Learning impedance control of antagonistic systems based on stochastic optimization principles

Djordje Mitrovic, Stefan Klanke and Sethu Vijayakumar

The International Journal of Robotics Research 2011 30: 556 originally published online 7 December 2010
DOI: 10.1177/0278364910387653

The International Journal of
Robotics Research
30(5) 556–573

DOI: 10.1177/0278364910387653
ijr.sagepub.com

Learning impedance control of antagonistic systems based on stochastic optimization principles

Djordje Mitrovic, Stefan Klanke and Sethu Vijayakumar

Abstract

Novel anthropomorphic robotic systems increasingly employ variable impedance actuation with a view to achieving robustness against uncertainty, superior agility and improved efficiency that are hallmarks of biological systems. Controlling and modulating impedance profiles such that they are optimally tuned to the controlled plant is crucial in realizing these benefits. In this work, we propose a methodology to generate optimal control commands for variable impedance actuators under a prescribed tradeoff of task accuracy and energy cost. We employ a supervised learning paradigm to acquire both *the plant dynamics and its stochastic properties. This enables us to prescribe an optimal impedance and command profile (i) tuned to the hard-to-model plant noise characteristics and (ii) adaptable to systematic changes. To evaluate the scalability of our framework to real hardware, we designed and built a novel antagonistic series elastic actuator (SEA) characterized by a simple mechanical architecture and we ran several evaluations on a variety of reach and hold tasks. These results highlight, for the first time on real hardware, how impedance modulation profiles tuned to the plant dynamics emerge from the first principles of stochastic optimization, achieving clear performance gains over classical methods that ignore or are incapable of incorporating stochastic information.*

Keywords

Antagonistic actuator, dynamics learning, equilibrium point control, impedance control, stochastic optimal control

1. Introduction

Humans have remarkable abilities in controlling their limbs in a fashion that outperforms most artificial systems in terms of versatility, compliance and energy efficiency. The fact that biological motor systems suffer from significant noise, sensory delays and other sources of stochasticity (Faisal et al. 2008) makes their performance even more impressive. Therefore, it comes as no surprise that biological motor control is often used as a benchmark for robotic systems. On the one hand, biological motor control characteristics are a result of the inherent biophysical properties of human limbs, and on the other hand, they are achieved through a framework of learning and adaptation processes (Wolpert et al. 1995; Kawato 1999; Davidson and Wolpert 2005). These concepts can be transferred to robotic systems by (i) developing appropriate anthropomorphic hardware and (ii) by employing learning mechanisms that support motor control in the presence of noise and perturbations (Mitrovic et al. 2008).

In this paper, we focus on issues related to adaptive motor control of antagonistically actuated robots. Antagonistic actuator designs are based on the biological principle of opposing muscle pairs. Therefore, the joint torque motors, for example, of a robotic arm are replaced by opposing actuators, typically using mechanical springs (Pratt and Williamson 1995). Such *series elastic actuators (SEA)* have had increasing attention in the last few decades (Vanderborght et al. 2009) as they provide several beneficial properties over classic joint torque actuated systems:

1. *Impedance control and variable compliance:* Through the use of antagonistic actuation, the system is able to vary co-contraction levels, which in turn change the system's mechanical properties: this is commonly referred to as *impedance control* (Hogan 1984). Impedance in a mechanical system is defined as a measure of force response to a motion exerted on the system and is

Institute for Perception, Action, and Behaviour, University of Edinburgh, Edinburgh, UK

Corresponding author:
Djordje Mitrovic
Institute for Perception, Action, and Behaviour, University of Edinburgh, 10 Crichton Street, Edinburgh EH8 9AB, UK.
Email: d.mitrovic@ed.ac.uk

composed of components such as *inertia*, *damping* and *stiffness*. In general SEAs can only vary stiffness of a system and achieving variable damping is technically challenging (e.g. Laffranchi et al. 2010). Consequently, in this paper, when we refer to *impedance control*, we will solely address a *change in stiffness* and ignore variable damping or variable inertia. Antagonistic actuation introduces an additional degree of freedom in the limb dynamics, i.e. the same joint torque can be achieved by different muscle activations. This means low co-contraction leads to low joint impedance whereas high co-contraction increases joint impedance. This degree of freedom can be used beneficially in many motion tasks, especially those involving manipulation or interaction with tools. It has been shown through many neurophysiological studies (e.g. Burdet et al. 2001) that humans are capable of modulating this impedance in an optimal way with respect to the task requirements, trading off selectively against energy consumption. For example, when you use a drill to drill holes in a wall, you *learn* how to co-contract your muscles such that the random perturbations of the drilling have a minimal impact. Furthermore, the ability to vary compliance plays a crucial role in robot safety (Zinn et al. 2004). In general, impedance modulation is an efficient way to control systems that suffer from noise, disturbances or sensorimotor delays.

2. *Energy efficiency and energy storage:* By appropriately controlling the SEA, one can take into account the passive properties of the springs and produce control strategies with low energy requirements. A well-known example is walking, where the spring properties combined with an ideal actuation timing can be used to produce energetically efficient gaits (Collins and Ruina 2005; Collins and Kuo 2010). Furthermore, an SEA has impressive energy storage and fast discharge capabilities, enabling "explosive" behavior such as throwing a ball (Wolf and Hirzinger 2008), which is quite hard to achieve with regular joint torque controllers. Therefore, series elasticity can amplify power and work output of an actuator, which is important in the fabrication of lightweight but powerful robotic or prosthetic devices (Paluska and Herr 2006).

A disadvantage of antagonistic actuation is that it imposes higher demands on the redundancy resolution capabilities of a motor controller. Optimality principles have successfully been used in biological (Flash and Hogan 1985; Scott 2004; Todorov 2004) and in artificial systems (Nakamura and Hanafusa 1987; Cortes et al. 2001) as a principled strategy to resolve redundancies in a way that is beneficial for the task at hand. More specifically, *stochastic optimal control (SOC)* (Stengel 1994; Bertsekas 1995; Todorov 2006) appears to be an especially appealing theory as it studies optimality principles under the premise of noisy and uncertain dynamics. Another important aspect when studying stochastic systems is how the information, for example, about noise or uncertainty is obtained without prior knowledge. Supervised learning methods can provide a viable solution to this problem as they can be used to extract information from the plant's sensorimotor data directly.

Here, we propose a control strategy for antagonistic systems that is based on stochastic optimal control theory under the premise of minimal energy cost. We propose to extend SOC by learning the dynamic model of the plant, which enables us (i) to adapt to systematic changes of the plant and (ii) extract its stochastic properties. Stochastic properties or stochastic information refers to noise or random perturbations of the controlled system that cannot be modeled deterministically. By incorporating this stochastic information into the optimization process, we show how impedance modulation and co-contraction behavior emerges as an optimal control strategy from first principles.

In the next section, we present a new antagonistic actuator, which serves as our implementation testbed for studying impedance control in the presence of stochasticity and which, compared to previous antagonistic designs, has a much simpler mechanical design. In Section 3, we introduce the basic concepts of optimal control and propose an extension that uses a learned dynamic model. This supervised learning methodology allows us to adapt online to changes in the dynamics as well as to extract localized stochastic information from movement data. We then propose a systematic methodology for incorporating deterministic and stochastic plant dynamic information into the optimal control framework, resulting in a scheme that improves performance significantly by exploiting the antagonistic redundancy of our plant. Our claims are supported by a number of experimental evaluations on real hardware in Section 4. We conclude the paper with a discussion and an outlook.

2. A Novel Antagonistic Actuator Design for Impedance Control

To study impedance control, we developed an antagonistic joint with a simple mechanical setup. Our design is based on the SEA approach in which the driven joint is connected via spring(s) to a stiff actuator (e.g. a servomotor). A variety of SEA designs have been proposed (for a recent review see Vanderborght et al. (2009)), which we here classify into *pseudo-antagonistic* and *antagonistic* setups. Pseudo-antagonistic SEAs have one or multiple elastic elements, which are connected between the driving motor and the driven joint. The spring tension and therefore, the joint stiffness, is regulated using a mechanism equipped with a second actuator. Antagonistic SEAs have one motor per opposing spring and the stiffness is controlled through a combination of both motor commands. Therefore, in antagonistic designs, the relationship between motor commands and stiffness must be resolved by the controller. This additional computational cost is the tradeoff for a biologically plausible architecture.

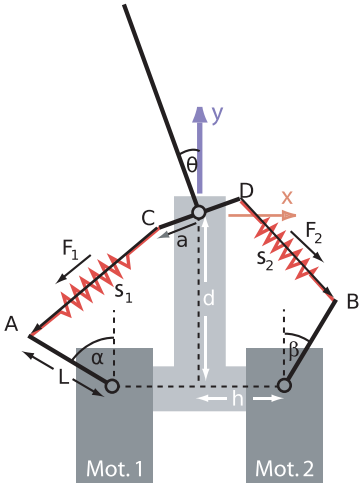

Fig. 1. Schematic of the variable stiffness actuator. The robot's dimensions are: $a = 15$ mm, $L = 26$ mm, $d = 81$ mm, $h = 27$ mm. The spring rest length is $s_0 = 27$ mm.

For antagonistic SEAs, non-linearity of the springs is essential to obtain variable compliance (van Ham et al. 2009). Because forces produced through springs with linear tension-to-force characteristics tend to cancel out in an antagonistic setup, an increase in the tension of both springs (i.e. co-contraction) does not change the stiffness of the system. Commercially available springs usually have linear tension-to-force characteristics and consequently most antagonistic SEAs require relatively *complex mechanical structures* to achieve a non-linear tension-to-force curve (Hurst et al. 2004; Migliore et al. 2005; Tonietti et al. 2005). These mechanisms typically increase construction and maintenance effort but also can complicate the system identification and controllability, for example, due to added drag and friction properties. We directly addressed this aspect in our design of the SEA, which aims to achieve variable stiffness characteristics using a simple mechanical setup.

2.1. Variable Stiffness with Linear Springs

Here we propose an SEA design that does not rely on complex mechanisms to achieve variable stiffness but achieves the desired properties through a specific geometric arrangement of the springs. While the emphasis of this paper is not on the mechanical design of actuators, we will explain the essential dynamic properties of our testbed. Figure 1 shows a sketch of the robot, which is mounted horizontally and consists of a single joint and two antagonistic servomotors that are connected to the joint via *linear springs*. The springs are mounted with a moment arm offset a at the joints and an offset of L at the motors. Therefore, the spring-endpoints move along circular paths at the joints and at the motors. Under the assumption that the servomotors are infinitely stiff, we can calculate the torque τ acting on the arm as follows. Let $\mathbf{s}_1$ denote the vector from point C to A, and $\mathbf{s}_2$ the vector from D to B, and s_1 and s_2 their respective lengths. Putting the origin of the coordinate system at the arm joint, we have

$$\mathbf{s}_1 = \begin{pmatrix} -h - L\sin\alpha \\ -d + L\cos\alpha \\ 0 \end{pmatrix} - \underbrace{\begin{pmatrix} -a\cos\theta \\ -a\sin\theta \\ 0 \end{pmatrix}}_{=\mathbf{a}_1},$$

$$\mathbf{s}_2 = \begin{pmatrix} h + L\sin\beta \\ -d + L\cos\beta \\ 0 \end{pmatrix} - \underbrace{\begin{pmatrix} a\cos\theta \\ a\sin\theta \\ 0 \end{pmatrix}}_{=\mathbf{a}_2}. \quad (1)$$

Denoting the spring constant by κ and the rest length by s_0, this yields forces

$$\mathbf{F}_1 = \kappa(s_1 - s_0)\frac{\mathbf{s}_1}{s_1} \quad \text{and} \quad \mathbf{F}_2 = \kappa(s_2 - s_0)\frac{\mathbf{s}_2}{s_2}. \quad (2)$$

Given the motor positions α and β and the arm position θ, the torque generated by the springs is

$$\tau(\alpha,\beta,\theta) = \hat{\mathbf{z}}^{\mathrm{T}}(\mathbf{F}_1 \times \mathbf{a}_1 + \mathbf{F}_2 \times \mathbf{a}_2), \quad (3)$$

where $\hat{\mathbf{z}}^{\mathrm{T}}$ denotes the three-dimensional basis vector $(0, 0, 1)^{\mathrm{T}}$. To calculate the equilibrium position θ_{eq} for given motor positions α and β, we need to solve $\tau(\alpha,\beta,\theta_{\mathrm{eq}}) = 0$, which in practice is by numerical optimization. At this position, we can calculate the joint stiffness as

$$K(\alpha,\beta) = \frac{\partial}{\partial\theta}\tau(\alpha,\beta,\theta)\Big|_{\theta=\theta_{\mathrm{eq}}}. \quad (4)$$

Note that K depends linearly on the spring stiffness κ, but the geometry of the arm induces a non-linear dependency on α and β. Figure 2 shows the computed profiles of the equilibrium position and stiffness, respectively.

