International Journal of Control
Vol. 80, No. 9, September 2007, 1439–1453

Taylor & Francis
Taylor & Francis Group

Iterative linearization methods for approximately optimal control and estimation of non-linear stochastic system

W. LI*† and E. TODOROV‡

†Department of Mechanical and Aerospace Engineering,
University of California San Diego, La Jolla, CA 92093-0411
‡Department of Cognitive Science, University of California San Diego,
La Jolla, CA 92093-0515

(*Received 28 November 2005; in final form 25 March 2007*)

This paper presents an iterative Linear-Quadratic-Gaussian method for locally-optimal control and estimation of non-linear stochastic systems. The new method constructs an affine feedback control law, obtained by minimizing a novel quadratic approximation to the optimal cost-to-go function, and a non-adaptive estimator optimized with respect to the current control law. The control law and filter are iteratively improved until convergence. The performance of the algorithm is illustrated on a complex biomechanical control problem involving a stochastic model of the human arm.

1. Introduction

Optimal control theory has received a great deal of attention since the late 1950s, and has found applications in many fields of science and engineering (Bryson and Ho 1969, Bitmead 1993, 2000, Bertsekas 2000, Kushner and Dupuis 2001). It has also provided the most fruitful general framework for constructing models of biological movement (Uno *et al*. 1989, Harris and Wolpert 1998, Todorov and Jordan 2002, Todorov 2004). In the field of motor control, optimality principles not only yield accurate descriptions of observed phenomena, but are well justified *a priori*. This is because the sensorimotor system is the product of optimization processes (i.e., evolution, development, learning, adaptation) which continuously act to improve behavioural performance. The majority of existing optimality models in motor control have been formulated in open-loop. However, the most remarkable property of biological movements (in comparison with synthetic ones) is that they can accomplish complex high-level goals in the presence of large internal fluctuations, noise, delays, and unpredictable changes in the environment. This is only possible through an elaborate feedback control scheme. Indeed, focus has recently shifted towards stochastic optimal feedback control models. This approach has already clarified a number of long-standing issues related to the control of redundant biomechanical systems (Todorov 2005).

However, solving complex optimal control of partially-observable stochastic systems (Phillis 1985, 1989) is generally intractable, because the optimal estimator tends to be infinite-dimensional and consequently the optimal controller is also infinite-dimensional. The only notable exception is the Linear-Quadratic-Gaussian (LQG) setting (Moore *et al*. 1999), where the posterior probability over the system state is Gaussian and the optimal controller only depends on the mean of that Gaussian. Unfortunately many real-world problems (including the biological control problems we are interested in) are not LQG and are not amenable to global LQG approximations. Local LQG approximations, on the other hand, are often quite reasonable: they rely on local low-order polynomial approximations to the system dynamics, cost function and noise log-probability—all of which tend to be smooth functions. It then makes sense to construct approximately-optimal estimators and controllers by starting with a local LQG approximation, finding the optimal solution, using it to construct a new LQG approximation, and iterating until convergence.

*Corresponding author. Email: wwli@mechanics.ucsd.edu

International Journal of Control
ISSN 0020–7179 print/ISSN 1366–5820 online © 2007 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/00207170701364913

This work represents the convergence of two lines of research we have previously pursued. In one line of research, we derived an iterative algorithm for optimal estimation and control of partially-observable linear-quadratic systems subject to state-dependent and control-dependent noise (Todorov 2005). This was possible due to a restriction to non-adaptive filters. In another line of research, we derived an iterative algorithm for optimal control of fully-observable stochastic non-linear systems with arbitrary costs and control constraints (Todorov and Li 2003, Li and Todorov 2004). This was possible due to a novel approximation to the optimal cost-to-go function. Here we combine these ideas, and derive an algorithm which can handle partially-observable non-linear systems, non-quadratic costs, state-dependent and control-dependent noise, and control constraints. Before deriving our iterative Linear-Quadratic-Gaussian (ILQG) method, we give a more detailed overview of what is new here.

(1) Most dynamic programming methods use quadratic approximations to the optimal cost-to-go function. All such methods are "blind" to additive noise. However, in many problems of interest the noise is control-dependent (Chow and Birkemeier 1990, De Oliveria and Skelton 2001, Gershon *et al*. 2001, Jimenez and Ozaki 2002), and such noise can easily be captured by quadratic approximations as we show below. Our new ILQG method incorporates control-dependent and state-dependent noise —which turns out to have an effect similar to an energy cost. In practical situations, the state of the plant is only available through noisy measurement. When the state of the plant is fully observable, optimal LQG-like solutions can be computed efficiently as shown by McLane (1971), Willems and Willems (1976), El Ghaoui (1995), Beghi and D'Alessandro (1998) and Rami *et al*. (2001). Such methodology has also been used to model reaching movements (Hoff 1992). Most relevant to the study of sensorimotor control, however, is the partially-observable case. Our goal here is to address that problem.

(2) Quadratic approximation methods are presently restricted to unconstrained problems (Ng *et al*. 2002). Generally speaking, constraints make the optimal cost-to-go function non-quadratic (Abu-Khalaf and Lewis 2005), but since we are approximating that function anyway, we might as well take into account the effects of control constraints to the extent possible. Our new ILQG method does that—by modifying the feedback gain matrix whenever an element of the open-loop control sequence lies on the constraint boundary.

(3) Quadratic approximation methods are based on Riccati equations: define a quadratic optimization problem that the optimal controls satisfy at time step t, solve it analytically, and obtain a formula for the optimal cost-to-go function at time step $t-1$. Optimizing a quadratic is only possible when the Hessian is positive-definite. This is of course true in the classic LQG setting, but when LQG methods are used to approximate general non-linear dynamics (Isidori 1995) with non-quadratic costs, the Hessian can (and in practice does) have zero and even negative eigenvalues. The traditional remedy is to "fix" the Hessian, using a Levenberg–Marquardt method, or an adaptive shift scheme, or simply replace it with the identity matrix (which yields the steepest descent method). The problem is that after fixing the Hessian, the optimization at time step t is no longer performed exactly—contrary to what the Riccati equations assume. Instead of making this invalid assumption, our new method takes the fixed Hessian into account, and constructs a cost-to-go approximation consistent with the resulting control law. This is done by modified Riccati-like equations.

While the new algorithm should be applicable to a range of problems, our specific motivation for developing it is the modelling of biological movement. Such modelling has proven extremely useful in the study of how the brain controls movement (Todorov and Jordan 2002). In § 2 we formalize the original optimal problem we want to solve. In § 3 we present a LQG approximation to our original optimal control problem and compute an approximately-optimal control law under the consideration that state estimates are obtained by an unbiased non-adaptive linear filter. In § 4 we compute the optimal feedback control law for any non-adaptive linear filter, and show that it is linear in the state estimate. In § 5 we derive the optimal filter corresponding to the given control law. Section 6 illustrates the application of our method to the analysis of reaching movements and explores numerically the convergence properties of the algorithm, in the context of reaching movements using a model of the human arm.

2. Problem formulation

Consider the non-linear dynamical system described by the stochastic differential equations

$$\mathrm{d}\mathbf{x}^p(t) = f(\mathbf{x}^p, \mathbf{u}^p)\,\mathrm{d}t + \mathcal{F}(\mathbf{x}^p, \mathbf{u}^p)\,\mathrm{d}\omega(t), \tag{1}$$

along with the output equation

$$\mathrm{d}\mathbf{y}^p(t) = g(\mathbf{x}^p, \mathbf{u}^p)\,\mathrm{d}t + \mathcal{G}(\mathbf{x}^p, \mathbf{u}^p)\,\mathrm{d}v(t), \tag{2}$$

where state variable $\mathbf{x}^p \in \mathbb{R}^{n_x}$, control input $\mathbf{u}^p \in \mathbb{R}^{n_u}$, measurement output $\mathbf{y}^p \in \mathbb{R}^{n_y}$, and standard Brownian motion noise $\omega \in \mathbb{R}^{n_\omega}$, $v \in \mathbb{R}^{n_v}$ are independent of each other.

Let $\ell(t, \mathbf{x^p}, \mathbf{u^p})$ be an instantaneous cost rate, $h(\mathbf{x^p}(T))$ the terminal cost incurred at the end of the process, T a specified final time. Define the cost-to-go function $v^\pi(t, \mathbf{x^p})$ as the total cost expected to accumulate if the system is initialized in state $\mathbf{x^p}$ at time t and controlled until time T according to the control law π

$$v^\pi(t, \mathbf{x^p}) \triangleq E\left[h(\mathbf{x^p}(T)) + \int_t^T \ell(\tau, \mathbf{x^p}(\tau), \pi(\tau, \mathbf{x^p}(\tau)))\,\mathrm{d}\tau\right]. \tag{3}$$

The expectation is taken over the instantiations of the stochastic process ω. The admissible control signals may be constrained: $\mathbf{u^p} \in \mathcal{U}$.

