq. 1) are linearly approximated as

$$\delta \mathbf{x}_{k+1} = \left( \mathbf{I} + \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\bar{\mathbf{x}}_k} \right) \delta \mathbf{x}_k + \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \Big|_{\bar{\mathbf{u}}_k} \delta \mathbf{u}_k. \quad (4)$$

<sup>3</sup>We approximate energy with joint torque consumption, and we ignore the efficiency factors of the motors.

Similarly one can derive a quadratic approximation of the cost function around  $\bar{\mathbf{x}}_k^i$  and  $\bar{\mathbf{u}}_k^i$ :

$$\begin{aligned} \text{cost}_k &= q_k + \delta \mathbf{x}_k^T \mathbf{q}_k + \frac{1}{2} \delta \mathbf{x}_k^T \mathbf{Q}_k \delta \mathbf{x}_k + \\ &\quad \delta \mathbf{u}_k^T \mathbf{r}_k + \frac{1}{2} \delta \mathbf{u}_k^T \mathbf{R}_k \delta \mathbf{u}_k + \delta \mathbf{u}_k^T \mathbf{P}_k \delta \mathbf{x}_k \end{aligned} \quad (5)$$

where

$$\begin{aligned} q_k &= \Delta t v_k; & \mathbf{q}_k &= \Delta t \left. \frac{\partial v_k}{\partial \mathbf{x}} \right|_{\bar{\mathbf{x}}_k} \\ \mathbf{Q}_k &= \Delta t \left. \frac{\partial^2 v_k}{\partial \mathbf{x} \partial \mathbf{x}} \right|_{\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k}; & \mathbf{P}_k &= \Delta t \left. \frac{\partial^2 v_k}{\partial \mathbf{u} \partial \mathbf{x}} \right|_{\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k} \\ \mathbf{r}_k &= \Delta t \left. \frac{\partial v_k}{\partial \mathbf{u}} \right|_{\bar{\mathbf{u}}_k}; & \mathbf{R}_k &= \Delta t \left. \frac{\partial^2 v_k}{\partial \mathbf{u} \partial \mathbf{u}} \right|_{\bar{\mathbf{u}}_k}. \end{aligned} \quad (6)$$

Both approximations are formulated as deviations  $\delta \mathbf{x}_k^i = \mathbf{x}_k^i - \bar{\mathbf{x}}_k^i$  and  $\delta \mathbf{u}_k^i = \mathbf{u}_k^i - \bar{\mathbf{u}}_k^i$  of the current optimal trajectory and therefore form a ‘‘local’’ LQG problem. This linear quadratic problem can be solved efficiently via a modified Ricatti-like set of equations that yields an affine control law  $\pi_k(\delta \mathbf{x}) = \mathbf{l}_k + \mathbf{L}_k \delta \mathbf{x}_k$ . This control law has a special form: since it is defined in terms of deviations of a nominal trajectory and since it needs to be implemented iteratively, it consists of an open loop component  $\mathbf{l}_k$  and a feedback-component  $\mathbf{L}_k \delta \mathbf{x}_k$ . The actual optimisation in ILQG 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 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 state sequence  $\bar{\mathbf{x}}$  (i.e., trajectory). Along with the 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 desired trajectory on the plant (Fig. 2). The control law for each time step  $k$  is defined as,

$$\mathbf{u}_k^{\text{plant}} = \bar{\mathbf{u}}_k + \delta \mathbf{u}_k \quad (7)$$

$$\delta \mathbf{u}_k = \mathbf{L}_k \cdot (\mathbf{x}_k - \bar{\mathbf{x}}_k), \quad (8)$$

where  $\mathbf{x}_k$  represents the real plant position and  $\bar{\mathbf{x}}_k$  the desired position at time  $k$ .

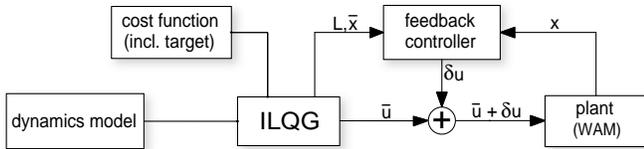


Fig. 2. The OFC scheme using ILQG.

For the control of robotic manipulators ILQG has several desirable properties: (i) ILQG resolves kinematic redundancies automatically, i.e., no explicit inverse kinematics method is required. (ii) It produces a feedforward control sequence

with corresponding optimal feedback control law. As we see later this allows us to achieve highly compliant movement plans that are still accurate w.r.t. the task at hand. (iii) We can specify a specific motion task in an ‘‘intuitive’’ high level formulation in the cost function.

At this point one should note that there is no guarantee that ILQG will converge to a global minimum. In fact experience from practice shows that the initial control sequence often affects the final outcome of the algorithm. From a computational perspective the dynamics linearisation steps in the ILQG algorithm loop prove to be the computational bottleneck. This process requires the partial derivatives  $\partial \mathbf{f}(\mathbf{x}, \mathbf{u}) / \partial \mathbf{x}$  and  $\partial \mathbf{f}(\mathbf{x}, \mathbf{u}) / \partial \mathbf{u}$ , which are computed, in a generally applicable case<sup>4</sup>, using finite differences.