Further, denoting the arm's inertia around the z-axis by I_z and a damping torque given by $\tau(\dot\theta) = -D\dot\theta$, the dynamic equation can be analytically written as

$$I_z\ddot\theta = \tau(\alpha,\beta,\theta) - D\dot\theta. \quad (5)$$

2.2. Actuator Hardware

Figure 3 depicts our prototype SEA hardware implementation of the discussed design. For actuation, we employ two servomotors (Hitec HSR-5990TG), each of which is connected to the arm via a spring mounted on two low friction ball bearings. To avoid excessive oscillations, the joint is attached to a rotary viscous damper. The servos are controlled using 50 Hz PWM signals by an Arduino Duemilanove microcontroller board (Atmel ATmega328). That board also measures the arm's joint angle θ with a contact-free rotary position encoder (Melexis MLX90316GO), as well as its angular acceleration $\ddot\theta$ using a LilyPad accelerometer (Analog Devices ADXL330). Finally, we also measure the servomotor positions by feeding a signal

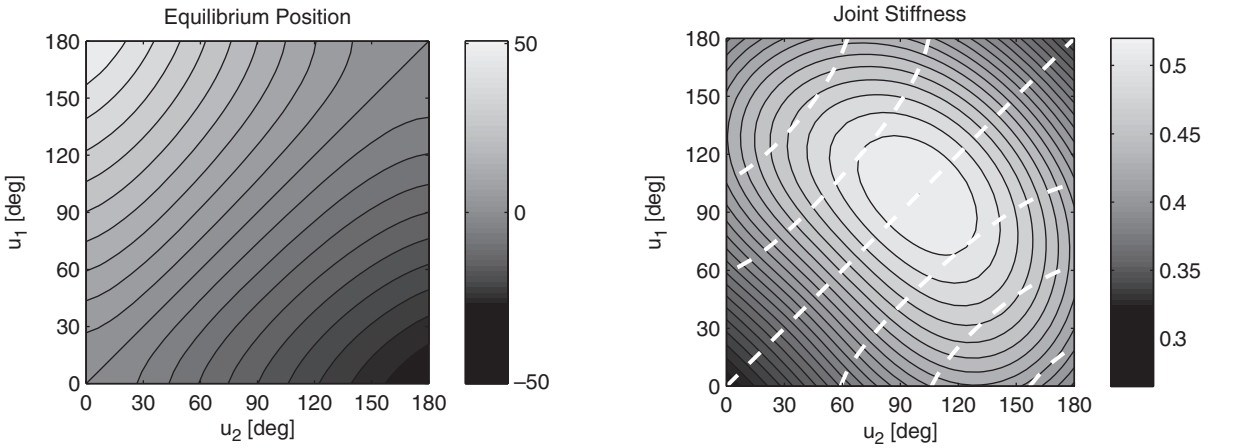

Fig. 2. Left: Equilibrium position as a function of the motor positions (in degrees), with contour lines spaced at 5° intervals. Right: Stiffness profile of the arm, as calculated from Equation (4). The maximum achievable stiffness is 150% of the intrinsic spring stiffness.

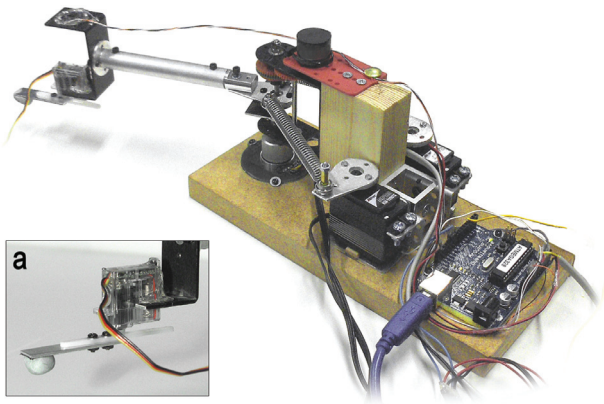

Fig. 3. Photograph of our antagonistic robot. Inset panel (a): Separate servomotor mounted at the end of the arm to create stochastic perturbations (see Section 4.2).

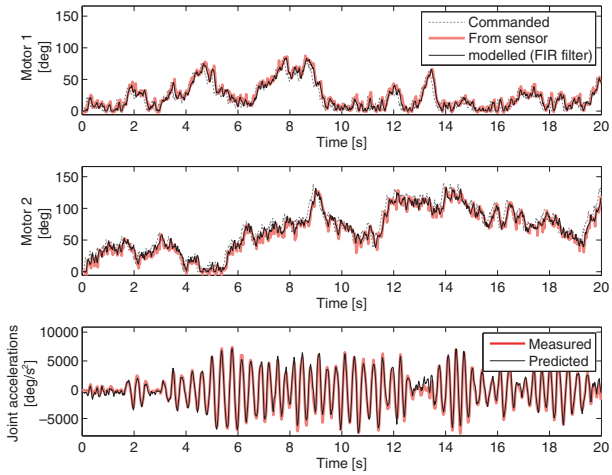

Fig. 4. Comparison of prediction of performance of estimated motor dynamics (top and middle) and of arm dynamics (bottom) for an independent test data set.

from their internal potentiometer to the AD converters of the Arduino. While the operating frequency is limited to 50 Hz due to the PWM control, all measurements are taken at a 4× higher frequency and averaged on the board to reduce the amount of noise, before sending the results to a PC via a serial connection (RS232/USB).

2.3. System Identification

Apart from measuring the exact dimensions ($L = 2.6$ cm, $a = 1.5$ cm, $h = 2.7$ cm, $d = 8.1$ cm) of the robot, and the stiffness constant of the spring ($\kappa = 424$ N m^{-1}), system identification consists of a series of steps, each of which involves a least-squares fit between known and actually measured quantities.

1. *Identify servomotor dynamics:* The servomotors are controlled by sending the desired position (encoded as a PWM signal), which we refer to as u_1 and u_2 for motors 1 and 2, respectively. Even though the servomotors we use are very accurate, they need some time to reach the desired position, and therefore we model the true motor positions (α, β) as a low-pass filtered version of the commands (u_1, u_2) using a finite impulse response (FIR) filter, i.e.

$$\alpha[n] = (h * u_1)[n] + \epsilon[n] = \sum_{k=0}^{K} h[k]u_1[n-k] + \epsilon[n] \tag{6}$$

and similarly for β and u_2. The term $\epsilon[t]$ denotes a noise component of the true motor position that cannot be modeled with the FIR filter. By using the internal potentiometer of the servomotors, we can measure the actual motor positions to identify the filter coefficients h_i using a least squares fit, that is, by minimizing $\sum_t (\alpha[n] - (h * u_1)[n])^2$ with respect to h_i. We retrieved a good fit of the motor dynamics (cf. Figure 4) using an

FIR filter with seven steps, with estimated coefficients $h = [0, 0, 0, 0.0445, 0.2708, 0.3189, 0.3658]$.

2. *Calibrate position sensor:* Tests with the position sensor revealed linear position characteristics. By moving the arm physically to several predefined and geometrically measured positions, we determined the sensor's offset and slope.
3. *Calibrate acceleration sensor:* We matched the accelerations measured with the accelerometer with accelerations derived from the position sensor (using finite differences).
4. *Collect training data and fit parameters:* We carried out motor babbling (any excitation movements are applicable) on the servos and measured the resulting arm positions, velocities and accelerations. Taking into account the estimated motor dynamics using the fitted filter, we estimated the arm's inertia ($I_z = kg^* m^{2^*} \text{rad}^{(-2)}$) and viscous damping ($D = N^* m^* s^* \text{rad}^{(-1)}$) coefficient using least squares from Equation (5).

On a large independent test set of $S_{\text{test}} = 300,000$ data points, the motor prediction produces a Normalized Mean Squared Error (NMSE) of $e_{\text{nmse}} = 1.85\%$. Figure 4 shows an example prediction of performance for a sequence of random motor commands (20 s from the test set S_{test}) using the estimated dynamic model.

3. Stochastic Optimal Control

In many control scenarios it is desirable to be able to perform in the "best way possible". For example, one may wish to move the system to a desired posture and consume as little energy as possible during the movement. This type of problem is studied in *optimal control* theory, the central ingredient of which is the minimization of an optimality criterion

$$J[\mathbf{u}] = \int_0^T c(\mathbf{x}(t), \mathbf{u}(t), t)\, \mathrm{d}t + h(\mathbf{x}(T)) \quad \text{or}$$

$$J[\mathbf{u}] = \int_0^\infty c(\mathbf{x}(t), \mathbf{u}(t), t)\, \mathrm{d}t, \tag{7}$$

for a task with a finite or infinite horizon. Apart from the optional final cost $h(\cdot)$, the criterion integrates a cost rate $c(\mathbf{x}, \mathbf{u})$ over the course of the movement. That cost may depend on the system's state $\mathbf{x}$, control commands $\mathbf{u}$ and on time t, where the initial state of the system is given as $\mathbf{x}(0)$, and $\mathbf{x}(t)$ evolves depending on the commands $\mathbf{u}(t)$. In the context of biological motor control, this theory has been studied for decades with the well-known examples of various optimality criteria such as minimum time (Bobrow et al. 1985), energy (Li and Todorov 2007), jerk (Flash and Hogan 1985) and torque change (Uno et al. 1989).

For a system with deterministic (and accurately modeled) dynamics $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$, it is sufficient to find the *open-loop* sequence of commands $\mathbf{u}(t)$ and the associated trajectory $\mathbf{x}(t)$ that minimizes J, which can usually be obtained by solving a two-point boundary difference/differential equation derived by applying *Pontryagin's minimum principle* (Stengel 1994). In practice, in the presence of small perturbations or modeling errors, the optimal open-loop sequence of commands can be run on the real plant together with a simple PD controller that corrects deviations from the planned trajectory. However, those corrections will usually not adhere to the optimality criterion, and the resulting cost J will be higher.