The objective of optimal control is to find the optimal control law $\mathbf{u}^*$ that minimizes $v^\pi(0, \mathbf{x^p}(0))$. Note that the globally-optimal control law does not depend on a specific initial state. However, being able to find this control law in complex problems is unlikely. Instead, we seek locally-optimal control laws: we will present a LQG approximation to our original optimal control problem and compute an approximately-optimal control law. The present formulation in this paper assumes that the state of system is measurable through delayed and noisy sensors. Therefore, we will also design an optimal filter in order to extract the accurate state information from noisy measurement data.

3. Local LQG approximation

3.1 *Linearization*

In this paper the locally-optimal control law is computed using the method of dynamic programming. Each sample time lasts N time steps with $\Delta t = T/N$. Our derived algorithm is iterative. Each iteration starts with a nominal control sequence $\bar{\mathbf{u}}_k^p$, and a corresponding nominal trajectory $\bar{\mathbf{x}}_k^p$ obtained by applying $\bar{\mathbf{u}}_k^p$ to the deterministic system $\dot{\mathbf{x}}^p = f(\mathbf{x}^p, \mathbf{u}^p)$ with $\bar{\mathbf{x}}^p(0) = \mathbf{x}_0^p$. This can be done by Euler integration $\bar{\mathbf{x}}_{k+1}^p = \bar{\mathbf{x}}_k^p + \Delta t f(\bar{\mathbf{x}}_k^p, \bar{\mathbf{u}}_k^p)$.

By linearizing the system dynamics and quadratizing the cost functions around $(\bar{\mathbf{x}}_k^p, \bar{\mathbf{u}}_k^p)$, we obtain a discrete-time linear dynamical system with quadratic cost. Note that the linearized dynamics no longer describe the state and control variables, instead they describe the state and control deviations $\mathbf{x}_k = \mathbf{x}_k^p - \bar{\mathbf{x}}_k^p$, $\mathbf{u}_k = \mathbf{u}_k^p - \bar{\mathbf{u}}_k^p$, and $\mathbf{y}_k = \mathbf{y}_k^p - \bar{\mathbf{y}}_k^p$, where the value of the outputs at the operating point are defined as $\bar{\mathbf{y}}_k^p = g(\bar{\mathbf{x}}_k^p, \bar{\mathbf{u}}_k^p)$. Written in terms of these deviations—state variable $\mathbf{x}_k \in \mathbb{R}^{n_x}$, control input $\mathbf{u}_k \in \mathbb{R}^{n_u}$, and measurement output $\mathbf{y}_k \in \mathbb{R}^{n_y}$, the modified LQG approximation to our original optimal control problem becomes

$$\mathbf{x}_{k+1} = A_k\mathbf{x}_k + B_k\mathbf{u}_k + \mathcal{C}_k(\mathbf{x}_k, \mathbf{u}_k)\xi_k, \quad k = 0, \ldots, N-1 \tag{4}$$

$$\mathbf{y}_k = F_k\mathbf{x}_k + E_k\mathbf{u}_k + \mathcal{D}_k(\mathbf{x}_k, \mathbf{u}_k)\eta_k, \tag{5}$$

$$\mathrm{cost}_k = q_k + \mathbf{x}_k^T\mathbf{q}_k + \frac{1}{2}\mathbf{x}_k^T Q_k\mathbf{x}_k + \mathbf{u}_k^T\mathbf{r}_k + \frac{1}{2}\mathbf{u}_k^T R_k\mathbf{u}_k + \mathbf{u}_k^T P_k\mathbf{x}_k, \tag{6}$$

Where

$$A_k = \frac{\partial f}{\partial \mathbf{x}_p^k}, \quad B_k = \frac{\partial f}{\partial \mathbf{u}_k^p}, \quad F_k = \frac{\partial g}{\partial \mathbf{x}_k^p}, \quad E_k = \frac{\partial g}{\partial \mathbf{u}_k^p}, \tag{7}$$

$$\mathcal{C}_k(\mathbf{x}_k, \mathbf{u}_k) \triangleq \Big[\mathbf{c}_{1,k} + C_{1,k}^{\mathbf{x}}\mathbf{x}_k + C_{1,k}^{\mathbf{u}}\mathbf{u}_k, \ldots, c_{n_\omega,k} + C_{n_\omega,k}^{\mathbf{x}}\mathbf{x}_k + C_{n_\omega,k}^{\mathbf{u}}\mathbf{u}_k\Big], \tag{8}$$

$$\mathcal{D}_k(\mathbf{x}_k, \mathbf{u}_k) \triangleq \Big[\mathbf{d}_{1,k} + D_{1,k}^{\mathbf{x}}\mathbf{x}_k + D_{1,k}^{\mathbf{u}}\mathbf{u}_k, \ldots, \mathbf{d}_{n_v,k} + D_{n_v,k}^{\mathbf{x}}\mathbf{x}_k + D_{n_v,k}^{\mathbf{u}}\mathbf{u}_k\Big], \tag{9}$$

$$\mathbf{c}_{i,k} = \sqrt{\Delta t}\mathcal{F}^{[i]}, \quad C_{i,k}^{\mathbf{x}} = \sqrt{\Delta t}\frac{\partial \mathcal{F}^{[i]}}{\partial \mathbf{x}_k^p}, \quad C_{i,k}^{\mathbf{u}} = \sqrt{\Delta t}\frac{\partial \mathcal{F}^{[i]}}{\partial \mathbf{u}_k^p}, \tag{10}$$

$$\mathbf{d}_{i,k} = \frac{1}{\sqrt{\Delta t}}\mathcal{G}^{[i]}, \quad D_{i,k}^{\mathbf{x}} = \frac{1}{\sqrt{\Delta t}}\frac{\partial \mathcal{G}^{[i]}}{\partial \mathbf{x}_k^p}, \quad D_{i,k}^{\mathbf{u}} = \frac{1}{\sqrt{\Delta t}}\frac{\partial \mathcal{G}^{[i]}}{\partial \mathbf{u}_k^p}, \tag{11}$$

and

$$q_k = \Delta t\ell, \quad \mathbf{q}_k = \Delta t\frac{\partial \ell}{\partial \mathbf{x}_k^p}, \quad Q_k = \Delta t\frac{\partial^2 \ell}{\partial (\mathbf{x}_k^p)^2}, \tag{12}$$

$$\mathbf{r}_k = \Delta t\frac{\partial \ell}{\partial \mathbf{u}_k^p}, \quad R_k = \Delta t\frac{\partial^2 \ell}{\partial (\mathbf{u}_k^p)^2}, \quad P_k = \Delta t\frac{\partial^2 \ell}{\partial \mathbf{u}_k^p \partial \mathbf{x}_k^p}, \tag{13}$$

are computed at each $(\bar{\mathbf{x}}_k^p, \bar{\mathbf{u}}_k^p)$.

The initial state has known mean $\hat{\mathbf{x}}_0$ and covariance Σ_0. All the matrices A_k, B_k, F_k, E_k, $\mathbf{c}_{i,k}$, $C_{i,k}^{\mathbf{x}}$, $C_{i,k}^{\mathbf{u}}$, $\mathrm{d}_{j,k}$, $D_{j,k}^{\mathbf{x}}$, $D_{j,k}^{\mathbf{u}}$ ($i = 1, \ldots, n_\omega$, and $j = 1, \ldots, n_v$) are assumed to be given with the proper dimensions. The independent random variables $\xi_k \in \mathbb{R}^{n_\omega}$ and $\eta_k \in \mathbb{R}^{n_v}$ are zero-mean Gaussian white noises with covariances $\Omega^\xi = I$ and $\Omega^\eta = I$ respectively. Note that $\mathcal{F}^{[i]}$ and $\mathcal{G}^{[i]}$ denote the ith column of matrix $\mathcal{F} \in \mathbb{R}^{n_x \times n_\omega}$ and $\mathcal{G} \in \mathbb{R}^{n_y \times n_v}$ respectively. At the final time step $k = N$, the cost is defined as

$q_N + \mathbf{x}_N^T q_N + \frac{1}{2}\mathbf{x}_N^T Q_N \mathbf{x}_N$, where $q_N = h$, $\mathbf{q}_N = \partial h/\partial \mathbf{x}_N^p$, and $Q_N = \partial^2 h/\partial(\mathbf{x}_N^p)^2$.

Here we are using a noise model which includes control-dependent, state-dependent and additive noises. This is sufficient to capture noise in the system—which is what we are mainly interested in. Considering the sensorimotor control, noise in the motor output increases with the magnitude of the control signal. Incorporating the state-dependent noise in the analysis of sensorimotor control could allow more accurate modelling of feedback form sensory modalities and various experimental perturbations. In the study of estimation and control design for the system with control-dependent and state-dependent noises, the well-known separation principle of standard LQG design is violated. This complicates the problem substantially, and forces us to develop a new structure of recursive controller and estimator.