### III. ROBOT MODEL AND CONTROL

In this paper we use the WAM arm (Barrett Technology Inc.) (Figure 1) as an implementation platform. The WAM is a cable driven 7 DoF anthropomorphic manipulator (4 DoF without wrist), with a reach of about  $1m$  and a payload of  $4kg$ . The platform is well suited for implementing dynamics model based control (like OFC) since the inertial parameters are publicly available and motor torques can be directly commanded to the WAM. On the sensing side the platform has joint position encoders but offers no joint torque or other external sensors.

#### A. Forward dynamics the cost function

We model the non-linear dynamics of our plant using standard equations of motion where the joint torques  $\boldsymbol{\tau}$  are given by

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

As before  $\mathbf{q}$  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 joint friction, and  $\mathbf{g}(\mathbf{q})$  defines the gravity loading depending on the joint angles  $\mathbf{q}$  of the manipulator. All inertial parameters are provided by the robot manufacturer.

#### B. Avoiding discontinuities in the dynamics

The WAM suffers from significant frictional joint torques  $\mathbf{b}(\dot{\mathbf{q}})$ , which had to be estimated separately. Joint friction is usually modelled to consist of a *static*<sup>5</sup> and *kinetic* Coulomb component as well as of a *viscous* friction component (Fig. 3). The Coulomb friction model is discontinuous and therefore has no derivatives defined at  $\dot{\mathbf{q}} = 0$ , which is problematic because internally ILQG relies on derivatives to improve the control law. Furthermore, very steep gradients (as occurring in step-like functions) can sometimes have a bad impact on the convergence speed of the algorithm. We overcome this problem in practice by ignoring the static Coulomb friction and by approximating the kinetic Coulomb

<sup>4</sup>In this work we also follow this approach

<sup>5</sup>Also called ‘‘stiction’’.

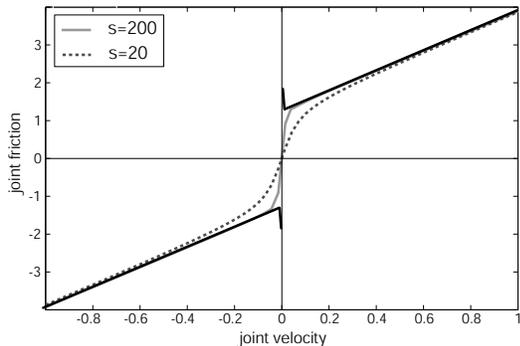


Fig. 3. Approximative continuous friction model. Solid black line represents the theoretical discontinuous Coulomb friction. For an (example) steepness parameter of  $s = 200$  the derivatives at the start condition ( $\dot{\mathbf{q}} = 0$ ) become too large and ILQG diverges whereas for  $s = 20$  it successfully converges.

and viscous friction in each joint with the following smooth and continuous sigmoid function

$$b(\dot{q}) = b_c \arctan(s\dot{q}) \frac{2}{\pi} + B\dot{q}, \quad (10)$$

where  $s$  indicates the “steepness” of the fitted arctan function (Fig. 3),  $b_c$  is the kinetic Coulomb friction, and  $B$  is the viscous friction coefficient. We heuristically identified the steepness parameter as  $s = 20$  (for all joints) such that it led to overall stable ILQG convergence while providing sufficient modelling accuracy.

We then used the constant angular velocity motion test [10] as an identification method for the viscous friction coefficient and the kinetic Coulomb friction. When a small step input torque  $\tau^*(i)$  is applied to the target joint  $i$  while keeping the other joints fixed,  $\dot{q}_i$  converges to some constant angular velocity as  $t \rightarrow \infty$  by the effect of the damping torque. By executing the test motions ten times with various values of  $\tau^*(i)$  for each joint,  $B$  and  $b_c$  can be easily estimated by a least-square method. Table I shows the obtained results for all joints.

Joint	i=1	i=2	i=3	i=4	i=5	i=6	i=7
$\mathbf{B}(i)$	1.142	0.946	0.309	0.255	0.025	0.039	0.004
$\mathbf{b}_c(i)$	2.516	2.581	2.038	0.956	0.323	0.315	0.066

TABLE I

ESTIMATED JOINT FRICTION PARAMETERS FOR THE WAM.