Alternatively, we can try to incorporate stochasticity, e.g. as a dynamic model

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u})\, \mathrm{d}t + \mathbf{F}(\mathbf{x}, \mathbf{u})\, \mathrm{d}\xi, \quad \xi \sim \mathcal{N}(0, \mathbf{I}) \tag{8}$$

directly into the optimization process and minimize the *expected* cost.[1] Here, $\mathrm{d}\xi$ is a Gaussian noise process and $\mathbf{F}(\cdot)$ tells us how strongly the noise affects each part of the state and control space. A well-studied example of this case is the LQG problem, which stands for *linear* dynamics ($\mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$), *quadratic* cost (in both $\mathbf{x}$ and $\mathbf{u}$), and *Gaussian* noise ($\mathbf{F}$ is constant). A solution to this class of problems is the *optimal feedback controller* (OFC), that is, a policy $\mathbf{u} = \pi(\mathbf{x}, t)$ that calculates the optimal command $\mathbf{u}$ based on feedback $\mathbf{x}$ from the real plant. In the LQG case, the solution is a linear feedback law $\mathbf{u} = \mathbf{L}(t)\,\mathbf{x}$ with precomputed time-dependent gain matrices[2] $\mathbf{L}(t)$ (Stengel 1994).

Solving OFC problems for more complex systems (non-linear dynamics, non-quadratic cost, varying noise levels $\mathbf{F}$) is a difficult computational task. A general way to solve OFC problems for non-linear quadratic problems is *Dynamic Programming* (DP) (Bellman 1957). DP in its basic form relies on a discretization of the state and action space, which in practice is difficult to obtain: On the one hand, tiling the state–action space too sparse will lead to poor representation of the underlying plant dynamics. On the other hand, a very fine discretization leads to a combinatorial explosion of the problem, which is commonly referred to as the *curse of dimensionality*. For example, consider a discretization of 100 steps for each variable of the state and action space. In the case of the presented SEA, this corresponds to a state space dimensionality $n = 2$, for positions and velocities, and action space dimensionality $m = 2$, for the two motors.[3] Even for this low-dimensional system the possible combinations of states and actions that DP needs to evaluate and store in order to find the optimal control law are $p = 100^4 = 100{,}000{,}000$. One way to avoid the curse of dimensionality is to restrict the state space to a region that is close to a nominal optimal trajectory. In the neighborhood of such trajectories the DP problem can be approximated analytically using Taylor expansions of the dynamics and the cost function. The idea is to compute an optimal trajectory together with a locally valid feedback law and then iteratively improve this nominal solution until convergence. Well-known examples of such iterative methods are *Differential Dynamic Programming* (DDP) (Dyer and McReynolds 1970; Jacobson and Mayne 1970) or the more recent *iterative Linear Quadratic Gaussian* (ILQG)

(Todorov and Li 2005), which will serve as solution technique of choice in this paper. ILQG yields both an open-loop sequence of commands and optimized feedback gain matrices, but these are not guaranteed to converge towards the global optimum: Depending on the initial guess of the trajectory, the iterative improvement might result in a solution with only a locally optimal expected cost J.

3.1. Modeling Dynamics and Noise through Learning

Analytical dynamic formulations, as described in Section 2.1 or in Equation (5), have the tremendous advantage of being compact and fast to evaluate numerically, but they also suffer from drawbacks. First, their accuracy is limited to the level of detail put into the physical model. For example, our model is based on the assumption that the robot is completely symmetric, that both motors are perfectly calibrated, and that the two springs are identical, but in reality we cannot avoid small errors in all of these. Second, the analytical model does not provide obvious ways to model changes in the dynamics, such as from wear and tear, or more *systematic* changes due to the weight of an added tool.

While these problems can to some extent be alleviated by a more involved and repeated system identification process, the situation is more difficult if we consider the noise model $\mathbf{F}(\cdot)$, or *stochastic* changes to the dynamics. For example, an arm might be randomly perturbed by tool interactions such as when drilling into a wall, with stronger effects for certain postures, and milder effects for others. It is not obvious how one can model state dependent noise analytically.

We therefore propose to include a supervised learning component and to acquire both the dynamics and the noise model in a data-driven fashion (Figure 5). Our method of choice in this paper is *Locally Weighted Projection Regression* (LWPR) or (Vijayakumar et al. 2005), because that algorithm allows us to adapt the models incrementally and online, and it is able to reflect *heteroscedastic*[4] noise in the training data through localized confidence intervals around its predictions. More details on learning with LWPR can be found in Appendix A.

In order to simplify the presentation as much as possible, and also due to technical challenges of operating on the real hardware (for details see Section 5), in this work we learn the stochastic mapping $f(\mathbf{u})$ from motor positions to joint angle θ, not taking into account velocities and accelerations. During stationary conditions and in the absence of perturbations, this mapping reflects the equilibrium position of the arm (Figure 2, left). In correspondence to the general dynamic equation (8), here the state $\mathbf{x} = \theta_{\text{eq}}$ represents the current equilibrium position, $\mathbf{u}$ the applied motor action, and $\mathrm{d}\mathbf{x}$ the resulting change in equilibrium position. Therefore the reduced dynamics used here, only depends on the control signals, i.e.

$$\mathrm{d}x = f(\mathbf{u})\,\mathrm{d}t + F(\mathbf{u})\,\mathrm{d}\xi, \quad \xi \sim \mathcal{N}(0,1). \tag{9}$$

Learning this mapping from data, we can directly account for asymmetries. More interestingly, when we collect data from the perturbed system, we can acquire a model of the arm's kinematic variability as a function of the motor positions.

We use this learned model $\tilde{f}$ in two ways: first, in (slow) position control tasks (Section 3.2), and in conjunction with full analytic dynamic models for dynamic reaching tasks (Section 3.3).

3.2. Energy Optimal (Equilibrium) Position Control

Consider the task of holding the arm at a certain position $\hat{\theta}$, while consuming as little energy as possible. Let us further assume that we have no feedback from the system,[5] but that the arm is perturbed randomly. We can state this mathematically as the minimization of a cost

$$J = \langle w_p (f(\mathbf{u}) - \hat{\theta})^2 + |\mathbf{u}|^2 \rangle, \tag{10}$$

where w_p is a factor that weights the importance of being at the right position against the energy consumption, which for simplicity we model by $|\mathbf{u}|^2$. Taking into account that the motor commands $\mathbf{u}$ are deterministic, and decomposing the expected position error into an error of the mean plus the variance, we can write the expected cost J as

$$J = w_p \left(\langle f(\mathbf{u}) \rangle - \hat{\theta} \right)^2 + w_p \left\langle \left(f(\mathbf{u}) - \langle f(\mathbf{u}) \rangle \right)^2 \right\rangle + |\mathbf{u}|^2, \tag{11}$$

which based on the LWPR learned model becomes

$$J = w_p (\tilde{f}(\mathbf{u}) - \hat{\theta})^2 + w_p \sigma^2(\mathbf{u}) + |\mathbf{u}|^2. \tag{12}$$

Here $\tilde{f}(\mathbf{u})$ and $\sigma(\mathbf{u})$ denote the prediction and the one-standard-deviation-based confidence interval of the LWPR model of $f(\mathbf{u})$. The constant w_p represents the importance of the accuracy requirements in our task. We then can easily minimize J with respect to $\mathbf{u} = (u_1, u_2)^{\mathrm{T}}$ numerically, taking into account the box constraints $0^\circ \leq u_i \leq 180^\circ$.[6]

3.3. Dynamic Control with Learned Stochastic Information

Equilibrium position control is ignorant about the dynamics of the arm, that is, going from one desired position to the next might induce swinging movements, which are not damped out actively. Proper dynamic control should take these effect into account and optimize the command sequence accordingly. What follows is a description of how we model the full dynamics of the arm, that is, the combination of the dynamics of the joint and the motors.

The state vector $\mathbf{x}[k]$ of our system at time k consists of the joint angle $x_1[k] = \theta[k]$ and joint velocity $x_2[k] = \dot{\theta}[k]$ as well as 12 *additional* state variables, which represent the command history of the two motors, i.e. the last six motor

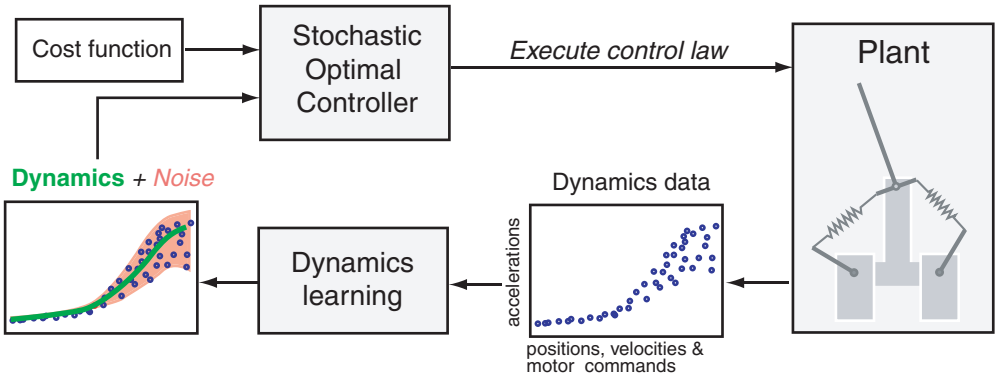

Fig. 5. Schematic diagram of our proposed combination of stochastic optimal control (SOC) and learning. The dynamic model used in SOC is acquired and constantly updated with data from the plant. The learning algorithm extracts the dynamics as well as stochastic information contained (noise model from confidence intervals). SOC takes into account both measures in the optimization.

commands that were applied to the system. The state vector, therefore, is

$$\mathbf{x}[k] = (\theta[k], \dot{\theta}[k], u_1[k-1], \ldots, u_1[k-6], u_2[k-1], \ldots, u_2[k-6])^{\mathrm{T}}, \quad (13)$$

where the additional state variables $x_3[k], \ldots, x_8[k]$ for motor 1, and similarly, $x_9[k], \ldots, x_{14}[k]$ for motor 2, are required to represent the FIR filter states of the motor dynamics from Equation (6). We can estimate the motor positions $\alpha[k]$ and $\beta[k]$ solely from these filter states because the FIR coefficients are $h_0 = h_1 = 0$:

$$\alpha[k] = \sum_{j=2}^{7} h_j u_1[k-j+1] = \sum_{j=2}^{7} h_j x_{j+1}[k] \quad (14)$$

$$\beta[k] = \sum_{j=2}^{7} h_j u_2[k-j+1] = \sum_{j=2}^{7} h_j x_{j+7}[k]. \quad (15)$$

Based on the rigid body dynamics from Equation (5) we can compute the acceleration from states (i.e. forward dynamics) as

$$\ddot{\theta}[k] = \frac{1}{I_z}\left(\tau(\alpha[k], \beta[k], \theta[k]) - D\dot{\theta}[k]\right). \quad (16)$$

Therefore "running" the dynamics here means accounting for motor dynamics by shifting the filter states, that is $x_{i+1}[k+1] = x_i[k]$ for $i = 3, \ldots, 7$ and $i = 9, \ldots, 13$, and then Euler-integrating the velocities and accelerations:

$$\mathbf{x}[k+1] = \mathbf{x}[k] + \Delta t\, \mathbf{f}(\mathbf{x}[k], \mathbf{u}[k]) \quad (17)$$

$$= (\theta[k] + \Delta t\dot{\theta}[k], \dot{\theta}[k] + \Delta t\ddot{\theta}[k], u_1[k], x_3[k], \ldots, x_7[k], u_2[k], x_9[k], \ldots, x_{13}[k])^{\mathrm{T}}. \quad (18)$$

Alternatively, we can drop the time index k and write the dynamics in compact form as

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \left(x_2, \ddot{\theta}(\mathbf{x}), \frac{1}{\Delta t}(u_1 - x_3), \frac{1}{\Delta t}(x_3 - x_4), \ldots, \frac{1}{\Delta t}(u_2 - x_8), \frac{1}{\Delta t}(x_8 - x_9), \ldots\right)^{\mathrm{T}}. \quad (19)$$

The gradient of $\ddot{\theta}(\mathbf{x})$ is given by the chain rule, where τ is the short notation for $\tau(\alpha, \beta, \theta)$. Note that $\theta = x_1$, $\dot{\theta} = x_2$, and α and β are calculated from $x_{3\ldots14}$:

$$\nabla_{\mathbf{x}}\ddot{\theta} = \frac{1}{I_z}\left(\frac{\partial\tau}{\partial\theta}, -D, \frac{\partial\tau}{\partial\alpha}h_2, \frac{\partial\tau}{\partial\alpha}h_3, \ldots, \frac{\partial\tau}{\partial\alpha}h_7, \frac{\partial\tau}{\partial\beta}h_2, \frac{\partial\tau}{\partial\beta}h_3, \ldots, \frac{\partial\tau}{\partial\beta}h_7\right). \quad (20)$$

This only shows the second row of the Jacobian $\nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x}, \mathbf{u})$ and for brevity we omitted the others as they are trivial. The other Jacobian $\nabla_{\mathbf{u}}\mathbf{f}(\mathbf{x}, \mathbf{u})$ consists of zero entries apart from the entry $1/\Delta t = 50$ at indices $(3, 1)$ and $(9, 2)$.