3.2 *Computing the cost-to-go function (partially observable case)*

In practical situations, the state of the controlled plant is only available through noisy measurement. While the implementation of the optimal control law depends on the state of the system, we have to design an estimator in order to extract the correct information of the state. In this paper we are assuming that the approximately-optimal control law is allowed to be an affine function of $\hat{\mathbf{x}}_k$—the unbiased estimate of state $\mathbf{x}_k$ and the estimator has the form

$$\hat{\mathbf{x}}_{k+1} = A_k\hat{\mathbf{x}}_k + B_k\mathbf{u}_k + K_k(\mathbf{y}_k - F_k\hat{\mathbf{x}}_k - E_u\mathbf{u}_k), \quad (14)$$

where the filter gains K_k are non-adaptive, i.e., they are determined in advance and cannot change as a function of the specific controls and observations within a simulation run. The detailed derivation for computing the filter gain K_k will be presented in § 5.

The approximately-optimal control law for the LQG approximation will be shown to be affine, in the form

$$\mathbf{u}_k = \pi_k(\hat{\mathbf{x}}_k) = l_k + L_k\hat{\mathbf{x}}_k, \quad k = 0, \ldots, N-1, \quad (15)$$

where l_k describes the open-loop control component (it arises because we are dealing with state and control deviations, and is needed to make the algorithm iterative), and L_k is the feedback control gain. The control law we design is approximately-optimal because we may have control constraints and non-convex costs, and also because we use linear Gaussian approximations. Let the cost-to-go function $v_k(\mathbf{x}_k, \hat{\mathbf{x}}_k)$ be the total cost expected to accumulate if the system (4) is initialized in state $\mathbf{x}_k$ at time step k, and controlled according to π_k for the remaining time steps.

Lemma 1: *Suppose the control law π_k for system* (4)–(5) *has already been designed for time steps $k, \ldots, N-1$. If the control law is affine in the form* (15), *then the cost-to-go function $v_k(\mathbf{x}_k, \hat{\mathbf{x}}_k)$ has the form*

$$v_k(\mathbf{x}_k, \hat{\mathbf{x}}_k) = \frac{1}{2}\mathbf{x}_k^T S_k^{\mathbf{x}}\mathbf{x}_k + \frac{1}{2}\hat{\mathbf{x}}_k^T S_k^{\hat{\mathbf{x}}}\hat{\mathbf{x}}_k + \mathbf{x}_k^T S_k^{\mathbf{x}\hat{\mathbf{x}}}\hat{\mathbf{x}}_k + \mathbf{x}_k^T s_k^{\mathbf{x}} + \hat{\mathbf{x}}_k^T \mathbf{s}_k^{\hat{\mathbf{x}}} + s_k \quad (16)$$

for all k. The parameters $S_k^{\mathbf{x}}, S_k^{\hat{\mathbf{x}}}, S_k^{\mathbf{x}\hat{\mathbf{x}}}, \mathbf{s}_k^{\mathbf{x}}, \mathbf{s}_k^{\hat{\mathbf{x}}}$ and s_k for $k < N$ can be computed recursively backwards in time as

$$S_k^{\mathbf{x}} = Q_k + A_k^T S_{k+1}^{\mathbf{x}} A_k + F_k^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k F_k + 2A_k^T S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}} K_k F_k + \sum_{i=1}^{n_\omega} \left(C_{i,k}^{\mathbf{x}}\right)^T S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{x}} + \sum_{i=1}^{n_v} \left(D_{i,k}^{\mathbf{x}}\right)^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{x}}, \quad S_N^{\mathbf{x}} = Q_N, \quad (17)$$

$$S_k^{\hat{\mathbf{x}}} = (A_k - K_k F_k)^T S_{k+1}^{\hat{\mathbf{x}}}(A_k - K_k F_k) + L_k^T H L_k + L_k^T G^{\hat{\mathbf{x}}} + (G^{\hat{\mathbf{x}}})^T L_k, \quad S_N^{\hat{\mathbf{x}}} = 0, \quad (18)$$

$$S_k^{\mathbf{x}\hat{\mathbf{x}}} = F_k^T K_k^T S_{k+1}^{\hat{\mathbf{x}}}(A_k - K_k F_k) + A_k^T S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}(A_k - K_k F_k) + (G^{\mathbf{x}})^T L_k, \quad S_N^{\mathbf{x}\hat{\mathbf{x}}} = 0, \quad (19)$$

$$\mathbf{s}_k^{\mathbf{x}} = \mathbf{q}_k + A_k^T \mathbf{s}_{k+1}^{\mathbf{x}} + F_k^T K_k^T s_{k+1}^{\hat{\mathbf{x}}} + \left(G^{\mathbf{x}}\right)^T l_k + \sum_{i=1}^{n_\omega} \left(C_{i,k}^{\mathbf{x}}\right)^T S_{k+1}^{\mathbf{x}} \mathbf{c}_{i,k} + \sum_{i=1}^{n_v} \left(D_{i,x}^{\mathbf{x}}\right)^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k \mathrm{d}_{i,k}, \quad \mathbf{s}_N^{\mathbf{x}} = \mathbf{q}_N, \quad (20)$$

$$\mathbf{s}_k^{\hat{\mathbf{x}}} = (A_k - K_k F_k)^T \mathbf{s}_{k+1}^{\hat{\mathbf{x}}} + L_k^T H l_k + L_k^T \mathbf{g} + (G^{\hat{\mathbf{x}}})^T l_k, \quad \mathbf{s}_N^{\hat{\mathbf{x}}} = 0, \quad (21)$$

$$\mathbf{s}_k = q_k + s_{k+1} + l_k^T \mathbf{g} + \frac{1}{2} l_k^T H l_k + \frac{1}{2}\left(\sum_{i=1}^{n_\omega} \mathbf{c}_{i,k}^T S_{k+1}^{\mathbf{x}} \mathbf{c}_{i,k} + \sum_{i=1}^{n_v} \mathbf{d}_{i,k}^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k \mathbf{d}_{i,k} \right), \quad s_N = q_N \quad (22)$$

and

$$H \triangleq R_k + B_k^T \left(S_{k+1}^{\mathbf{x}} + S_{k+1}^{\hat{\mathbf{x}}} + 2S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}\right) B_k + \sum_{i=1}^{n_\omega} \left(C_{i,k}^{\mathbf{u}}\right)^T S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{u}} + \sum_{i=1}^{n_v} \left(D_{i,k}^{\mathbf{u}}\right)^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{u}}, \quad (23)$$

$$\mathbf{g} \triangleq \mathbf{r}_k + B_k^T \left(\mathbf{s}_{k+1}^{\mathbf{x}} + \mathbf{s}_{k+1}^{\hat{\mathbf{x}}}\right) + \sum_{i=1}^{n_\omega} \left(C_{i,k}^{\mathbf{u}}\right)^T S_{k+1}^{\mathbf{x}} \mathbf{c}_{i,k} + \sum_{i=1}^{n_v} \left(D_{i,k}^{\mathbf{u}}\right)^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k \mathbf{d}_{i,k}, \quad (24)$$

$$G^{\mathbf{x}} \triangleq P_k + B_k^T(S_{k+1}^{\mathbf{x}} + S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}})A_k + B_k^T(S_{k+1}^{\hat{\mathbf{x}}} + S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}})K_kF_k + \sum_{i=1}^{n_\omega}(C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{x}} + \sum_{i=1}^{n_v}(D_{i,k}^{\mathbf{u}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{x}}, \quad (25)$$

$$G^{\hat{\mathbf{x}}} \triangleq B_k^T\big(S_{k+1}^{\hat{\mathbf{x}}} + S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}\big)(A_k - K_kF_k). \quad (26)$$

Proof: Consider the control law which has been designed for time steps $k, \ldots, N-1$, and at time step k is given by $\mathbf{u}_k = \pi_k(\hat{\mathbf{x}}_k) = l_k + L_k\hat{\mathbf{x}}_k$ (note that in the later derivation we will use the shortcut π_k in place of the control signal $\pi_k(\hat{\mathbf{x}}_k)$ that our control law generates). Let $v_k(\mathbf{x}_k, \hat{\mathbf{x}}_k)$ be the corresponding cost-to-go function, then the Bellman equation is

$$v_k(\mathbf{x}_k, \hat{\mathbf{x}}_k) = \text{immediate cost} + E[v_{k+1}(\mathbf{x}_{k+1}, \hat{\mathbf{x}}_{k+1})|\mathbf{x}_k, \hat{\mathbf{x}}_k, \pi_k]. \quad (27)$$