Since we are commanding joint torques ( $\tau = \mathbf{u}$ ) the deterministic forward dynamics used in ILQG takes the form

$$\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})^{-1} (\mathbf{u} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}) - \mathbf{b}(\dot{\mathbf{q}})). \quad (11)$$

### C. Incorporating real world constraints into OFC

The model based control of real hardware must obey several physical constraints, which somehow need to be incorporated into the OFC framework. These correspond to the physical boundaries of the manipulator, namely the maximally applicable motor torques ( $\mathbf{u}_{min}, \mathbf{u}_{max}$ ), the joint angle limits ( $\mathbf{q}_{min}, \mathbf{q}_{max}$ ), and the maximally executable

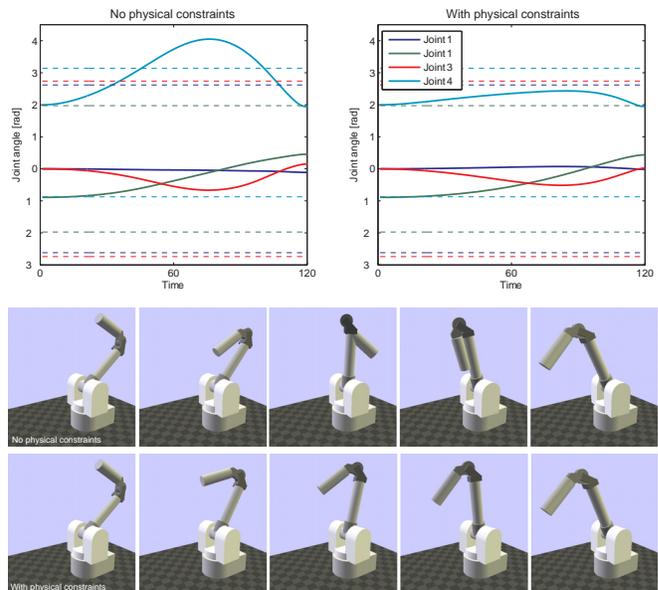


Fig. 4. Comparison of ILQG results obtained without (left) and with (right) physical constraint terms  $P(\mathbf{q})$  and  $V(\dot{\mathbf{q}})$ . The unconstrained solution violates the physical limits which would lead to a self collision applied to the WAM (top row of simulation screenshots).

joint velocities ( $\dot{\mathbf{q}}_{min}, \dot{\mathbf{q}}_{max}$ ). ILQG handles the control constraints during optimisation by enforcing control boundaries on  $\bar{\mathbf{u}}$  and by modifying the feedback gain matrix  $\mathbf{L}_k$  (i.e., setting  $\mathbf{L}_k$  to zero) whenever an element of the control sequence lies on the constraint boundary. Applied to the hardware however we found that control constraints are rarely violated whereas state constraints are much more critical (Fig. 4) and ILQG does not handle those constraints in any form. We therefore propose to incorporate the joint angle and joint velocity boundaries as optimisation constraints into the running cost in (3) as

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

$$P(\mathbf{q}) = w_{pb} \sum_{i=1}^4 ([q_i - q_i^{max}]_+^2 + [q_i^{min} - q_i]_+^2) \quad (13)$$

$$V(\dot{\mathbf{q}}) = w_{vb} \sum_{i=1}^4 ([\dot{q}_i - \dot{q}_i^{max}]_+^2 + [\dot{q}_i^{min} - \dot{q}_i]_+^2) \quad (14)$$

For the joint angle boundaries ( $w_{pb}$ ), and the joint velocity boundaries ( $w_{vb}$ ) we use following notational convention  $[x]_+ = \max(0, x)$  given that for each joint ( $q_i^{min} < 0 < q_i^{max}$ ) and ( $\dot{q}_i^{min} < 0 < \dot{q}_i^{max}$ ).

Another issue that needs to be addressed is the correct initialisation of the robot’s joint torque state. Before starting the optimal reaching movement the robot is assumed to be in a stationary state, which is achieved by applying the torque sequence  $\mathbf{u}_{init}$ . For the WAM the gravity compensation is known and we therefore set  $\mathbf{u}_{init} = \mathbf{g}(\mathbf{q}_0)$ . At reaching start time  $k = 0$  the torque sequences of the plant and the ILQG

result should be equal, i.e.,  $\mathbf{g}(\mathbf{q}_0) = \bar{\mathbf{u}}_0$ . However there is no way to "tell" ILQG what the initial torques should be at time  $k = 0$  and therefore a "torque jump" will be commanded to the plant. A similar effect arises at the end of the motion ( $k = K$ ) where we usually transfer from reaching back to gravity compensation and it would be desirable that  $\mathbf{g}(\mathbf{q}_K) = \bar{\mathbf{u}}_K$ . Such torque transition errors must be avoided since they destabilise the plant and produce high stresses on the actuators, which contradicts the energy efficient control law that we want to achieve. Therefore an additional constraint is required in order to avoid excessively large motor jumps at the beginning and end of the reaching. In theory one can partially avoid that problem by modelling the underlying motor dynamics that define the maximal change in motor control signals. However this approach blows up the state space by additional  $N$  states, which in consequence increases the computational load on ILQG. We solve this problem alternatively by (i) using the gravity compensation torques  $\mathbf{g}(\mathbf{q}_0)$  as initial torque sequence and by (ii) introducing an additional "soft-start" and "soft-stop" constraint into the cost function. We extend our cost function  $v$  by another time-dependent term in the running cost

$$v^* = v + \Delta t \sum_{k=0}^K w_v^* |\dot{\mathbf{q}}_k|^2 \quad (15)$$

$$w_v^* = \begin{cases} 1 - \frac{k}{K_s}, & \text{if } k < K_s \\ 1 - \frac{K-k}{K_s}, & \text{if } k > (K - K_s) \\ 0, & \text{otherwise} \end{cases} .$$

Therefore the reaching has now become closer to a hold-reach-hold task, where  $K_s$  determines the transition smoothness, which is formulated in terms of joint angle velocities, irrespective of the current arm position at start and end.