Since the dynamics of our system is non-linear and high-dimensional, we have to employ an iterative local optimization approach. We employ the ILQG method due to its ability to include constraints on the commands. More details of the ILQG algorithm can be found in Appendix B.

The usual ILQG formulation is based on an analytically given cost function (deterministic) and a stochastic dynamic function. Here we use a deterministic dynamics (with the idealized analytic model) and we propose a cost function that takes stochastic information into account.

$$c(\mathbf{x}, \mathbf{u}) = w_p(x_1 - \hat{\theta})^2 + w_v x_2^2 + w_e\,|\,\mathbf{u}\,|^2 + w_d((u_1 - x_3)^2 + (u_2 - x_9)^2) + w_p\sigma^2(\mathbf{u}). \quad (21)$$

All quantities in Equation (21) (also possibly the prefactors) are time-dependent, but we have dropped the time indices for notational simplicity. As before w_p governs the accuracy requirement. In addition, a stability term w_v governs the importance of having zero velocity and w_e penalizes energy consumption at the level of springs. The weighting factor w_d penalizes changes in motor commands and therefore energy consumption at the level of the servomotor. The last term includes the learned uncertainty in our equilibrium positions, which is here also scaled by w_p. This is justified because, for example, for a reaching task, the arm will finish with the servomotors in a position such that the arm's equilibrium position is the desired position $\hat{\theta}$, and

we have learned from data how much perturbation we can expect at any such configuration. The same holds true for slow tracking tasks, where the servos will be moved such that the equilibrium positions track the desired trajectory.

4. Results

In this section we present results from the optimal control model applied to the hardware described earlier in Section 2. We first highlight the adaptation capabilities of this framework experimentally and then show how the learned stochastic information leads to an improved control strategy over solutions obtained without stochastic information. More specifically the new model achieves higher positional accuracy by varying impedance of the arm through motor co-contraction. We study position holding, trajectory tracking and target reaching tasks.

4.1. Experiment 1: Adaptation towards a Systematic Change in the System

An advantage of the learned dynamic paradigm is that it allows us to account for systematic changes without prior knowledge of the shape or source of the perturbation. To demonstrate such an adaptation scenario we set up a systematic change in the hardware by replacing the left spring, between motor 1 and the joint (i.e. between points A and C in Figure 1), with one that has a lower, "unknown" spring constant. The aim is to hold a certain equilibrium position using the energy optimal position controller described in Section 3.2. As expected, the prediction about the equilibrium points (i.e. $\tilde{f}(\mathbf{u})$) does not match the real, changed system properties.

Next, we demonstrate how the system can adapt online and increase performance, trial by trial. We specified a target trajectory that is a linear interpolation of 200 steps between the start position $\theta_0 = -30^\circ$ and the target position $\hat{\theta} = 30^\circ$. We tracked this trajectory by recomputing the equilibrium positions, i.e. by minimizing Equation (12) at a rate of 50 Hz. At the same time we updated $\tilde{f}(\mathbf{u})$ during reaching. Due to the nature of local learning algorithms, $\tilde{f}$ is only updated in the neighborhood of the current trajectory and therefore shows limited generalization. To account for this, after each trial, we additionally updated the model with 400 training data points, collected from a 20×20 grid of the motor's range $u_1 = u_2 = [0^\circ, 180^\circ]$. Figure 6 depicts the outcome of this adaptation experiment. One can observe that the controller initially (lighter lines) fails to track the desired trajectory (red). However, there is significant improvement between each trial, especially between trials 1 to 5. After about nine trials the internal model has been updated and manages to track the desired trajectory well (up to the hardware's level of precision). The equilibrium position predictions in Figure 7 confirm that the the systematic shift has been successfully learned, as shown by the asymmetric shape. Analyzing the motor commands (Figure 6 right) shows that the optimal controller, for all trials, chooses the motor commands with virtually no co-contraction. This is a sensible choice as co-contraction would contradict the minimum energy cost function that we have specified.

4.2. The Role of Stochastic Information for Impedance Control

Because co-contraction and energy consumption are opposing properties, our controller will hardly make use of the redundant degree of freedom in the actuation. Even though minimum energy optimal control in an antagonistic system seems to be "unable to co-contract" it remains our favorite choice of performance index as it also implies compliant movement and as it follows the biological motivation. If we consider the stochastic information that would arise from a task involving random perturbations, we can see that the produced stochasticity holds valuable information about the stability of the system.[7] If the uncertainty can be reduced by co-contracting it will be reflected in the data, i.e. in the LWPR confidence bounds. Therefore the answer to the previous question is that, given that we wish to achieve high task accuracy, the controller should co-contract whenever it can reduce the expected noise/stochasticity in the system (weighted with the accuracy requirement).

Suppose our system experiences some form of small random perturbations during control. In the hardware we realize such a scenario by adding a perturbation motor at the end of the arm, which mimics, for example, a drilling tool (panel "a" in Figure 3). The perturbation on the arm is produced by alternating the servomotor positions quickly every 200 ms from 40° to -40°. The inertia of the additional weight then produces deflections of the arm from the current equilibrium position. With these perturbations, we collected new training data and updated the existing LWPR model $\tilde{f}$. The collected data reveals that the arm stabilizes in regions with higher co-contraction, where the stiffness is higher. This behavior is illustrated in Figure 8, which shows motion traces around $\theta = 0^\circ$ due to the perturbation motor for different co-contraction levels. This information is contained in the learned confidence bounds (Figure 9) and, therefore, the optimal controller effectively tries to find the tradeoff between accuracy and energy consumption.

4.3. Experiment 2: Impedance Control for Varying Accuracy Requirements

Based on the learned LWPR model $\tilde{f}$ from the previous section, we can demonstrate the improved control behavior of the stochastic optimization with emerging impedance control. We formulate a task to hold the arm at the fixed positions $\hat{\theta} = 15^\circ$ and $\hat{\theta} = 0^\circ$, respectively. While minimizing for the cost function in Equation (12), we continuously and slowly increased the position penalty within the range

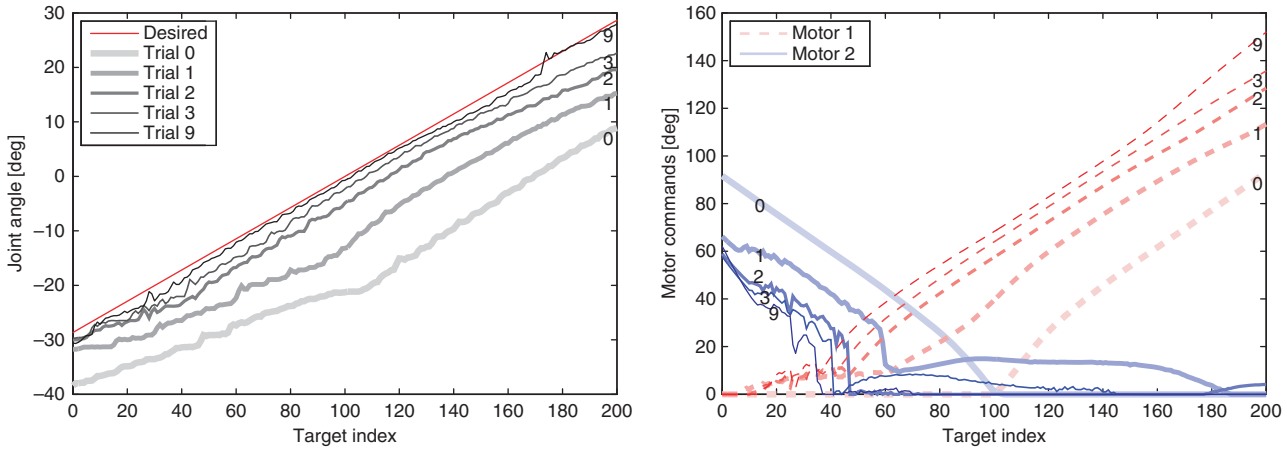

Fig. 6. Visualization of the adaptation process. Left: Desired (red) and observed arm positions. Right: Motor commands for the corresponding trials. Darker and thinner lines indicate later stages of learning.

$w_p = [10^{-2}, 10^5]$. The left column in Figure 10 summarizes the results we discuss next: At $w_p = 10^{-2}$ to approximately $w_p = 10^0$ the optimization neglects position accuracy and minimizes mainly for energy, i.e. $u_1 = u_2 = 0$. The actual joint positions, because of the perturbations, oscillate around the mean $\theta = 0^\circ$ as indicated by the shaded area. Between $w_p = 10^0$ and $w_p = 10^2$ the position constraint starts to "catch up" with the energy constraint; a shift in the mean position towards $\hat{\theta}$ can be observed. At about $w_p = 5 \cdot 10^1$, the variance in the positions increases as the periodic perturbation seems to coincide with the resonance frequency of the system. For $w_p > 10^2$ the stochastic information is weighted sufficiently such that the optimal solution increases the co-contraction and the accuracy improves further.

In contrast, if we run the same experiment while ignoring the stochastic part of the cost function, i.e. we minimize for the deterministic cost function $J = w_p(\tilde{f}(\mathbf{u}) - \hat{\theta})^2 + |\mathbf{u}|^2$ only, we can can see (Figure 11) that the system does, as expected, not co-contract and hardly improves performance accuracy.

4.4. Experiment 3: ILQG-Reaching Task with a Stochastic Cost Function

For certain tasks, such as quick target reaching or faster tracking of trajectories, the system dynamics based on equilibrium points $\theta = f(\mathbf{u})$ may not be sufficient, as it contains no information about the velocities and accelerations of the system. Next, we assume a full forward dynamic description of our system as identified in Equation (13), where the state consists of joint angles, joint velocities, and twelve motor states.

The task is to start at position $\theta_0 = 0^\circ$ and reach towards the target $\hat{\theta} = 0.3$ rad $(= 17.18^\circ)$. The reaching movement duration is fixed at 2 s, which corresponds to $T = 100$ discretized time steps at the hardware's operation rate of 50 Hz. This task can be formalized based on the cost function (21) by setting the weighting terms as follows: the time-dependent position penalty is a monotonically increasing linear interpolation of 100 steps, i.e. $w_p[t] = [0.1, 0.2, \ldots, 10]$. The penalty for zero endpoint velocity was set to $w_v[t] = 0$ for $0 < t < 80$ and $w_v[t] = 1$ for $t \geq 80$. The energy penalties are assumed constant $w_e = w_d = 1$ during the whole movement.