Based on the dynamics function (4) and (14), the conditional mean and covariance of $\mathbf{x}_{k+1}$ and $\hat{\mathbf{x}}_{k+1}$ are

$$E[\mathbf{x}_{k+1}|\mathbf{x}_k, \hat{\mathbf{x}}_k, \pi_k] = A_k\mathbf{x}_k + B_k\pi_k, \quad (28)$$

$$E[\hat{\mathbf{x}}_{k+1}|\mathbf{x}_k, \hat{\mathbf{x}}_k, \pi_k] = (A_k - K_kF_k)\hat{\mathbf{x}}_k + B_k\pi_k + K_kF_k\mathbf{x}_k, \quad (29)$$

$$\mathrm{Cov}[\mathbf{x}_{k+1}|\mathbf{x}_k, \hat{\mathbf{x}}_k, \pi_k] = \sum_{i=1}^{n_\omega}\big(\mathbf{c}_{i,k} + C_{i,k}^{\mathbf{x}}\mathbf{x}_k + C_{i,k}^{\mathbf{u}}\pi_k\big) \times \big(\mathbf{c}_{i,k} + C_{i,k}^{\mathbf{x}}\mathbf{x}_k + C_{i,k}^{\mathbf{u}}\pi_k\big)^T, \quad (30)$$

$$\mathrm{Cov}[\hat{\mathbf{x}}_{k+1}|\mathbf{x}_k, \hat{\mathbf{x}}_k, \pi_k] = K_k\sum_{i=1}^{n_v}\big(\mathbf{d}_{i,k} + D_{i,k}^{\mathbf{x}}\mathbf{x}_k + D_{i,k}^{\mathbf{u}}\pi_k\big) \times \big(\mathbf{d}_{i,k} + D_{i,k}^{\mathbf{x}}\mathbf{x}_k + D_{i,k}^{\mathbf{u}}\pi_k\big)^T K_k^T. \quad (31)$$

Since

$$E\big[\mathbf{x}_{k+1}\mathbf{x}_{k+1}^T|\mathbf{x}_k, \hat{\mathbf{x}}_k, \pi_k\big] = \mathrm{Cov}[\mathbf{x}_{k+1}|\mathbf{x}_k, \hat{\mathbf{x}}_k, \pi_k] + E[\mathbf{x}_{k+1}|\mathbf{x}_k, \hat{\mathbf{x}}_k, \pi_k](E[\mathbf{x}_{k+1}|\mathbf{x}_k, \hat{\mathbf{x}}_k, \pi_k])^T, \quad (32)$$

$$E\big[\hat{\mathbf{x}}_{k+1}\hat{\mathbf{x}}_{k+1}^T|\mathbf{x}_k, \hat{\mathbf{x}}_k, \pi_k\big] = \mathrm{Cov}[\hat{\mathbf{x}}_{k+1}|\mathbf{x}_k, \hat{\mathbf{x}}_k, \pi_k] + E[\hat{\mathbf{x}}_{k+1}|\mathbf{x}_k, \hat{\mathbf{x}}_k, \pi_k](E[\hat{\mathbf{x}}_{k+1}|\mathbf{x}_k, \hat{\mathbf{x}}_k, \pi_k])^T, \quad (33)$$

$$E\big[\hat{\mathbf{x}}_{k+1}\mathbf{x}_{k+1}^T|\mathbf{x}_k, \hat{\mathbf{x}}_k, \pi_k\big] = ((A_k - K_kF_k)\hat{\mathbf{x}}_k + K_kF_k\mathbf{x}_k + B_k\pi_k) \times (A_k\mathbf{x}_k + B_k\pi_k)^T, \quad (34)$$

applying the formulation of cost-to-go function defined in (16) and substituting (28)–(34) into the conditional expectation in Bellman equation, it yields

$$\begin{aligned}
&E[v_{k+1}(\mathbf{x}_{k+1}, \hat{\mathbf{x}}_{k+1})|\mathbf{x}_k, \hat{\mathbf{x}}_k, \pi_k] \\
&= \frac{1}{2}\mathrm{tr}\,S_{k+1}^{\mathbf{x}}\Bigg[\sum_{i=1}^{n_\omega}\big(\mathbf{c}_{i,k} + C_{i,k}^{\mathbf{x}}\mathbf{x}_k + C_{i,k}^{\mathbf{u}}\pi_k\big)\big(\mathbf{c}_{i,k} + C_{i,k}^{\mathbf{x}}\mathbf{x}_k + C_{i,k}^{\mathbf{u}}\pi_k\big)^T + (A_k\mathbf{x}_k + B_k\pi_k)(A_k\mathbf{x}_k + B_k\pi_k)^T\Bigg] \\
&\quad + \frac{1}{2}\mathrm{tr}S_{k+1}^{\hat{\mathbf{x}}}\Bigg[K_k\sum_{i=1}^{n_v}\big(\mathbf{d}_{i,k} + D_{i,k}^{\mathbf{x}}\mathbf{x}_k + D_{i,k}^{\mathbf{u}}\pi_k\big) \times \big(\mathbf{d}_{i,k} + D_{i,k}^{\mathbf{x}}\mathbf{x}_k + D_{i,k}^{\mathbf{u}}\pi_k\big)^T K_k^T\Bigg] \\
&\quad + \frac{1}{2}\mathrm{tr}S_{k+1}^{\hat{\mathbf{x}}}\Big[\big((A_k - K_kF_k)\hat{\mathbf{x}}_k + K_kF_k\mathbf{x}_k + B_k\pi_k\big) \times \big((A_k - K_kF_k)\hat{\mathbf{x}}_k + K_kF_k\mathbf{x}_k + B_k\pi_k\big)^T\Big] \\
&\quad + \mathrm{tr}S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}[((A_k - K_kF_k)\hat{\mathbf{x}}_k + K_kF_k\mathbf{x}_k + B_k\pi_k) \times (A_k\mathbf{x}_k + B_k\pi_k)^T] + (A_k\mathbf{x}_k + B_k\pi_k)^T\mathbf{s}_{k+1}^{\mathbf{x}} \\
&\quad + ((A_k - K_kF_k)\hat{\mathbf{x}}_k + K_kF_k\mathbf{x}_k + B_k\pi_k)^T s_{k+1}^{\hat{\mathbf{x}}} + s_{k+1}.
\end{aligned}$$

Using the fact that $\mathrm{tr}(UV) = tr(VU)$ in the above equation, and substituting the immediate cost (6) and the above equation into (27), the resulting cost-to-go function becomes

$$\begin{aligned}
&v_k(\mathbf{x}_k, \hat{\mathbf{x}}_k) \\
&= \frac{1}{2}\mathbf{x}_k^T\Bigg[Q_k + A_k^T S_{k+1}^{\mathbf{x}} A_k + F_k^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k F_k + 2A_k^T S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}} K_k F_k + \sum_{i=1}^{n_\omega}(C_{i,k}^{\mathbf{x}})^T S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{x}} \\
&\quad + \sum_{i=1}^{n_v}(D_{i,k}^{\mathbf{x}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{x}}\Bigg]\mathbf{x}_k \\
&\quad + \frac{1}{2}\hat{\mathbf{x}}_k^T(A_k - K_kF_k)^T S_{k+1}^{\hat{\mathbf{x}}}(A_k - K_kF_k)\hat{\mathbf{x}}_k \\
&\quad + \frac{1}{2}\pi_k^T\Bigg[R_k + B_k^T(S_{k+1}^{\mathbf{x}} + S_{k+1}^{\hat{\mathbf{x}}} + 2S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}})B_k + \sum_{i=1}^{n_\omega}(C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{u}} + \sum_{i=1}^{n_v}(D_{i,k}^{\mathbf{u}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{u}}\Bigg]\pi_k \\
&\quad + \mathbf{x}_k^T\Big[F_k^T K_k^T S_{k+1}^{\hat{\mathbf{x}}}(A_k - K_kF_k) + A_k^T S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}(A_k - K_kF_k)\Big]\hat{\mathbf{x}}_k \\
&\quad + \pi_k^T\Bigg[P_k + B_k^T(S_{k+1}^{\mathbf{x}} + S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}})A_k + B_k^T(S_{k+1}^{\hat{\mathbf{x}}} + S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}})K_kF_k + \sum_{i=1}^{n_\omega}(C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{x}} + \sum_{i=1}^{n_v}(D_{i,k}^{\mathbf{u}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{x}}\Bigg]\mathbf{x}_k \\
&\quad + \pi_k^T B_k^T(S_{k+1}^{\hat{\mathbf{x}}} + S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}})(A_k - K_kF_k)\hat{\mathbf{x}}_k
\end{aligned}$$

$$
\begin{aligned}
&+\mathbf{x}_k^T\Bigg[\mathbf{q}_k+\sum_{i=1}^{n_\omega}(C_{i,k}^{\mathbf{x}})^T S_{k+1}^{\mathbf{x}}\mathbf{c}_{i,k}+\sum_{i=1}^{n_v}(D_{i,k}^{\mathbf{x}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}}K_k\mathbf{d}_{i,k}\\
&+A_k^T\mathbf{s}_{k+1}^{\mathbf{x}}+F_k^T K_k^T\mathbf{s}_{k+1}^{\hat{\mathbf{x}}}\Bigg]\\
&+\hat{\mathbf{x}}_k^T(A_k-K_kF_k)^T\mathbf{s}_{k+1}^{\hat{\mathbf{x}}}\\
&+\pi_k^T\Bigg[\mathbf{r}_k+\sum_{i=1}^{n_\omega}(C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}}\mathbf{c}_{i,k}+\sum_{i=1}^{n_v}(D_{i,k}^{\mathbf{u}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}}K_k\mathbf{d}_{i,k}\\
&+B_k^T(\mathbf{s}_{k+1}^{\mathbf{x}}+\mathbf{s}_{k+1}^{\hat{\mathbf{x}}})\Bigg]\\
&+q_k+s_{k+1}+\frac{1}{2}\Bigg(\sum_{i=1}^{n_\omega}\mathbf{c}_{i,k}^T S_{k+1}^{\mathbf{x}}\mathbf{c}_{i,k}+\sum_{i=1}^{n_v}\mathbf{d}_{i,k}^T K_k^T S_{k+1}^{\hat{\mathbf{x}}}K_k\mathbf{d}_{i,k}\Bigg).
\end{aligned}
\tag{35}
$$