#### IV. RESULTS

In this section we discuss the results from controlling the WAM using the proposed (local) optimal feedback controller. We study two setups: First we use the 4 DoF setup to show the basic concepts and compare the results to other trajectory planners in terms of task achievement, compliance and energy consumption. The second setup contains all 7 DoF and here we will highlight the scalability of our approach and we present a set of control applications.

##### A. OFC with 4 DoF

We study movements for the fixed motion duration of 1.6 seconds, which we discretise into  $K = 800$  steps ( $\Delta t = 0.002$  sec) corresponding to the robot's operation rate of  $500\text{Hz}$ . The manipulator starts at an initial position  $\mathbf{q}_0$  and reaches towards several targets  $\mathbf{t}$ , defined in end effector task space coordinates  $(x, y, z)$  in meters. During movement we wish to minimise the proposed cost function (15).

We set the arm's start position as  $\mathbf{q}_0 = (0, -0.89, 0, 2.0)^T \text{ rad}$  and 3 reference targets (left, middle, right):  $\mathbf{t}_{left} = (0.55, 0.3, 0.25)$ ,  $\mathbf{t}_{center} = (0.55, 0, 0.25)$ ,  $\mathbf{t}_{right} = (0.55, -0.3, 0.25)$ . These targets represent a reasonably far distance (center 0.73m; left/right 0.79m)

for a reaching time of 1.6 sec. We used following cost parameters:  $w_p = 50000$ ,  $w_v = 5000$ ,  $w_e = 1$ ,  $w_{pb} = 50$ ,  $w_{vb} = 100$ ,  $K_s = 10$ . Next we found the optimal solutions using ILQG for the three targets. Using the gravity compensation as initial trajectory, the algorithm converged after following number of iterations:  $iter_{center} = 63$ ,  $iter_{left} = 89$ ,  $iter_{right} = 68$ . The ILQG results are applied to the WAM following the control law defined in (8). The matrix  $\mathbf{L}_k = [\mathbf{L}_P \ \mathbf{L}_D]_k$  is the  $4 \times 8$  time-dependent locally valid control law that contains the optimal PD gains for each time step  $k$ . It has been shown previously [11] that these gains follow the so called minimum intervention principle [17]: As can be seen in the center panel of Fig. 5, on the example of  $\mathbf{t}_{center}$ , the L gains take into account the nature of the specified task in the cost function. Therefore the gains are very low up to about 500 time steps and then grow towards the end of the motion where task accuracy and stability are more important<sup>6</sup>. This trade-off between energy preservation and reaching task achievement is present in all joints. Notably the L-matrix is diagonally dominant with an additional coupling between joints 1 and joint 3. This coupling appears due to the redundancy those joint have for the task of reaching to the center targets, e.g., perturbations in joint 1 can partly be corrected by joint 3 and vice versa. A comparison between the desired optimal trajectories (dashed lines in Fig. 5) and the feedback corrected trajectories on the plant (solid lines) indicate that the modelled forward dynamics is fairly accurate, especially for joint 2 that exhibits the largest joint torques. We can observe effects of "unmodelled dynamics", which can be attributed to the errors in the friction estimation (see section III-B).

As mentioned in the introduction in our OFC control strategy the primary aim is task achievement in target reaching and the energy preservation during control, whereas the exact trajectory tracking here is no performance criterion. As we show next these properties allow us (i) to use less energy than other (open loop) optimal control algorithms, and (ii) to be compliant during motion.

We compare the ILQG results against an open loop optimiser following the minimum jerk optimisation criterion [3]. We use the same start and end position as before and the optimisation gives us a optimal kinematic trajectory  $\mathbf{x}^*$ , which must be tracked using a (typically) hand tuned PD feedback control law

$$\mathbf{u}_k^{plant} = \mathbf{P} \cdot (\mathbf{q}_k^* - \mathbf{q}_k) + \mathbf{D} \cdot (\dot{\mathbf{q}}_k^* - \dot{\mathbf{q}}_k). \quad (16)$$

We used two sets of PD gains: (a) The high gain factory default values of the Barrett WAM controller:  $P = 2000$ ,  $D = 500$ . (b) The maximal diagonal values of the ILQG feedback gain matrix  $\mathbf{L} : \mathbf{P} = \max(\text{diag}(\mathbf{L}_P))$ ,  $\mathbf{D} = \max(\text{diag}(\mathbf{L}_D))$ .