By using ILQG, we then compute an optimal control sequence $\bar{\mathbf{u}}$ with the corresponding desired trajectory $\bar{\mathbf{x}}$ and a feedback control law $\mathbf{L}$. Figure 12 depicts the reaching performance of the ILQG trajectory, applied in *open-loop* mode and in *closed-loop* mode (i.e. using feedback law $\mathbf{L}$), where the robot has been perturbed by a manual push. The closed-loop scheme successfully corrects the perturbation and reaches the target while the open-loop controller oscillates and fails to reach the target. This experiment highlights the benefits of closed-loop optimization which can, by incorporating the full dynamic description of the system, account for such perturbations. However, the ability to correct perturbations is limited by the hardware control bandwidth (i.e. slow servomotor dynamics and 50 Hz control board frequency). If the system also suffers from feedback or motor delays the correction ability is limited and for example accounting for vibrations or noise[8] is difficult to achieve using feedback signals only. For such stochastic perturbations, impedance control can improve performance as it changes the mechanical properties of the system in a feed-forward manner, i.e. it reduces the effects of the perturbations in the first place.

To realize such a scenario, we defined a tracking task that starts at the zero position then moves away and back again along a sinusoidal curve for 2.5 s. The cost function parameters for this task are defined as follows: The time-dependent position penalty is $w_p[t] = [50, 100, \ldots, 4{,}000]$ for $0 < t < 80$ and $w_p[t] = 4{,}000$ for $t \geq 80$. The endpoint velocity term is $w_v[t] = 0$ for $0 < t < 80$ and $w_v[t] = 10$ for $t \geq 80$. The energy penalties are held constant, i.e. $w_e = w_d = 1$.

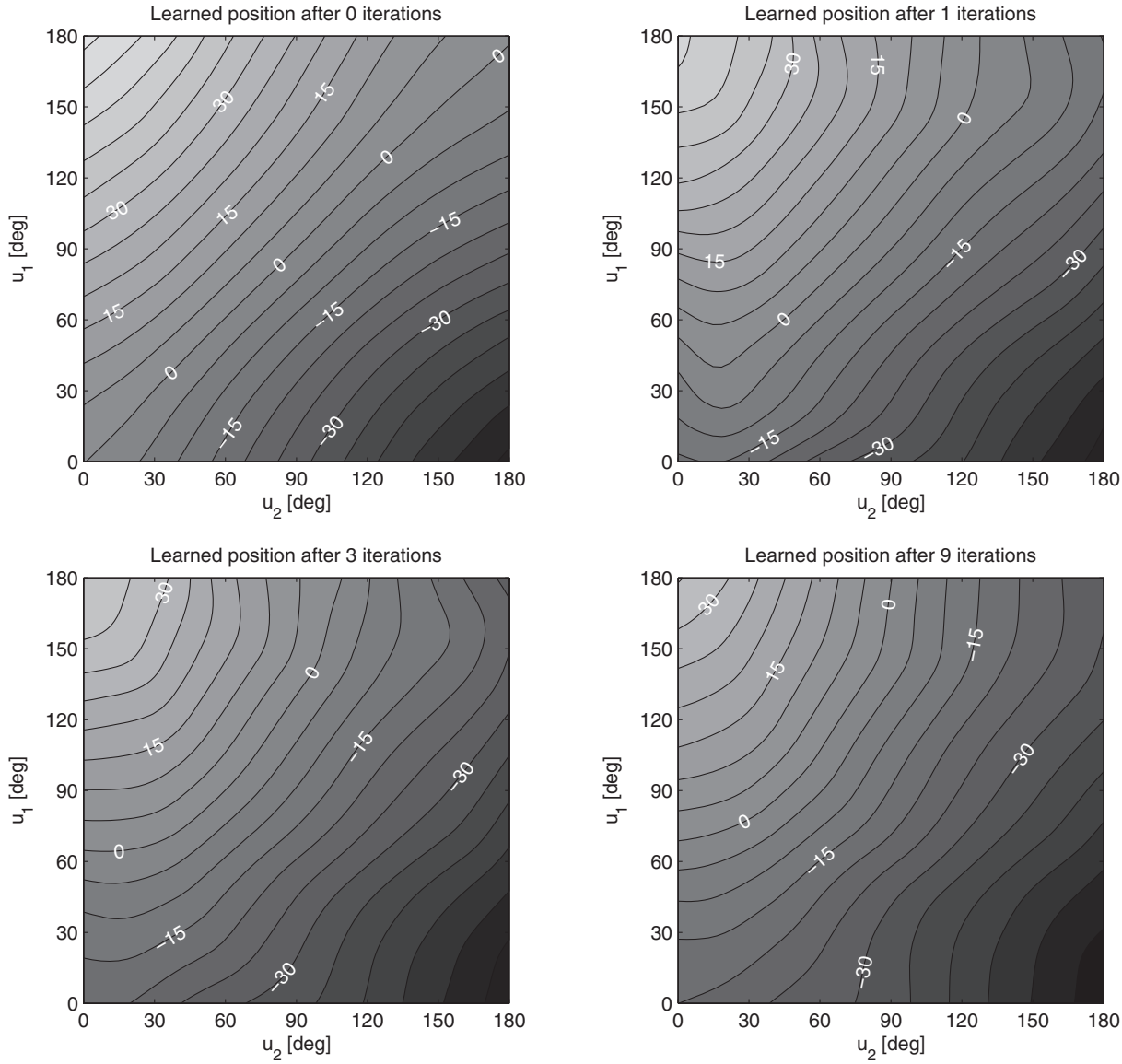

Fig. 7. Learned position models during the adaptation process. The white numbers represent the equilibrium points.

As before, we observe the benefits of using stochastic information for optimization compared to a deterministic optimization (not using the LWPR confidence bounds). After computing the optimal control using ILQG we ran the optimal feedback control law (see Appendix B) consecutively 20 times in each condition, i.e. with and without stochastic optimization. Note that the perturbation motor is switched on at all times. Figure 13 summarizes the results: as expected the stochastic information in the cost function induces a co-activation for the reaching task, which shows generally better performance in terms of reduced variability of the trajectories. Evaluating the movement variability where the accuracy weight is maximal, i.e. for $t > 80$, the standard deviation of the trajectories is significantly lower with $\sigma_{\text{stoch}} = 0.55^\circ$ for the stochastic optimization compared to the deterministic optimization with $\sigma_{\text{det}} = 1.38^\circ$. A detailed look at the bottom right plot in Figure 13 reveals a minor shift in the recorded trajectory compared to the planned one from the analytic model. We attribute this error to imprecisions in the hardware, i.e. tiny asymmetries, which are not included in the analytic model. In the case of higher co-contraction, small manufacturing errors and an increased joint friction lead to deviations towards the idealized analytic model predictions. Indeed the learned dynamic model can account for these asymmetries as can be seen in Figure 9 (left), along the equilibrium position $\theta = 0^\circ$, i.e. the line $u_1 = u_2$ is slightly skewed.

5. Conclusion and Outlook

In this paper we have presented a stochastic optimal control model for antagonistically actuated systems. We proposed

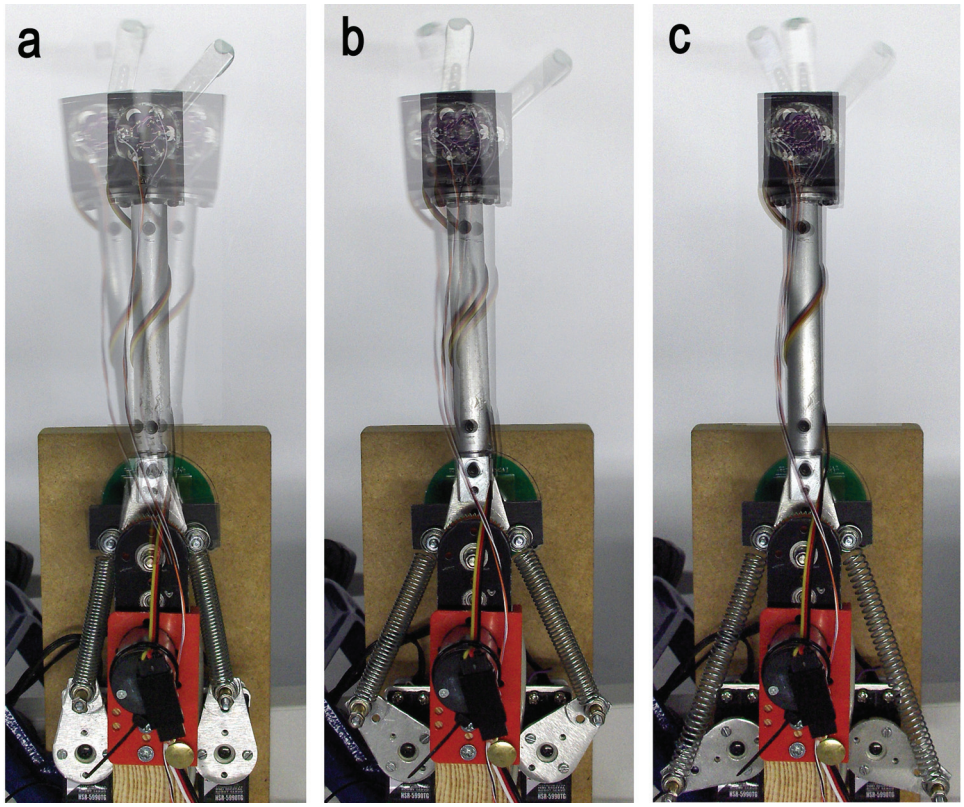

Fig. 8. Motion traces of our SEA hardware around $\theta = 0^\circ$. The perturbation motor causes different deflections depending on the co-contraction levels: (a) $u_1 = u_2 = 0^\circ$, (b) $u_1 = u_2 = 45^\circ$, (c) $u_1 = u_2 = 120^\circ$.

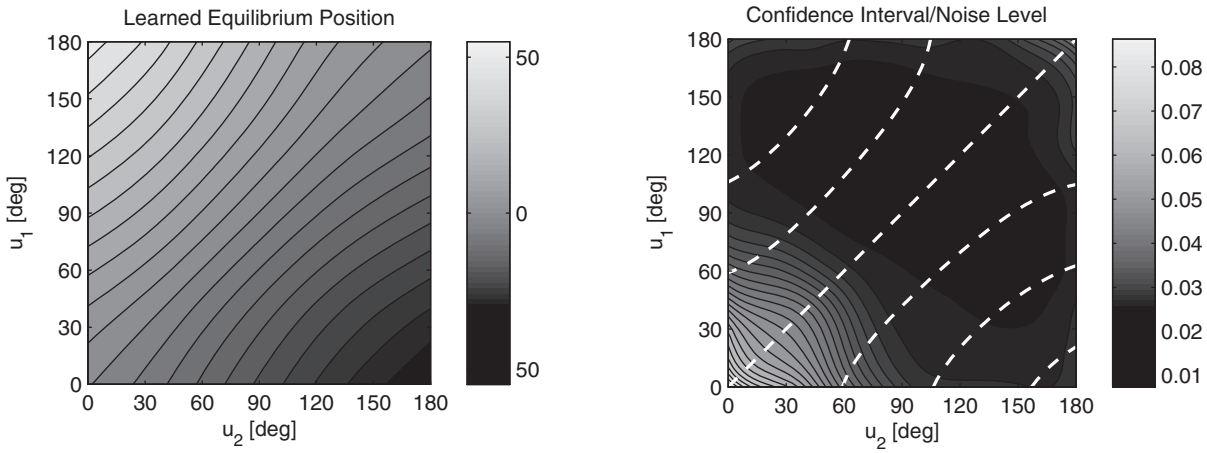

Fig. 9. Left: Learned equilibrium position as a function of the motor positions (in degrees), with contour lines spaced at 5° intervals. Right: Stochastic information given by the heteroscedastic confidence intervals of LWPR.