Substituting (23)–(26) into the above equation, the π_k-dependent terms in (35) becomes

$$
\frac{1}{2}\pi_k^T H\pi_k+\pi_k^T\big(\mathbf{g}+G^{\mathbf{x}}\mathbf{x}_k+G^{\hat{\mathbf{x}}}\hat{\mathbf{x}}_k\big).
\tag{36}
$$

Since we assume that the control law has the general form given in (15), replacing π_k with $l_k+L_k\hat{\mathbf{x}}_k$, it yields

$$
\begin{aligned}
&\frac{1}{2}\hat{\mathbf{x}}_k^T\big(L_k^T HL_k+L_k^T G^{\hat{\mathbf{x}}}+(G^{\hat{\mathbf{x}}})^T L_k\big)\hat{\mathbf{x}}_k+\mathbf{x}_k^T(G^{\mathbf{x}})^T L_k\hat{\mathbf{x}}_k\\
&\quad+\mathbf{x}_k^T(G^{\mathbf{x}})^T l_k+\hat{\mathbf{x}}_k^T\Big(L_k^T Hl_k+L_k^T\mathbf{g}+(G^{\hat{\mathbf{x}}})^T l_k\Big)\\
&\quad+l_k^T\mathbf{g}+\frac{1}{2}l_k^T Hl_k.
\end{aligned}
\tag{37}
$$

Now the cost-to-go function $v_k(\mathbf{x}_k,\hat{\mathbf{x}}_k)$ becomes

$$
\begin{aligned}
&v_k(\mathbf{x}_k,\hat{\mathbf{x}}_k)\\
&\quad=\frac{1}{2}\mathbf{x}_k^T\Bigg[Q_k+A_k^T S_{k+1}^{\mathbf{x}}A_k+F_k^T K_k^T S_{k+1}^{\hat{\mathbf{x}}}K_kF_k\\
&\qquad+2A_k^T S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}K_kF_k+\sum_{i=1}^{n_\omega}(C_{i,k}^{\mathbf{x}})^T S_{k+1}^{\mathbf{x}}C_{i,k}^{\mathbf{x}}\\
&\qquad+\sum_{i=1}^{n_v}(D_{i,k}^{\mathbf{x}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}}K_kD_{i,k}^{\mathbf{x}}\Bigg]\mathbf{x}_k\\
&\qquad+\frac{1}{2}\hat{\mathbf{x}}_k^T\Big[(A_k-K_kF_k)^T S_{k+1}^{\hat{\mathbf{x}}}(A_k-K_kF_k)+L_k^T HL_k\\
&\qquad+L_k^T G^{\hat{\mathbf{x}}}+(G^{\hat{\mathbf{x}}})^T L_k\Big]\hat{\mathbf{x}}_k\\
&\qquad+\mathbf{x}_k^T\Big[F_k^T K_k^T S_{k+1}^{\hat{\mathbf{x}}}(A_k-K_kF_k)\\
&\qquad+A_k^T S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}(A_k-K_kF_k)+(G^{\mathbf{x}})^T L_k\Big]\hat{\mathbf{x}}_k\\
&\qquad+\mathbf{x}_k^T\Bigg[\mathbf{q}_k+\sum_{i=1}^{n_\omega}(C_{i,k}^{\mathbf{x}})^T S_{k+1}^{\mathbf{x}}\mathbf{c}_{i,k}\\
&\qquad+\sum_{i=1}^{n_v}(D_{i,k}^{\mathbf{x}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}}K_k\mathbf{d}_{i,k}+A_k^T\mathbf{s}_{k+1}^{\mathbf{x}}\\
&\qquad+F_k^T K_k^T\mathbf{s}_{k+1}^{\hat{\mathbf{x}}}+(G^{\mathbf{x}})^T l_k\Bigg]\\
&\qquad+\hat{\mathbf{x}}_k^T\Big[(A_k-K_kF_k)^T\mathbf{s}_{k+1}^{\hat{\mathbf{x}}}+L_k^T Hl_k+L_k^T\mathbf{g}+(G^{\hat{\mathbf{x}}})^T l_k\Big]\\
&\qquad+q_k+\mathbf{s}_{k+1}+\frac{1}{2}\Bigg(\sum_{i=1}^{n_\omega}\mathbf{c}_{i,k}^T S_{k+1}^{\mathbf{x}}\mathbf{c}_{i,k}\\
&\qquad+\sum_{i=1}^{n_v}\mathbf{d}_{i,k}^T K_k^T S_{k+1}^{\hat{\mathbf{x}}}K_k\mathbf{d}_{i,k}\Bigg)+l_k^T\mathbf{g}+\frac{1}{2}l_k^T Hl_k.
\end{aligned}
\tag{38}
$$

By applying the defined formulation of cost-to-go function given in (16), we can obtain (17)–(22) immediately which completes the proof.

3.3 *Computing the cost-to-go function (fully observable case)*

Suppose the state of system (4) is available for measurement in the implementation of the optimal control design, then Lemma 1 readily leads to the following corollary.

Corollary 1: *Suppose the control law π_k for system* (4) *has already been designed for time steps $k,\ldots,N-1$. If the control law is affine in the form $\mathbf{u}_k=l_k+L_k\mathbf{x}_k$, $k=0,\ldots,N-1$, then the cost-to-go function $v_k(\mathbf{x}_k)$ has the form*

$$
v_k(\mathbf{x}_k)=\frac{1}{2}\mathbf{x}_k^T S_k^{\mathbf{x}}\mathbf{x}_k+\mathbf{x}_k^T\mathbf{s}_k^{\mathbf{x}}+s_k
\tag{39}
$$

where the parameters $S_k^{\mathbf{x}}$, $\mathbf{s}_k^{\mathbf{x}}$, and s_k can be computed recursively backwards in time as

$$
\begin{aligned}
S_k^{\mathbf{x}}&=Q_k+A_k^T S_{k+1}^{\mathbf{x}}A_k+\sum_{i=1}^{n_\omega}(C_{i,k}^{\mathbf{x}})^T S_{k+1}^{\mathbf{x}}C_{i,k}^{\mathbf{x}}+L_k^T HL_k\\
&\quad+L_k^T G+G^T L_k,\quad S_N^{\mathbf{x}}=Q_N,
\end{aligned}
\tag{40}
$$

$$
\begin{aligned}
\mathbf{s}_k^{\mathbf{x}}&=\mathbf{q}_k+A_k^T\mathbf{s}_{k+1}^{\mathbf{x}}+\sum_{i=1}^{n_\omega}(C_{i,k}^{\mathbf{x}})^T S_{k+1}^{\mathbf{x}}\mathbf{c}_{i,k}\\
&\quad+L_k^T Hl_k+L_k^T\mathbf{g}+G^T l_k,\quad \mathbf{s}_N^{\mathbf{x}}=\mathbf{q}_N,
\end{aligned}
\tag{41}
$$

$$
\begin{aligned}
s_k&=q_k+s_{k+1}+\frac{1}{2}\sum_{i=1}^{n_\omega}\mathbf{c}_{i,k}^T S_{k+1}^{\mathbf{x}}\mathbf{c}_{i,k}+\frac{1}{2}l_k^T Hl_k+l_k^T\mathbf{g},\\
&\quad s_N=q_N,
\end{aligned}
\tag{42}
$$

and

$$
H\triangleq R_k+B_k^T S_{k+1}^{\mathbf{x}}B_k+\sum_{i=1}^{n_\omega}(C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}}C_{i,k}^{\mathbf{u}},
\tag{43}
$$

$$
\mathbf{g}\triangleq\mathbf{r}_k+B_k^T s_{k+1}^{\mathbf{x}}+\sum_{i=1}^{n_\omega}(C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}}\mathbf{c}_{i,k},
\tag{44}
$$

$$
G\triangleq P_k+B_k^T S_{k+1}^{\mathbf{x}}A_k+\sum_{i=1}^{n_\omega}(C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}}C_{i,k}^{\mathbf{x}}.
\tag{45}
$$