For a better comparison with the ILQG control paradigm we also ran the minimum jerk results with a feed-forward

<sup>6</sup>Generally the L-gains are significantly smaller than the WAM factory-default PD gains which are  $P = 2000$  and  $D = 500$  for each joint.

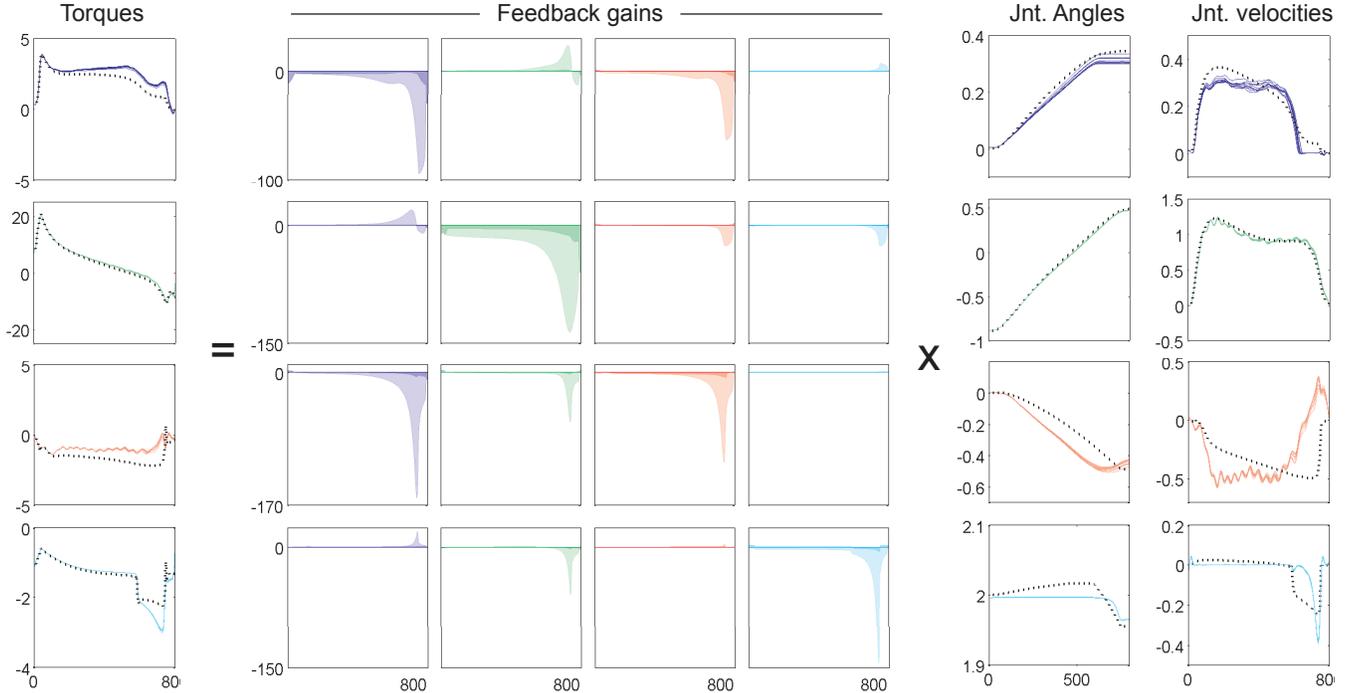


Fig. 5. Results of the ILQG control law (eq. 8) applied to the WAM with 4 DOF for a reaching to the center target  $\mathbf{t}_{center}$ . The dashed lines represent the optimal desired trajectories produced by ILQG, and the solid lines show the 10 trajectories recorded from the WAM. The shaded areas depict the feedback gain matrix ( $L$ ), where brighter shadings indicate the position gains ( $P$ ) and the darker areas depict the velocity-gains ( $D$ ).

torque sequence that we computed using the inverse dynamics

$$\mathbf{u}_k^{plant} = \boldsymbol{\tau}_k(\mathbf{q}_k^*, \dot{\mathbf{q}}_k^*, \ddot{\mathbf{q}}_k^*) + \mathbf{P} \cdot (\mathbf{q}_k^* - \mathbf{q}_k) + \mathbf{D} \cdot (\dot{\mathbf{q}}_k^* - \dot{\mathbf{q}}_k). \quad (17)$$

As before we used: (c) the standard WAM gains and (d)  $max(L)$ . Figure 6 summarises the results.

As expected, in terms of accuracy the high gain feedforward minimum jerk trial (c) is the most accurate. However it achieves this performance with the price of a fairly high energy consumption, i.e., 25% higher than ILQG. Due to the high corrective gains its compliance is reduced making the robot much more destructive in the case of unexpected collisions. In summary ILQG offers the best trade-off between end-point accuracy and energy consumption. The ILQG results for all targets are accumulated in Table II. The reaching produced by ILQG is very compliant allowing an interaction at all times with the robot (Fig. 7 and accompanying video).