to learn the *dynamics* as well as the *stochastic information* of the controlled system from sensorimotor feedback of the plant. This control architecture can account for a systematic change in the system properties (Experiment 1) and, furthermore, is able, by incorporating the heteroscedastic prediction variances into the optimization, to compensate for stochastic perturbations that were induced in the plant. Doing so, our control model demonstrated significantly better accuracy performance than the deterministic optimization in both energy-optimal equilibrium point control (Experiment 2) and energy-optimal reaching using dynamic optimization (Experiment 3). The improved behavior was achieved by co-activating antagonistic motors, i.e. by using the redundant degree of freedom in the system based on the

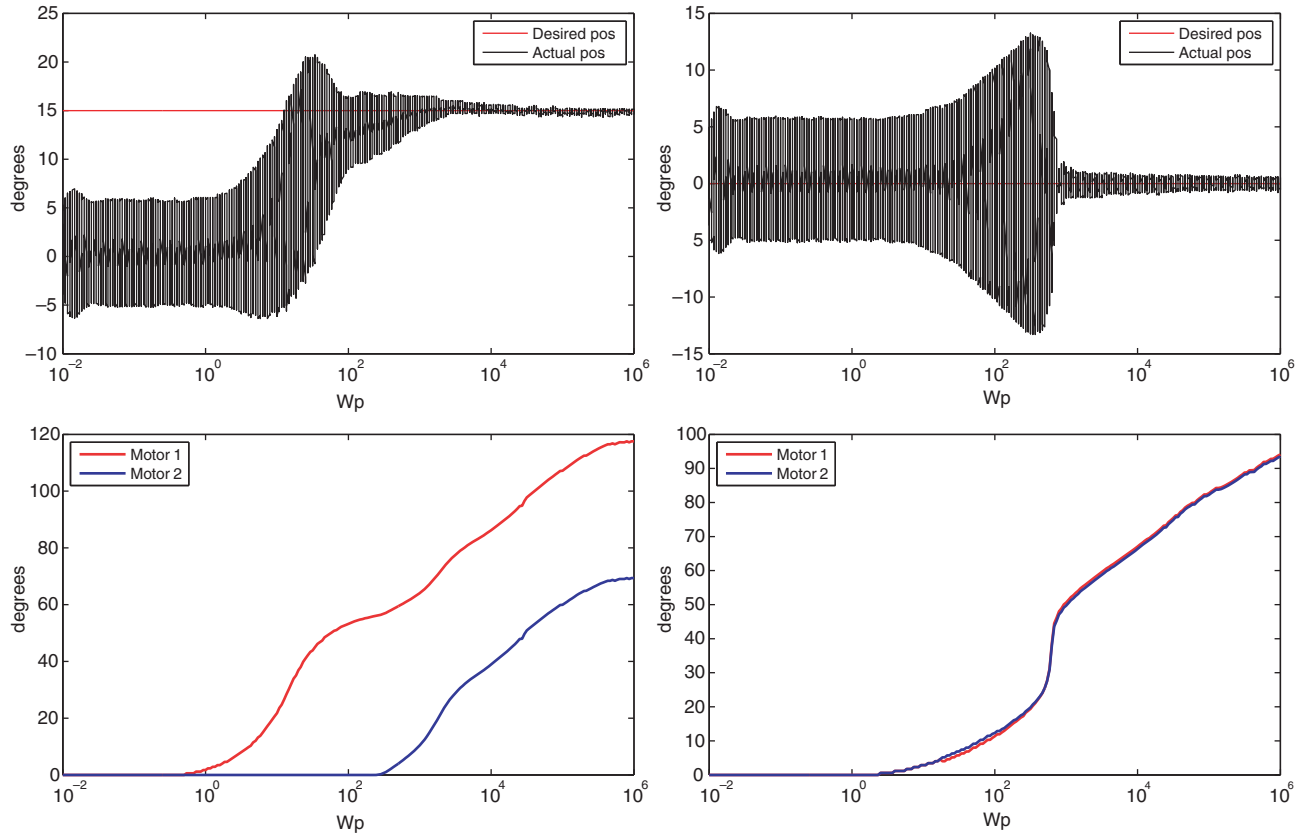

Fig. 10. Experiment with increasing position penalty w_p for two targets. Left plot column: $\hat{\theta} = 15°$; right plot column: $\hat{\theta} = 0°$. The plots show the desired vs. measured position and corresponding motor commands as a function of the accuracy requirement (pre-factor w_p). The shaded area results from the perturbation motor.

first principles of optimality. The presented results demonstrate that this is a viable optimal control strategy for real hardware systems that exhibit hard-to-model system properties (e.g. asymmetries, systematic changes) as well as stochastic characteristics (e.g. using a power tool) that may be unknown *a priori*.

An advantage of the presented control architecture is that motor co-activation (or impedance) does not need to be specified explicitly as a control variable but emerges from the actual learned stochasticity within the system (scaled with the specified accuracy requirements of the task). Therefore, co-activation (i.e. higher impedance), since it is energetically expensive, will only be applied if it actually is beneficial for the accuracy of the task.

Exploiting stochasticity in wider domains The methodology we suggest for optimal exploitation of sensorimotor stochasticity through learning is a generic principle that goes beyond applications to impedance modulation of antagonistic systems but can be generalized to deal with any kind of control or state dependent uncertainties. For example, if we wish to control a robot arm that suffers from poor repeatability in certain joint angles or in a particular range of velocities, this would be visible in the noise landscape (given one has learned state dependent stochastic dynamics) and consequently those regions would be "avoided" by the optimal controller. In this context, the source of the stochasticity is irrelevant for the learner and therefore, it could arise from internal (i.e. noise in the motor), as well as external (i.e. power tool) sources. However, the stochastic system properties must, to a certain degree, be *stationary in time* so that the learner can acquire enough information about the noise landscape.

Biological relevance As mentioned in the introduction, biological systems are often used as a benchmark for the control of artificial systems. In this work not only the antagonistic hardware but also the actual control architecture is motivated by biological principles. Optimality approaches have been a very fruitful line of research (Todorov 2004; Scott 2004; Shadmehr and Krakauer 2008) and its combination with a learning paradigm (Mitrovic et al. 2008) is biologically justified *a priori*, since the sensorimotor system can be seen as the product of an optimization process (i.e. evolution, development, learning, adaptation) that constantly learns to improve its behavioral performance (Li 2006). Indeed, internal models play a key role in efficient human motor control (Davidson and Wolpert 2005) and it

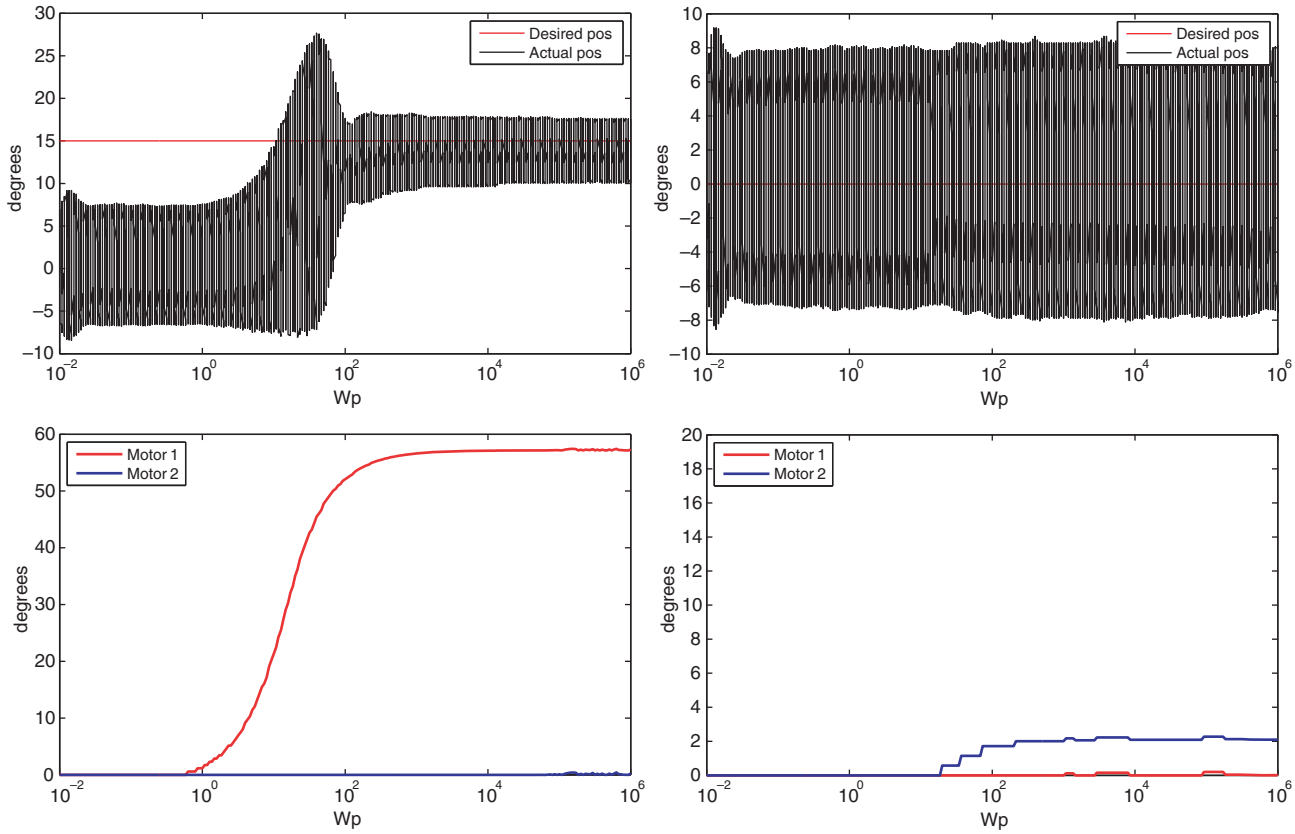

Fig. 11. The same experiment as in Figure 10 where the stochastic information was not incorporated into the optimization.

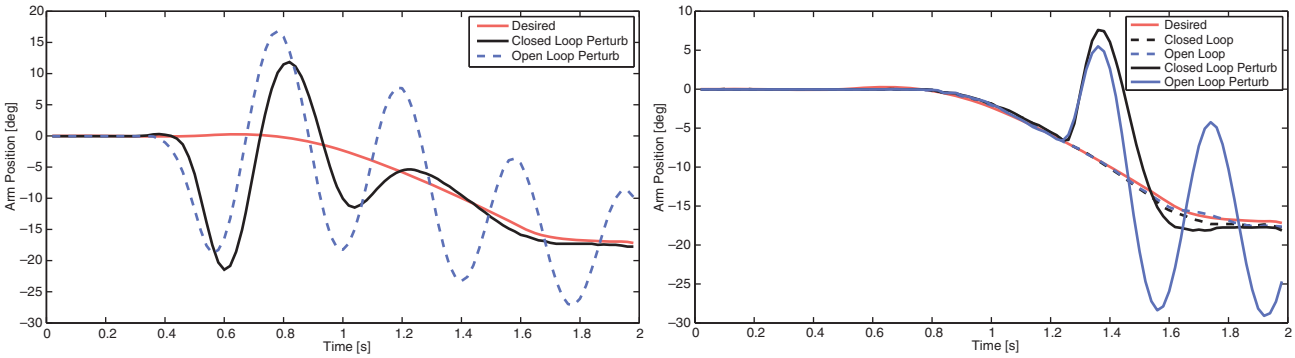

Fig. 12. ILQG-reaching task without stochastic information used in open-loop and closed-loop control. We perturbed the arm by hitting it once after 0.4 s (left plot) and 1.2 s (right plot), respectively. The dashed lines in the right plot represent unperturbed trials (open loop and closed loop).

has been suggested that the motor system forms an internal forward dynamic model to compensate for delays, uncertainty of sensory feedback, and environmental changes in a predictive fashion (Wolpert et al. 1995; Kawato 1999; Shadmehr and Wise 2005). Notably a *learned* optimal trade-off between energy consumption, accuracy and impedance has been repeatedly observed in human impedance control studies (Burdet et al. 2001; Franklin et al. 2008). More specifically, the amount of impedance modulation in humans seems to be governed by some measure of uncertainty, which could arise from internal (e.g. motor noise) or external (e.g. tools) sources (Selen et al. 2009).