4. Controller design

As we saw in (35), the cost-to-go function $v_k(\mathbf{x}_k, \hat{\mathbf{x}}_k)$ depends on the control $\mathbf{u}_k = \pi_k(\hat{\mathbf{x}}_k)$ through the term

$$a(\mathbf{x}_k, \hat{\mathbf{x}}_k, \pi_k) = \frac{1}{2}\pi_k^T H \pi_k + \pi_k^T(\mathbf{g} + G^{\mathbf{x}}\mathbf{x}_k + G^{\hat{\mathbf{x}}}\hat{\mathbf{x}}_k).$$

This expression is quadratic in π_k and can be minimized analytically, but the problem is that the minimum depends on $\mathbf{x}_k$ while π_k is only a function of $\hat{\mathbf{x}}_k$. To obtain the optimal control law at time step k, we have to take an expectation over $\mathbf{x}_k$ conditional on $\hat{\mathbf{x}}_k$, and find the function π_k that minimizes the resulting expression. Since $E[\mathbf{x}_k|\hat{\mathbf{x}}_k] = \hat{\mathbf{x}}_k$, we have

$$\begin{aligned}\alpha(\hat{\mathbf{x}}_k, \pi_k) &\triangleq E[a(\mathbf{x}_k, \hat{\mathbf{x}}_k, \pi_k)|\hat{\mathbf{x}}_k] \\ &= \frac{1}{2}\pi_k^T H \pi_k + \pi_k^T(g + G\hat{\mathbf{x}}_k),\end{aligned} \tag{46}$$

where $G = G^{\mathbf{x}} + G^{\hat{\mathbf{x}}}$. Ideally we would choose π_k that minimizes $\alpha(\hat{\mathbf{x}}_k, \pi_k)$ subject to whatever control constraints are present. However, this is not always possible within the family of affine control laws $\pi_k(\hat{\mathbf{x}}_k) = l_k + L_k\hat{\mathbf{x}}_k$ that we are considering. Since the goal of the LQG stage is to approximate the optimal controller for the non-linear system in the vicinity of $\bar{\mathbf{x}}_k^p$, we will give preference to those control laws that are optimal/feasible for small $\mathbf{x}_k$, even if that (unavoidably) makes them sub-optimal/infeasible for larger $\mathbf{x}_k$.

4.1 *Second-order methods*

If the symmetric matrix H in (46) is positive semi-definite, we can compute the unconstrained optimal control law

$$\pi_k = -H^{-1}(\mathbf{g} + G\hat{\mathbf{x}}_k), \tag{47}$$

and deal with the control constraints as described below, but when H has negative eigenvalues, there exist $\pi_k's$ that make a arbitrarily negative. Note that the cost-to-go function for the non-linear problem is always non-negative, but as we are using an approximation to the true cost, we may encounter situations where a does not have a minimum. In that case we use $\mathcal{H}$ to resemble H, because H still contains correct second-order information; and so the true cost-to-go decreases in the direction $-\mathcal{H}^{-1}(\mathbf{g} + G\hat{\mathbf{x}}_k)$ for any positive definite matrix $\mathcal{H}$.

One possibility is to set $\mathcal{H} = H + (\epsilon - \lambda_{\min}(H))I$ where $\lambda_{\min}(H)$ is the minimum eigenvalue of H and $\epsilon > 0$. This is related to the Levenberg–Marquardt method, and has the potentially undesirable effect of increasing all eigenvalues of H and not just those that are negative. Another possibility is to compute the eigenvalue decomposition $[V, D] = \text{eig}(H)$, replace all elements of the diagonal matrix D that are smaller than ϵ with ϵ (obtaining a new diagonal matrix $\mathcal{D}$), and then set $\mathcal{H} = V\mathcal{D}V^T$. The eigenvalue decomposition is not a significant slowdown, because we have to perform a matrix inversion anyway and we can do so by $\mathcal{H}^{-1} = V\mathcal{D}^{-1}V^T$. It is not yet clear which of the two methods works better in practice. Note that we may also want to use $\mathcal{H}$ instead of H when the eigenvalues are positive but very small—because in that case H^{-1} can cause very large control signal that will push the original system outside the range of validity of our LQG approximation.

Lemma 2: *The optimal control law is computed as*

$$\begin{aligned}
&\mathbf{u}_k = l_k + L_k\hat{\mathbf{x}}_k, \quad k = 0, \ldots, N-1, \\
&l_k = -\mathcal{H}^{-1}\mathbf{g}, \quad L_k = -\mathcal{H}^{-1}G, \\
&\mathcal{H} = H + (\epsilon - \lambda_{\min}(H))I, \quad \epsilon > 0, \\
&H \triangleq B_k^T(S_{k+1}^{\mathbf{x}} + S_{k+1}^{\hat{\mathbf{x}}} + 2S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}})B_k + \sum_{i=1}^{n_\omega}(C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{u}} \\
&\quad + \sum_{i=1}^{n_v}(D_{i,k}^{\mathbf{u}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{u}}, \\
&\mathbf{g} \triangleq \mathbf{r}_k + B_k^T(s_{k+1}^{\mathbf{x}} + s_{k+1}^{\hat{\mathbf{x}}}) + \sum_{i=1}^{n_\omega}(C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}} \mathbf{c}_{i,k} \\
&\quad + \sum_{i=1}^{n_v}(D_{i,k}^{\mathbf{u}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k \mathbf{d}_{i,k}, \\
&G \triangleq P_k + B_k^T(S_{k+1}^{\mathbf{x}} + S_{k+1}^{\hat{\mathbf{x}}} + 2S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}})A_k \\
&\quad + \sum_{i=1}^{n_\omega}(C_{i,k}^{\mathbf{u}})^T S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{x}} + \sum_{i=1}^{n_v}(D_{i,k}^{\mathbf{u}})^T K_k^T S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{x}},
\end{aligned} \tag{48}$$

where $S_{k+1}^{\mathbf{x}}$, $S_{k+1}^{\hat{\mathbf{x}}}$, $S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}$, $\mathbf{s}_{k+1}^{\mathbf{x}}$, $s_{k+1}^{\hat{\mathbf{x}}}$, $\mathbf{s}_{k+1}$ *can be obtained through* (17)–(22) *backwards in time.*

4.2 *Constrained second-order methods*

The problem here is to find the control law $\mathbf{u}_k = \pi_k(\hat{\mathbf{x}}_k) = l_k + L_k\hat{\mathbf{x}}_k$ minimizing (46) subject to constraints $\mathbf{u}_k + \bar{\mathbf{u}}_k^p \in \mathcal{U}$, assuming that H has already been replaced with a positive definite $\mathcal{H}$ (see the above section). Given that $\mathbf{x}_k$ is unconstrained, the only general way to enforce the constraints $\mathcal{U}$ is to set $L_k = 0$. In practice we do not want to be that conservative, since we are looking for an approximation to the non-linear problem that is valid around $\mathbf{x}_k = 0$. Either way we can ignore the $L_k\hat{\mathbf{x}}_k$ term in the constraint satisfaction phase, and come back to the computation of L_k after the open-loop component l_k has been determined.

The unconstrained minimum of $\mathbf{u}_k^T\mathbf{g} + \frac{1}{2}\mathbf{u}_k^T\mathcal{H}\mathbf{u}_k$ is $\mathbf{u}_k^* = -H^{-1}\mathbf{g}$. If it satisfies the constraint $\mathbf{u}_k^* + \bar{\mathbf{u}}_k^p \in \mathcal{U}$ we are done. Otherwise we have two options. The more efficient but less accurate method is to backtrack once, i.e. to find the maximal $\epsilon \in [0; 1]$ such that $\epsilon\mathbf{u}_k^* + \bar{\mathbf{u}}_k^p \in \mathcal{U}$.

This is appropriate in the early phase of the iterative algorithm when the nominal trajectory $\bar{\mathbf{x}}_k^p$ is still far away from $\bar{\mathbf{x}}_k^{p^*}$; in that phase it makes more sense to quickly improve the control law rather than refine the solution to a LQG problem that is an inaccurate approximation to the original problem. However in the final phase of the iterative algorithm we want to obtain the best control law possible for the given LQG problem. In that phase we use quadratic programming. When the constraint set is specified by a collection of linear inequalities, and given that $\mathcal{H}$ is positive definite, the active set algorithm (which is a greedy quadratic programming method) can be used to quickly find the global constrained minimum.

Once the open-loop component l_k is determined, we have to compute the feedback gain matrix L_k. If $l_k + \bar{\mathbf{u}}_k^p$ is inside $\mathcal{U}$, small changes $L_k\hat{\mathbf{x}}_k$ will not cause constraint violations and so we can use the optimal $L_k = -\mathcal{H}^{-1}G$. But if $l_k + \bar{\mathbf{u}}_k^p$ lies on the constraint boundary $\partial\mathcal{U}$, we have to modify L_k so that $L_k\hat{\mathbf{x}}_k$ can only cause changes along the boundary. This is not only because we want to avoid constraint violations. The fact that $l_k + \bar{\mathbf{u}}_k^p$ is on $\partial\mathcal{U}$ means that the unconstrained minimum $\mathbf{u}_k^*$ is actually outside $\mathcal{U}$, and so a change of $L_k\hat{\mathbf{x}}_k$ orthogonal to the boundary ∂U cannot produce a better feasible control.