TABLE II

ILQG RESULTS (MEAN  $\pm$  STD OVER 10 TRIALS) ONF 4 DOF WAM FOR 3 REFERENCE TARGETS.

ILQG	Euclidean target error [mm]	Energy [N · m]
$\mathbf{t}_{center}$	$7.576 \pm 0.148$	$8.637 \cdot 10^3 \pm 0.015 \cdot 10^3$
$\mathbf{t}_{left}$	$9.707 \pm 0.233$	$9.746 \cdot 10^3 \pm 0.020 \cdot 10^3$
$\mathbf{t}_{right}$	$9.213 \pm 0.273$	$8.994 \cdot 10^3 \pm 0.031 \cdot 10^3$

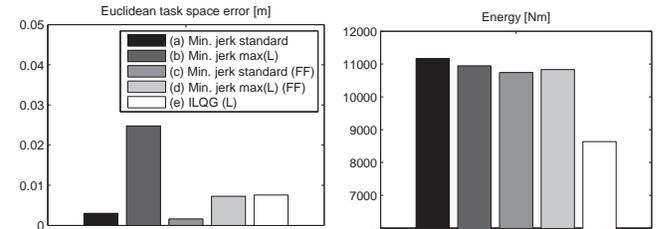


Fig. 6. Comparison of min jerk (conditions 1)-(4) and ILQG (5) - Results are averaged over 10 reaches to the center target ( $\mathbf{t}_{center}$ ). Left: Euclidean distance (error) between target and end-point. Right: “Energy” consumption computed as sum of all torques in all joints over the entire trajectory:  $\sum_{i=1}^4 \sum_{k=1}^K |\mathbf{u}_k^{plant}(i)|$ .

### B. Scaling to 7 DoF

In this section we demonstrate the scaling and redundancy resolution abilities of ILQG. We demonstrate results on two types of reaching tasks:

1) *Task space reaching without orientation*: We repeated the reaching experiment from the section with the 7 dof setup. Initial position, targets, reaching time and cost function parameters were the same as before. Unlike before we now have 4 redundant degrees of freedom, since we only look at  $(x, y, z)$  coordinates and ignore the end-effector orientation. ILQG successfully converges after  $iter_{center} = 74$ ,  $iter_{left} = 80$ ,  $iter_{right} = 71$  iterations and resolves the kinematic redundancies. As shown in Table III the reaching performance is comparable to the 4 DoF case. Notably the 7 DoF setup has a larger energy consumption as the 4 DoF

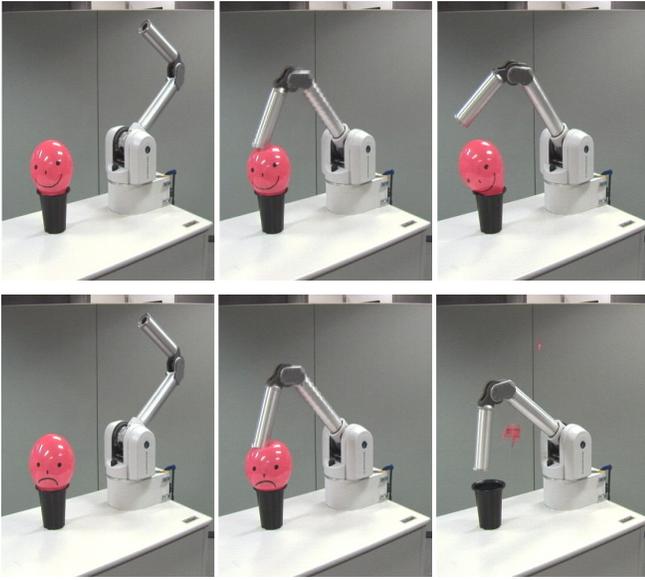


Fig. 7. Demonstration of the compliance of ILQG motion. Top row: The reaching with ILQG towards the center of the black bucket. The obstacle (balloon) cannot get damaged by the robot, due to the compliant motion of ILQG. In contrast the minimum jerk planning using standard gains is not compliant and therefore highly destructive.

arm. We attribute this to the higher weight of the additional motors and gearing (+2kg) that are added at the last 3 joints.

TABLE III

ILQG RESULTS (MEAN  $\pm$  STD OVER 10 TRIALS) ON 7 DOF WAM FOR 3 REFERENCE TARGETS.