In the computational model presented here, these uncertainties are represented by the heteroscedastic confidence bounds of LWPR and integrated into the optimization process via the performance index (i.e. cost function). Such an assumption is biologically plausible, since humans have the ability to learn not only the dynamics but also the

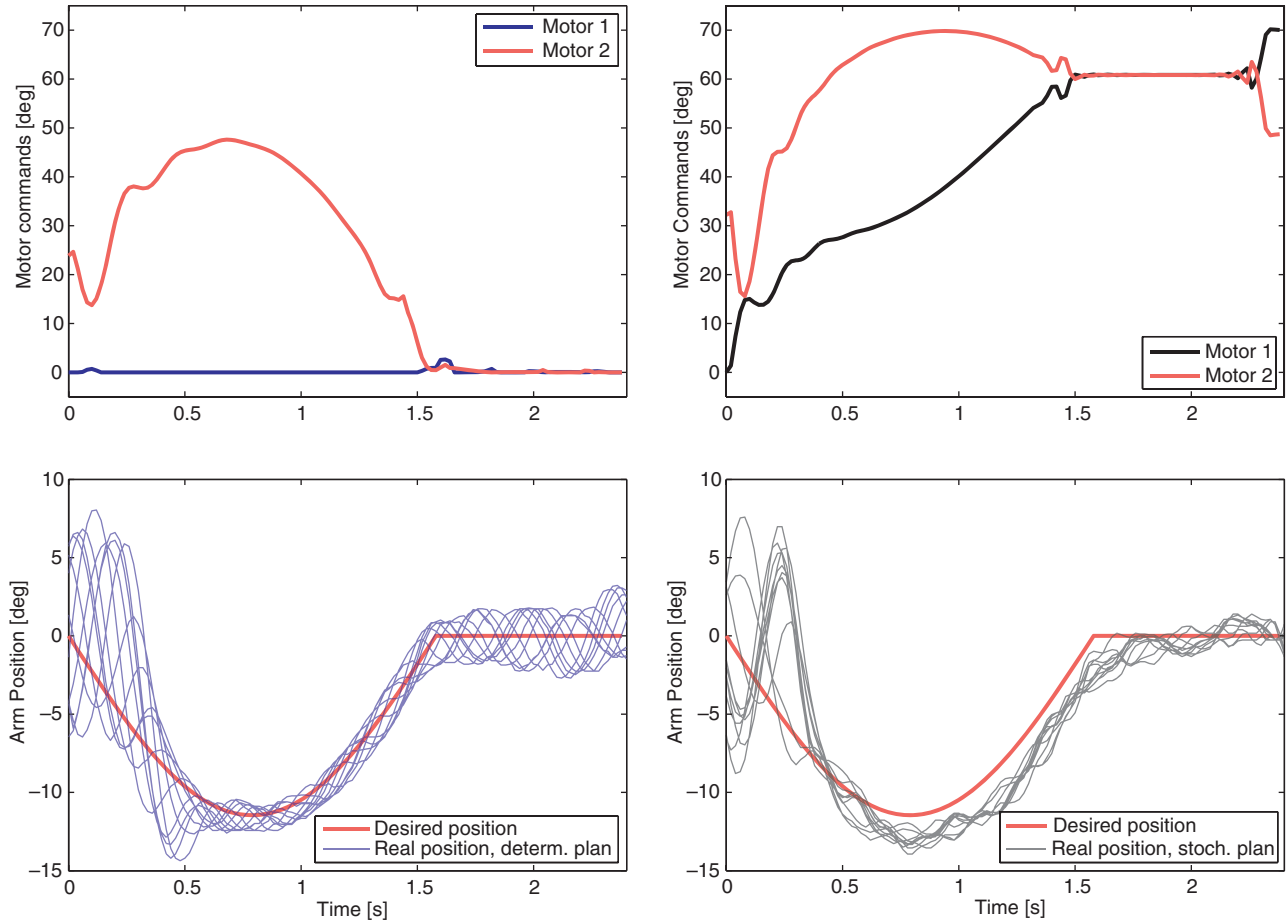

Fig. 13. Twenty trials of ILQG for a tracking task of 2.5 s. Left column: Deterministic optimization does not exhibit co-contraction. Right column: Twenty trials of ILQG using stochastic information in the cost function. The system co-contracts as the accuracy requirements increase.

stochastic characteristics of tasks, in order to optimally learn the control of a complex task (Chhabra and Jacobs 2006; Selen et al. 2009).

Hardware Limitations and Scalability This work represents an initial attempt to modulate the impedance of a real antagonistic system in a principled fashion. The proposed SEA has been primarily designed to perform as a "proof of concept" of our control method on a real system. Specifically we can identify several limitations of our system that need further investigation in the future.

First, the stiffness range of the system is fairly low as spring non-linearities are achieved by the geometric effect of changing the moment arms. There are other, mechanically sophisticated, SEA designs with large stiffness ranges (e.g. Grebenstein and van der Smagt 2008; van Ham et al. 2009), which also could serve as attractive implementation platforms for our algorithm. Specifically the MACCEPA design (van Ham et al. 2007) is very appealing as it is technically simple and offers a large stiffness range; however, parallels to biologically realistic implementations are less obvious in this design, as the system is not antagonistically actuated. The fact that we were able to obtain a significant increase in co-contraction from the learned stochastic information, even for hardware with a very low stiffness range is promising, indicating good resolution capabilities of the localized variance measure in LWPR.

Second, the relatively slow control loop (50 Hz) causes controllability issues (i.e. slow feedback) and, furthermore, turned out to be sensitive to numerical integration errors within ILQG. While these numerical issues have not caused problems in an analytic dynamic formulation (Experiment 3), they turned out to be critical when we run ILQG using the full learned forward dynamics $\tilde{f}(\mathbf{x}, \mathbf{u})$. Under these conditions, for most of the time ILQG does not converge to a reasonable solution. A potential route of improvement could be a combination of LWPR learning with an analytic model. Instead of "ignoring" valuable knowledge about the system given in analytic form, one could focus on learning an *error model* only, i.e. aspects of the dynamics that are not described by the analytic model.

Third, the transfer of optimal controls from simulation to the real hardware has proven to be very challenging. Currently we are computing ILQG solutions for a fixed time

horizon and later applying them to the SEA. For slower movements this approach produces satisfying accuracy. In Experiment 3 we "enforced" slower and smoother movements by formulating an appropriate time-dependent cost function. However, for movements with higher frequency the situation is more difficult: Errors accumulate on the hardware over the course of the trajectory, since the feedback loop for corrections is very slow. This leads to solutions that differ significantly from the preplanned optimal solution. A potential route to resolve this problem is to use a model predictive control approach in which the optimal solutions are re-computed during control with current states of the plant as initial states. However, this approach requires computationally efficient re-computations of the optimal control law, which may be hard to obtain, especially for systems with higher dimensionality.

Finally, our experiments were carried out on a low-dimensional system with a single joint and two motors. Implementations on systems with higher dimensionality, however, are still very challenging as the construction of antagonistic robots is non-trivial and the availability of large degrees of freedom systems is very limited. Due to the curse of dimensionality, high-dimensional systems impose serious computational challenges on both optimal control methods and machine learning techniques. While some of these issues have been addressed in previous work (Todorov et al. 2005; Mitrovic et al. 2008, 2010), we believe that the study of impedance control based on stochastic sensorimotor feedback is a promising route of research for both robotic and biological systems.

Funding

This work was partially supported by the EU FP7 Project STIFF and by a Royal Academy of Engineering and Microsoft Research Fellowship to SV.

Notes

1. This means we put expectation brackets around the integrals and $h(\cdot)$ in (7).
2. For the infinite-horizon case, the matrix is constant.
3. Note that we have ignored any motor dynamics.
4. Stability here refers to the desired equilibrium position.
5. Heteroscedastic noise has different variances across the state and action space. For example, the variance of the noise can scale with the magnitude of the control signal **u**, which is also called signal dependent noise.
6. Alternatively, assume the feedback loop is so slow that it is practically unusable.
7. For our SEA this optimization can be performed in real time, i.e. at least 50 times per second, which corresponds to the maximum control frequency of our system (50 Hz).
8. or any other high-frequency perturbation.

References

Bellman R (1957) *Dynamic Programming*. Princeton, NJ: Princeton University Press.

Bertsekas DP (1995) *Dynamic programming and optimal control*. Belmont, MA: Athena Scientific.

Bobrow J, Dubowsky S and Gibson J (1985) Time-optimal control of robotic manipulators along specified paths. *The International Journal of Robotics Research* 4(3): 3–17.

Burdet E, Osu R, Franklin DW, Milner TE and Kawato M (2001) The central nervous system stabilizes unstable dynamics by learning optimal impedance. *Nature* 414: 446–449.

Chhabra M and Jacobs RA (2006) Near-optimal human adaptive control across different noise environments. *Journal of Neuroscience* 26: 10883–10887.

Collins S and Ruina A (2005) A bipedal walking robot with efficient human-like gait. In *Proceedings of the IEEE International Conference on Robotics and automation (ICRA)*, pp. 1983–1988.

Collins SH and Kuo AD (2010) Recycling energy to restore impaired ankle function during human walking. *PLoS ONE* 5 pages-e9307.

Cortes J, Martinez S, Ostrowski J and McIsaac KA (2001) Optimal gaits for dynamic robotic locomotion. *The International Journal of Robotics Research* 20: 707–728.

Davidson PR and Wolpert DM (2005) Widespread access to predictive models in the motor system: a short review. *Journal of Neural Engineering* 2: 313–319.

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

Faisal AA, Selen LPJ and Wolpert DM (2008) Noise in the nervous system. *Nature Reviews Neuroscience* 9: 292–303.

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

Franklin D, Burdet E, Tee KP, Osu R, Chew CM, Milner TE, et al. (2008) CNS learns stable, accurate, and efficient movements using a simple algorithm. *The Journal of Neuroscience* 28: 11165–11173.

Grebenstein M and van der Smagt P (2008) Antagonism for a highly anthropomorphic hand–arm system. *Advanced Robotics* 22: 39–55.

Hogan N (1984) Adaptive control of mechanical impedance by coactivation of antagonist muscles. *IEEE Transactions on Automatic Control* 29: 681–690.

Hurst JW, Chestnutt J and Rizzi A (2004) *An Actuator with Mechanically Adjustable Series Compliance*. Technical Report CMU-RI-TR-04-24, Robotics Institute, Carnegie Mellon University.

Jacobson DH and Mayne DQ (1970) *Differential Dynamic Programming*. New York: Elsevier.

Kawato M (1999) Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology* 9: 718–727.

Laffranchi M, Tsagarakis NG and Caldwell DG (2010) A variable physical damping actuator (VPDA) for compliant robotic joints. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

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

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: 1439–1453.

Migliore SA, Brown EA and DeWeerth SP (2005) Biologically inspired joint stiffness control. In *Proceedings of the 2005*

IEEE International Conference on Robotics and Automation (ICRA), pp. 4508–4513.