Modifying L_k is straightforward in the typical case when the range of each element of $\mathbf{u}_k$ is specified independently. In that case we simply set to zero the rows of $-\mathcal{H}^{-1}G$ corresponding to elements of $l_k + \bar{\mathbf{u}}_k^p$ that have reached their limits.

5. Estimator design

It is well known that, for models with control-dependent and state-dependent noises, the optimal filter is very difficult to compute in practice. For this kind of model, the construction of suboptimal filters that approximate the optimal one becomes very important.

So far we computed the optimal control law for any fixed sequence of filter gains K_k. In order to preserve the optimality of the control law obtained in the previous section and attain an iterative algorithm with guaranteed convergence, we need to compute a fixed sequence of filter gains that are optimal for a given control law. Thus our objective here is the following: given the control law $\mathbf{u}_0, \ldots, \mathbf{u}_{N-1}$ (which is optimal for the previous filter $K_0, \ldots, K_{N-1}$), compute a new suboptimal filter evaluated by minimizing the magnitude of its estimation errors, in conjunction with the given control law, which results in minimal expected cost. Once the iterative algorithm has converged and the control law has been designed, we could use an adaptive filter in place of the fixed-gain filter in run time.

Lemma 3: *By defining the unconditional means* $m_k^e \triangleq E[e_k]$, $m_k^{\hat{\mathbf{x}}} \triangleq E[\hat{\mathbf{x}}_k]$, *and the unconditional covariances* $\Sigma_k^e \triangleq E[e_k e_k^T]$, $\Sigma_k^{\hat{\mathbf{x}}} \triangleq E[\hat{\mathbf{x}}_k \hat{\mathbf{x}}_k^T]$, *and* $\Sigma_k^{\hat{\mathbf{x}}e} \triangleq E[\hat{\mathbf{x}}_k e_k^T]$, *and assuming that the initial state of system has known mean* $\hat{\mathbf{x}}_0$ *and covariance* Σ_0, *the optimal filter gain for system* (4)–(5) *is computed as*

$$\hat{\mathbf{x}}_{k+1} = A_k\hat{\mathbf{x}}_k + B_k\pi_k + K_k(y_k - F_k\hat{\mathbf{x}}_k - E_k\pi_k), \tag{49}$$

$$K_k = A_k\Sigma_k^e F_k^T\left(F_k\Sigma_k^e F_k^T + \mathcal{P}_k\right)^{-1}, \tag{50}$$

$$m_{k+1}^{\hat{\mathbf{x}}} = (A_k + B_kL_k)m_k^{\hat{\mathbf{x}}} + K_kF_km_k^e + B_kl_k,$$

$$m_0^{\hat{\mathbf{x}}} = \hat{\mathbf{x}}_0, \tag{51}$$

$$m_{k+1}^e = (A_k - K_kF_k)m_k^e, \quad m_0^e = 0, \tag{52}$$

$$\begin{aligned}\Sigma_{k+1}^{\hat{\mathbf{x}}} = {} & (A_k + B_kL_k)\Sigma_k^{\hat{\mathbf{x}}}(A_k + B_kL_k)^T + K_kF_k\Sigma_k^eA_k^T \\ & + (A_k + B_kL_k)\Sigma_k^{\hat{\mathbf{x}}e}F_k^TK_k^T + K_kF_k\Sigma_k^{e\hat{\mathbf{x}}}(A_k + B_kL_k)^T \\ & + \left((A_k + B_kL_k)m_k^{\hat{\mathbf{x}}} + K_kF_km_k^e\right)l_k^TB_k^T \\ & + B_kl_k\left((A_k + B_kL_k)m_k^{\hat{\mathbf{x}}} + K_kF_km_k^e\right)^T \\ & + B_kl_kl_k^TB_k^T, \quad \Sigma_0^{\hat{\mathbf{x}}} = \hat{\mathbf{x}}_0\hat{\mathbf{x}}_0^T,\end{aligned} \tag{53}$$

$$\Sigma_{k+1}^e = (A_k - K_kF_k)\Sigma_k^eA_k^T + \mathcal{M}_k, \quad \Sigma_0^e = \Sigma_0, \tag{54}$$

$$\begin{aligned}\Sigma_{k+1}^{\hat{\mathbf{x}}e} = {} & (A_k + B_kL_k)\Sigma_k^{\hat{\mathbf{x}}e}(A_k - K_kF_k)^T \\ & + B_kl_k\left(m_k^e\right)^T(A_k - K_kF_k)^T, \quad \Sigma_0^{\hat{\mathbf{x}}e} = 0,\end{aligned} \tag{55}$$

$$\begin{aligned}\mathcal{P}_k = \sum_{i=1}^{n_v}\Bigg[& \mathbf{d}_{i,k}\mathbf{d}_{i,k}^T + \mathbf{d}_{i,k}\left(m_k^{\hat{\mathbf{x}}} + m_k^e\right)^T\left(D_{i,k}^{\mathbf{x}}\right)^T \\ & + D_{i,k}^{\mathbf{x}}\left(m_k^{\hat{\mathbf{x}}} + m_k^e\right)\mathbf{d}_{i,k}^T + \mathbf{d}_{i,k}\left(l_k + L_km_k^{\hat{\mathbf{x}}}\right)^T\left(D_{i,k}^{u}\right)^T \\ & + D_{i,k}^{\mathbf{u}}\left(l_k + L_km_k^{\hat{\mathbf{x}}}\right)\mathbf{d}_{i,k}^T \\ & + D_{i,k}^{\mathbf{x}}\left((m_k^{\hat{\mathbf{x}}} + m_k^e)l_k^T + \left(\Sigma_k^{\hat{\mathbf{x}}} + \Sigma_k^{e\hat{\mathbf{x}}}\right)L_k^T\right)\left(D_{i,k}^{\mathbf{u}}\right)^T \\ & + D_{i,k}^{\mathbf{u}}\left(l_k(m_k^{\hat{\mathbf{x}}} + m_k^e)^T + L_k\left(\Sigma_k^{\hat{\mathbf{x}}} + \Sigma_k^{\hat{\mathbf{x}}e}\right)\right)\left(D_{i,k}^{\mathbf{x}}\right)^T \\ & + D_{i,k}^{\mathbf{x}}\left(\Sigma_k^{\hat{\mathbf{x}}} + \Sigma_k^{\hat{\mathbf{x}}e} + \Sigma_k^{e\hat{\mathbf{x}}} + \Sigma_k^e\right)\left(D_{i,k}^{\mathbf{x}}\right)^T \\ & + D_{i,k}^{\mathbf{u}}\left(l_kl_k^T + l_k\left(m_k^{\hat{\mathbf{x}}}\right)^TL_k^T + L_km_k^{\hat{\mathbf{x}}}l_k^T + L_k\Sigma_k^{\hat{\mathbf{x}}}L_k^T\right) \\ & \times\left(D_{i,k}^{\mathbf{u}}\right)^T\Bigg],\end{aligned} \tag{56}$$

$$\mathcal{M}_k = \sum_{i=1}^{n_\omega} \Big[\mathbf{c}_{i,k}\mathbf{c}_{i,k}^T + \mathbf{c}_{i,k}\big(m_k^{\hat{\mathbf{x}}} + m_k^e\big)^T \big(C_{i,k}^{\mathbf{x}}\big)^T + C_{i,k}^{\mathbf{x}}\big(m_k^{\hat{\mathbf{x}}} + m_k^e\big)\mathbf{c}_{i,k}^T + \mathbf{c}_{i,k}\big(l_k + L_k m_k^{\hat{\mathbf{x}}}\big)^T \big(C_{i,k}^{\mathbf{u}}\big)^T + C_{i,k}^{\mathbf{u}}\big(l_k + L_k m_k^{\hat{\mathbf{x}}}\big)\mathbf{c}_{i,k}^T + C_{i,k}^{\mathbf{x}}\Big(\big(m_k^{\hat{\mathbf{x}}} + m_k^e\big)l_k^T + \big(\Sigma_k^{\hat{\mathbf{x}}} + \Sigma_k^{e\hat{\mathbf{x}}}\big)L_k^T\Big)\big(C_{i,k}^{\mathbf{u}}\big)^T + C_{i,k}^{\mathbf{u}}\Big(l_k\big(m_k^{\hat{\mathbf{x}}} + m_k^e\big)^T + L_k\big(\Sigma_k^{\hat{\mathbf{x}}} + \Sigma_k^{\hat{\mathbf{x}}e}\big)\Big)\big(C_{i,k}^{\mathbf{x}}\big)^T + C_{i,k}^{\mathbf{x}}\Big(\Sigma_k^{\hat{\mathbf{x}}} + \Sigma_k^{\hat{\mathbf{x}}e} + \Sigma_k^{e\hat{\mathbf{x}}} + \Sigma_k^e\Big)\big(C_{i,k}^{\mathbf{x}}\big)^T + C_{i,k}^{\mathbf{u}}\big(l_k l_k^T + l_k\big(m_k^{\hat{\mathbf{x}}}\big)^T L_k^T + L_k m_k^{\hat{\mathbf{x}}} l_k^T + L_k\Sigma_k^{\hat{\mathbf{x}}}L_k^T\big)\big(C_{i,k}^{\mathbf{u}}\big)^T\Big]. \tag{57}$$