ILQG	Euclidean target error [mm]	Energy [ $N \cdot m$ ]
$\mathbf{t}_{center}$	$5.494 \pm 0.150$	$16.254 \cdot 10^3 \pm 0.023 \cdot 10^3$
$\mathbf{t}_{left}$	$6.891 \pm 0.175$	$17.690 \cdot 10^3 \pm 0.017 \cdot 10^3$
$\mathbf{t}_{right}$	$9.210 \pm 0.156$	$16.272 \cdot 10^3 \pm 0.021 \cdot 10^3$

2) *Task space reaching with orientation*: Many manipulator tasks, for example pick and place tasks, require a specification of the end-effector orientation. Therefore instead of defining the reaching target in task space coordinates  $(x, y, z)$  only (eq. 15), we additionally specify the desired end-effector rotation as *yaw, pitch, roll*  $(y, p, r)$ . We set two reaching tasks towards  $\mathbf{t}_{center}$  with different end-effector orientations, one pointing horizontally to the front and the other pointing vertically down. The  $4 \times 4$  transformation matrices of the targets are

$$\mathbf{t}_{front} = \begin{pmatrix} 0 & 0 & 1.0 & 0.45 \\ 0 & 1.0 & 0 & 0 \\ -1.0 & 0 & 0 & 0.25 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{t}_{down} = \begin{pmatrix} -1.0 & 0 & 0 & 0.45 \\ 0 & 1.0 & 0 & 0 \\ 0 & 0 & -1.0 & 0.25 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Applied to the WAM (Fig. 8), it successfully reached the targets with high accuracy in position and orientation as can

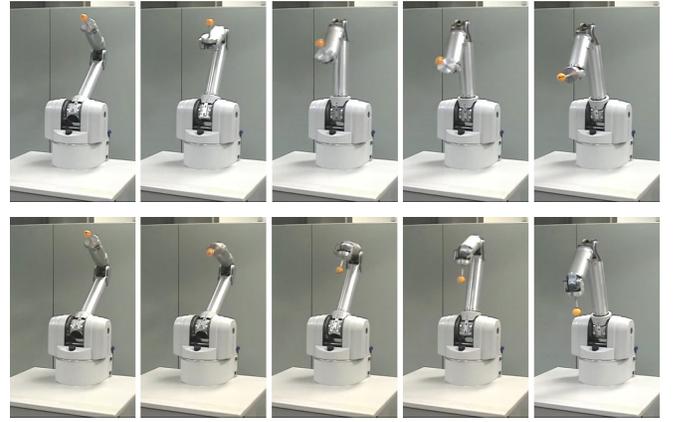


Fig. 8. ILQG reaching of 7 DoF WAM for targets with two different end point orientations. Top row:  $\mathbf{t}_{front}$ ; bottom row:  $\mathbf{t}_{down}$

be seen from the WAM's end-effector transformation matrix of the two targets (averaged over 10 trials).

$$\mathbf{r}(\mathbf{q}_K^{front}) = \begin{pmatrix} 0.0032 & 0.0020 & 1.0000 & 0.4585 \\ -0.0005 & 1.0000 & -0.0020 & -0.0003 \\ -1.0000 & -0.0005 & 0.0032 & 0.2631 \\ 0 & 0 & 0 & 1.0 \end{pmatrix}$$

$$\mathbf{r}(\mathbf{q}_K^{down}) = \begin{pmatrix} -1.0000 & -0.0052 & 0.0017 & 0.4448 \\ -0.0052 & 1.0000 & -0.0028 & -0.0018 \\ -0.0017 & -0.0028 & -1.0000 & 0.2502 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Table IV summarises the performance over 10 trials.

TABLE IV

ILQG RESULTS (MEAN  $\pm$  STD OVER 10 TRIALS) ON 7 DOF WAM FOR 2 REFERENCE TARGETS WITH ORIENTATION CONSTRAINTS .

ILQG	$\mathbf{t}_{front}$
Euclidean target error [mm]	$14.672 \pm 0.501$
Yaw error [rad]	$0.0024 \pm 0.0011$
Pitch error [rad]	$0.0014 \pm 0.0011$
Roll error [rad]	$0.0013 \pm 0.0005$
Energy [ $N \cdot m$ ]	$12.485 \cdot 10^3 \pm 0.0431 \cdot 10^3$
ILQG	$\mathbf{t}_{down}$
Euclidean target error [mm]	$5.538 \pm 0.329$
Yaw error [rad]	$0.0035 \pm 0.0027$
Pitch error [rad]	$0.0019 \pm 0.0012$
Roll error [rad]	$0.0064 \pm 0.0053$
Energy [ $N \cdot m$ ]	$13.271 \cdot 10^3 \pm 0.677 \cdot 10^3$

### C. Reducing the computational costs of ILQG

For any motion control algorithm, real-time planning is desirable and computational costs therefore play an important role. Given the high operation rate of the WAM (500 Hz), we face serious limitations due to the computational efficiency of iterative methods. These scale linearly with the number of time steps, linearly with the number of iterations and in the number of input dimensions  $\mathbf{x} = (\mathbf{q}; \dot{\mathbf{q}})$  and  $\mathbf{u}$  (i.e.,  $3N$  for an  $N$  DoF robot).