Mitrovic D, Klanke S and Vijayakumar S (2008) Adaptive optimal control for redundantly actuated arms. In *Proceedings of the 10th International Conference on Simulation of Adaptive Behaviour (SAB)*, Osaka, Japan.

Mitrovic D, Klanke S and Vijayakumar S (2010) Optimal feedback control for anthropomorphic manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Nakamura Y and Hanafusa H (1987) Optimal redundancy control of robot manipulators. *The International Journal of Robotics Research* 6: 32–42.

Paluska D and Herr H (2006) The effect of series elasticity on actuator power and work output: Implications for robotic and prosthetic joint design. *Robotics and Autonomous Systems* 54: 667–673.

Pratt GA and Williamson MM (1995) Series elastic actuators. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 399–406.

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

Selen LP, Franklin DW and Wolpert DM (2009) Impedance control reduces instability that arises from motor noise. *Journal of Neuroscience* 40(9): 12606–12616.

Shadmehr R and Krakauer JW (2008) A computational neuroanatomy for motor control. *Experimental Brain Research* 185: 359–381.

Shadmehr R and Wise S (2005) *The Computational Neurobiology of Reaching and Pointing*. Cambridge, MA: The MIT Press.

Stengel RF (1994) *Optimal control and estimation*. New York: Dover Publications.

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

Todorov E (2006) Optimal Control Theory. In Doya K (Ed.), *Bayesian Brain: Probabilistic Approaches to Neural Coding*. Cambridge, MA: MIT Press, pp. 269–298.

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

Todorov E, Li W and Pan X (2005) From task parameters to motor synergies: A hierarchical framework for approximately-optimal control of redundant manipulators. *Journal of Robotic Systems* 22: 691–710.

Tonietti G, Schiavi R and Bicchi A (2005) Design and control of a variable stiffness actuator for safe and fast physical human/robot interaction. *IEEE International Conference Robotics and Automation (ICRA)* pp. 526–531.

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.

van Ham R, Sugar T, Vanderborght B, Hollander K and Lefeber D (2009) Compliant actuator designs. *IEEE Robotics and Automation Magazine* 16(3): 81–94.

van Ham R, Vanderborght B, Van Damme M, Verrelst B and Lefeber D (2007) MACCEPA, the mechanically adjustable compliance and controllable equilibrium position actuator: Design and implementation in a biped robot. *Robotics and Autonomous Systems* 55: 761–768.

Vanderborght B, Van Ham R, Lefeber D, Sugar TG and Hollander KW (2009) Comparison of mechanical design and energy consumption of adaptable, passive–compliant actuators. *The International Journal of Robotics Research* 28: 90–103.

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

Wolf S and Hirzinger G (2008) A new variable stiffness design: Matching requirements of the next robot generation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Wolpert DM, Ghahramani Z and Jordan MI (1995) An internal model for sensorimotor integration. *Science* 269: 1880–1882.

Zinn M, Khatib O, Roth B and Salisbury JK (2004) Playing it safe. *IEEE Robotics and Automation Magazine* 11: 12–21.

Appendix A: Learning with LWPR

In order to learn the plant dynamics various supervised learning algorithms could be applied. Here we focus on *local* learning methods, which represent a function by using small simplistic patches, e.g. first-order polynomials. The size of the locality is determined by gating activation kernels, and the positions and number of the local kernels are adapted during learning to represent the non-linear function. Because the input data activates only local patches, local learning algorithms are robust against global negative interference. This ensures the flexibility of the learned model towards systematic changes in the dynamic properties of the arm (e.g. load, material wear). Furthermore, the domain of real-time robot control demands certain properties of a learning algorithm, namely fast learning rates and high prediction speeds at run-time if the model is trained incrementally. LWPR has been shown to exhibit these properties, and to be very efficient for incremental learning of non-linear models (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 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 modeled by Gaussian kernels, so their activation or response to a query vector $\mathbf{z}$ (here the inputs are the two motor commands $\mathbf{u}$) is given by

$$w_k(\mathbf{z}) = \exp\left(-\frac{1}{2}(\mathbf{z}-\mathbf{c}_k)^T \mathbf{D}_k(\mathbf{z}-\mathbf{c}_k)\right), \qquad (22)$$

where $\mathbf{c}_k$ is the center of the kth 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

$$\tilde{f}(\mathbf{z}) = \frac{1}{W}\sum_{k=1}^{K} w_k(\mathbf{z})\,\psi_k(\mathbf{z}), \quad W = \sum_{k=1}^{K} w_k(\mathbf{z}), \qquad (23)$$

$$\psi_k(\mathbf{z}) = b_k^0 + \mathbf{b}_k^{\mathrm{T}}(\mathbf{z} - \mathbf{c}_k), \qquad (24)$$

where b_k^0 and $\mathbf{b}_k$ denote the offset and slope of the kth 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. Furthermore, the statistical parameters of LWPR regression models provide access to the confidence intervals, here termed *confidence bounds*, of new prediction inputs (Vijayakumar et al. 2005). In LWPR the predictive variances are assumed to evolve as an additive combination of the variances within a local model and the variances independent of the local model. The predictive variance estimates $\sigma^2_{\mathrm{pred},k}$ for the kth local model can be computed by analogy with ordinary linear regression. Similarly one can formulate the global variances σ^2 across models. By analogy with Equation (23), LWPR then combines both variances additively to form the confidence bounds given by

$$\sigma^2_{\mathrm{pred}} = \frac{1}{W^2}\left(\sum_{k=1}^{K} w_k(\mathbf{z})\,\sigma^2 + \sum_{k=1}^{K} w_k(\mathbf{z})\,\sigma^2_{\mathrm{pred},k}\right). \qquad (25)$$

The local nature of LWPR leads to the intuitive requirement that only receptive fields that actively contribute to the prediction (e.g. large linear regions) are involved in the actual confidence bounds calculation. Large confidence bound values typically evolve if the training data contains much noise and other sources of variability, such as changing output distributions. Further regions with sparse or no training data, i.e. unexplored regions, show large confidence bounds compared with densely trained regions. Figure 14 depicts the learning concepts of LWPR graphically for a learned model with one input and one output dimension. The noisy training data was drawn from an example function that becomes more linear and more noisy for larger z values. Furthermore, in the range $z = [5..6]$ no data was sampled for training to show the effects of sparse data on LWPR learning.

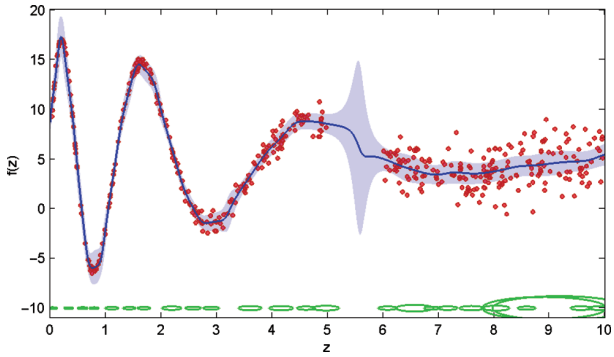

Fig. 14. Typical regression function (blue continuous line) using LWPR. The dots indicate a representative training data set. The receptive fields are shown as ellipses drawn at the bottom of the plot. The shaded region represents the confidence bounds around the prediction function. The confidence bounds grow between $\mathbf{z} = [5..6]$ (no training data) and generally towards larger $\mathbf{z}$ values (noise grows with larger values).

Appendix B: The ILQG Algorithm

The ILQG algorithm starts with a time-discretized initial guess of an optimal control sequence and then iteratively improves it with respect to the cost function. From the initial control sequence $\bar{\mathbf{u}}^i$ at the ith iteration, the corresponding state sequence $\bar{\mathbf{x}}^i$ is retrieved using the deterministic forward dynamics $\mathbf{f}$ with a standard Euler integration $\bar{\mathbf{x}}^i[k+1] = \bar{\mathbf{x}}^i[k] + \Delta t\,\mathbf{f}(\bar{\mathbf{x}}^i[k], \bar{\mathbf{u}}^i[k])$. Next, the discretized dynamics (Equation (5)) are linearly approximated as

$$\delta\mathbf{x}[k+1] = \left(\mathbf{I} + \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\bigg|_{\bar{\mathbf{x}}[k]}\right)\delta\mathbf{x}[k] + \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\bigg|_{\bar{\mathbf{u}}[k]}\delta\mathbf{u}[k]. \qquad (26)$$

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

$$\begin{aligned}\mathrm{cost}[k] = {} & q[k] + \delta\mathbf{x}[k]^{\mathrm{T}}\mathbf{q}[k] + \frac{1}{2}\delta\mathbf{x}[k]^{\mathrm{T}}\mathbf{Q}[k]\delta\mathbf{x}[k] + \\ & \delta\mathbf{u}[k]^{\mathrm{T}}\mathbf{r}[k] + \frac{1}{2}\delta\mathbf{u}[k]^{\mathrm{T}}\mathbf{R}[k]\delta\mathbf{u}[k] + \\ & \delta\mathbf{u}[k]^{\mathrm{T}}\mathbf{P}[k]\delta\mathbf{x}[k]\end{aligned} \qquad (27)$$

where

$$\begin{aligned} q[k] &= \Delta t v[k] & \mathbf{q}[k] &= \Delta t \frac{\partial v[k]}{\partial \mathbf{x}}\bigg|_{\bar{\mathbf{x}}[k]} \\ \mathbf{Q}[k] &= \Delta t \frac{\partial^2 v[k]}{\partial \mathbf{x}\,\partial \mathbf{x}}\bigg|_{\bar{\mathbf{x}}[k],\bar{\mathbf{u}}[k]} & \mathbf{P}[k] &= \Delta t \frac{\partial^2 v[k]}{\partial \mathbf{u}\,\partial \mathbf{x}}\bigg|_{\bar{\mathbf{x}}[k],\bar{\mathbf{u}}[k]} \\ \mathbf{r}[k] &= \Delta t \frac{\partial v[k]}{\partial \mathbf{u}}\bigg|_{\bar{\mathbf{u}}[k]} & \mathbf{R}[k] &= \Delta t \frac{\partial^2 v[k]}{\partial \mathbf{u}\,\partial \mathbf{u}}\bigg|_{\bar{\mathbf{u}}[k]}. \end{aligned} \qquad (28)$$

Both approximations are formulated as deviations $\delta\mathbf{x}^i[k] = \mathbf{x}^i[k] - \bar{\mathbf{x}}^i[k]$ and $\delta\mathbf{u}^i[k] = \mathbf{u}^i[k] - \bar{\mathbf{u}}^i[k]$ 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 optimization 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 $\bar{\mathbf{u}}^{i+1}[k] = \bar{\mathbf{u}}^i[k] + \delta\bar{\mathbf{u}}^i[k]$. Finally, $\bar{\mathbf{u}}^{i+1}[k]$ is applied to the system dynamics (Equation (5)) 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 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 of the plant (Figure 15). The control law for each time step k is defined as

$$\mathbf{u}[k]^{\text{plant}} = \bar{\mathbf{u}}[k] + \delta\mathbf{u}[k] \tag{29}$$

$$\delta\mathbf{u}[k] = \mathbf{L}[k] \cdot (\mathbf{x}[k] - \bar{\mathbf{x}}[k]), \tag{30}$$

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

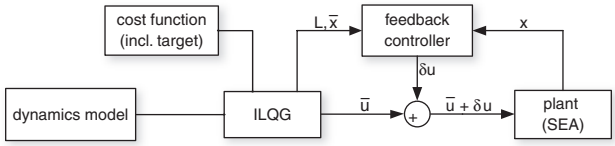

Fig. 15. The optimal feedback control scheme using ILQG.