Typical ILQG simulations produce accurate optimisation results with  $dt = 0.01$ . Therefore when calculating the ILQG control law we do this initially with  $dt = 0.01$  to quickly obtain an optimal control sequence  $\bar{\mathbf{u}}$  of length  $n = 160$ . We

then subsample this optimal torque sequence to get longer control sequence  $\bar{\mathbf{u}}_{ext}$  of length  $n = 800$ . This sequence serves as the new initial control sequence of ILQG with  $dt = 0.005$  and since the sequence is located near the optimal solution already, ILQG converges after only 2 to 4 iterations on average.

In order to reduce the finite differences calculations one can employ analytic derivatives of the dynamics function. We have shown previously that such approaches can speed up the ILQG computations by a factor 10 or more [11]. Another practical speed-up approach limits the required number of iterations by remembering previously calculated optimal trajectories, which then can be used as an “initialisation library” near an expected optimum, performed for example with a nearest neighbour search. A similar approach has proved to create a globally optimal behaviour in a real-time simulated swimmer [15]. Applying the above mentioned speed-up methods we were able to perform complete ILQG computations for the 7 DoF WAM in the range of 2 to 5 seconds on a regular Notebook (*Intel Core 2 1.8GHz*). Notably these solutions were obtained using finite differences instead of analytic derivatives.

## V. CONCLUSION & OUTLOOK

In this paper we proposed to use OFC for the control of anthropomorphic robots. We developed a locally valid optimal feedback controller for the WAM manipulator, which we achieved by incorporating the physical constraints of the robot into the performance index of the optimal controller. We further elaborated on the problems and solutions of discontinuous dynamics as they occur on real hardware. We successfully tested our control method to the manipulator and demonstrated the practical benefits of this control strategy, that unifies motion planning, redundancy resolution and compliant control as a result of a single algorithm.

Like for any model-based control method, the dynamics model is the bottleneck in terms of task achievement and accuracy. Even though we sacrificed model accuracy to achieve numerical stability, we still are able to get reasonable reaching accuracy that can be thought to be sufficient for most daily life tasks.

Whereas here we studied reaching tasks, the OFC is easily extendable to more complex tasks such as throwing and hitting (see accompanying video), via point tasks, or walking [5]. We are currently working on extensions of OFC for the control of redundantly actuated (i.e., variable impedance) manipulators under the assumption of stochastic plant dynamics.

## REFERENCES

- [1] D. P. Bertsekas. *Dynamic programming and optimal control*. Athena Scientific, Belmont, Mass., 1995.
- [2] P. Dyer and S. McReynolds. *The Computational Theory of Optimal Control*. Academic Press, New York, 1970.
- [3] T. Flash and N. Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience*, 5:1688–1703, 1985.
- [4] C. M. Harris and D. M. Wolpert. Signal-dependent noise determines motor planning. *Nature*, 394:780–784, 1998.
- [5] Morimoto J. and Atkeson C. Minimax differential dynamic programming: An application to robust biped walking. In *Advances in Neural Information Processing Systems*, 15, 2002.
- [6] D. H. Jacobson and D. Q. Mayne. *Differential Dynamic Programming*. Elsevier, New York, 1970.
- [7] D. E. Kirk. *Optimal Control Theory: An Introduction*. Prentice-Hall, 1970.
- [8] W. Li. *Optimal Control for Biological Movement Systems*. PhD dissertation, University of California, San Diego, 2006.
- [9] W. Li and E. Todorov. Iterative linearization methods for approximately optimal control and estimation of non-linear stochastic system. *International Journal of Control*, 80(9):14391453, September 2007.
- [10] H. Mayeda, K. Osuka, and Kangawa A. A new identification method for serial manipulator arms. *Proc. 9th IFAC World Congress*, pages 2429–2434, 1984.
- [11] D. Mitrovic, S. Klanke, and S. Vijayakumar. Optimal control with adaptive internal dynamics models. In *In proceedings of the 5th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, Madeira, Portugal, 2008.
- [12] W. L. Nelson. Physical principles for economies of skilled movements. *Biological Cybernetics*, 46:135147, 1983.
- [13] S. H. Scott. Optimal feedback control and the neural basis of volitional motor control. *Nature Reviews Neuroscience*, 5:532–546, 2004.
- [14] R. F. Stengel. *Optimal control and estimation*. Dover Publications, New York, 1994.
- [15] Y. Tassa, T. Erez, and W. D. Smart. Receding horizon differential programming. In *Advances in Neural Information Processing Systems*, 2007.
- [16] E. Todorov. Optimality principles in sensorimotor control. *Nature Neuroscience*, 7(9):907–915, 2004.
- [17] E. Todorov and M. Jordan. A minimal intervention principle for coordinated movement. In *Advances in Neural Information Processing Systems*, number 15, pages 27–34. MIT Press, 2003.
- [18] E. Todorov and W. Li. A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *Proc. of the American Control Conference*, 2005.
- [19] Y. Uno, M. Kawato, and R. Suzuki. Formation and control of optimal trajectories in human multijoint arm movements: minimum torque-change model. *Biological Cybernetics*, 61:89–101, 1989.