Proof: Rewrite the system dynamics and state estimator as the following:

$$\begin{aligned}\mathbf{x}_{k+1} &= A_k\mathbf{x}_k + B_k\mathbf{u}_k + \mathcal{C}_k(\mathbf{x}_k,\mathbf{u}_k)\xi_k,\\ \mathbf{y}_k &= F_k\mathbf{x}_k + E_k\mathbf{u}_k + \mathcal{D}_k(\mathbf{x}_k,\mathbf{u}_k)\eta_k,\\ \hat{\mathbf{x}}_{k+1} &= A_k\hat{\mathbf{x}}_k + B_k\mathbf{u}_k + K_k(\mathbf{y}_k - F_k\hat{\mathbf{x}}_k - E_k\mathbf{u}_k),\end{aligned}$$

where $\mathbf{u}_k = \pi_k(\hat{\mathbf{x}}_k) = l_k + L_k\hat{\mathbf{x}}_k$, $(k = 0,\ldots,N-1)$, and K_k is the filter gain that minimizes the functional

$$J = E\big[e_{k+1}^T\mathcal{T}e_{k+1}\big], \quad \mathcal{T} \geq 0, \tag{58}$$

where the estimation error $e_{k+1} = \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}$, and the estimation error dynamics are given by

$$e_{k+1} = (A_k - K_kF_k)e_k + \mathcal{C}_k(\mathbf{x}_k,\pi_k)\xi_k - K_k\mathcal{D}_k(\mathbf{x}_k,\pi_k)\eta_k. \tag{59}$$

Note that we use the shortcut π_k in place of the control signal for the convenience. Based on the estimation error dynamics (59), the conditional mean and covariance of e_{k+1} are

$$E[e_{k+1}|\mathbf{x}_k,\hat{\mathbf{x}}_k,\pi_k] = (A_k - K_kF_k)e_k, \tag{60}$$

$$\mathrm{Cov}[e_{k+1}|\mathbf{x}_k,\hat{\mathbf{x}}_k,\pi_k] = \mathcal{C}_k(\mathbf{x}_k,\pi_k)\mathcal{C}_k(\mathbf{x}_k,\pi_k)^T + K_k\mathcal{D}_k(\mathbf{x}_k,\pi_k)\mathcal{D}_k(\mathbf{x}_k,\pi_k)^TK_k^T. \tag{61}$$

By applying the properties of conditional expectation, we obtain

$$\begin{aligned}&E[e_{k+1}e_{k+1}^T|\mathbf{x}_k,\hat{\mathbf{x}}_k,\pi_k]\\ &\quad = \mathrm{Cov}[e_{k+1}|\mathbf{x}_k,\hat{\mathbf{x}}_k,\pi_k]\\ &\qquad + E[e_{k+1}|\mathbf{x}_k,\hat{\mathbf{x}}_k,\pi_k](E[e_{k+1}|\mathbf{x}_k,\hat{\mathbf{x}}_k,\pi_k])^T,\end{aligned} \tag{62}$$

$$E[e_{k+1}e_{k+1}^T] = E\big[E[e_{k+1}e_{k+1}^T|\mathbf{x}_k,\hat{\mathbf{x}}_k,\pi_k]\big]. \tag{63}$$

The terms in $E[e_{k+1}e_{k+1}^T|\mathbf{x}_k,\hat{\mathbf{x}}_k,\pi_k]$ that dependent on K_k are

$$(A_k - K_kF_k)e_ke_k^T(A_k - K_kF_k)^T + K_k\mathcal{D}_k(\mathbf{x}_k,\pi_k)\mathcal{D}_k(\mathbf{x}_k,\pi_k)^TK_k^T.$$

With the definition of $\Sigma_k^e \triangleq E\big[e_ke_k^T\big]$ and

$$\mathcal{P}_k = E\big[\mathcal{D}_k(\mathbf{x}_k,\pi_k)\mathcal{D}_k^T(\mathbf{x}_k,\pi_k)\big], \tag{64}$$

the unconditional expectation of the K_k-dependent expression above becomes

$$(A_k - K_kF_k)\Sigma_k^e(A_k - K_kF_k)^T + K_k\mathcal{P}_kK_k^T.$$

Using the fact that $E[e_{k+1}^T\mathcal{T}e_{k+1}] = \mathrm{tr}(E[e_{k+1}e_{k+1}^T]\mathcal{T})$, it follows that the K_k-dependent terms in J becomes

$$\beta(K_k) = \mathrm{tr}\mathcal{T}(A_k - K_kF_k)\Sigma_k^e(A_k - K_kF_k)^T + \mathrm{tr}\mathcal{T}K_k\mathcal{P}_kK_k^T, \tag{65}$$

and that the minimum of $\beta(K_k)$ is found by setting the derivative with respect to K_k to zero. Using the matrix identities $(\partial/\partial X)\mathrm{tr}(XA) = A^T$ and $(\partial/\partial X)\mathrm{tr}(AXBX^T) = A^TXB^T + AXB$, we obtain

$$\frac{\partial\beta(K_k)}{\partial K_k} = \mathcal{T}\big(K_k\big(F_k\Sigma_k^eF_k^T + \mathcal{P}_k\big) - A_k\Sigma_k^eF_k^T\big) = 0. \tag{66}$$

Hence

$$K_k = A_k\Sigma_k^eF_k^T\big(F_k\Sigma_k^eF_k^T + \mathcal{P}_k\big)^{-1}. \tag{67}$$

To complete the proof of the lemma, we need to compute the unconditional covariance. By substituting the control law $\mathbf{u}_k = \pi_k(\hat{\mathbf{x}}_k) = l_k + L_k\hat{\mathbf{x}}_k$, we can rewrite the state estimator as

$$\hat{\mathbf{x}}_{k+1} = (A_k + B_kL_k)\hat{\mathbf{x}}_k + B_kl_k + K_kF_ke_k + K_k\mathcal{D}_k(\mathbf{x}_k,\pi_k)\eta_k. \tag{68}$$

With the definition of the unconditional means m_k^e, $m_k^{\hat{\mathbf{x}}}$, the unconditional covariances $\Sigma_k^{\hat{\mathbf{x}}}$, $\Sigma_k^{\hat{\mathbf{x}}e}$ and the additional definition

$$\mathcal{M}_k = E\big[\mathcal{C}_k(\mathbf{x}_k,\pi_k)\mathcal{C}_k^T(\mathbf{x}_k,\pi_k)\big], \tag{69}$$

given in Lemma 3, using the fact that $\Sigma_k^{e\hat{\mathbf{x}}} = (\Sigma_k^{\hat{\mathbf{x}}e})^T$ and $\Sigma_k^{\mathbf{x}} = E(\hat{\mathbf{x}}_k + e_k)(\hat{\mathbf{x}}_k + e_k)^T] = \Sigma_k^{\mathbf{x}} + \Sigma_k^{\hat{\mathbf{x}}e} + \Sigma_k^{e\hat{\mathbf{x}}} + \Sigma_k^e$ and equations (59) and (68), the updates for the unconditional means and covariances are

$$\begin{aligned}m_{k+1}^{\hat{\mathbf{x}}} &= (A_k + B_kL_k)m_k^{\hat{\mathbf{x}}} + K_kF_km_k^e + B_kl_k,\\ m_{k+1}^e &= (A_k - K_kF_k)m_k^e,\end{aligned}$$

$$\begin{aligned}\Sigma^{\hat{\mathbf{x}}}_{k+1} &= (A_k + B_k L_k)\Sigma^{\hat{\mathbf{x}}}_k (A_k + B_k L_k)^T + K_k F_k \Sigma^e_k F_k^T K_k^T \\ &+ K_k \mathcal{P}_k K_k^T + (A_k + B_k L_k)\Sigma^{\hat{\mathbf{x}}e}_k F_k^T K_k^T \\ &+ K_k F_k \Sigma^{e\hat{\mathbf{x}}}_k (A_k + B_k L_k)^T \\ &+ \left((A_k + B_k L_k) m^{\hat{\mathbf{x}}}_k + K_k F_k m^e_k\right) l_k^T B_k^T \\ &+ B_k l_k \left((A_k + B_k L_k) m^{\hat{\mathbf{x}}}_k + K_k F_k m^e_k\right)^T + B_k l_k l_k^T B_k^T,\end{aligned} \tag{70}$$

$$\Sigma^e_{k+1} = (A_k - K_k F_k)\Sigma^e_k (A_k - K_k F_k)^T + K_k \mathcal{P}_k K_k^T + \mathcal{M}_k, \tag{71}$$

$$\begin{aligned}\Sigma^{\hat{\mathbf{x}}e}_{k+1} &= (A_k + B_k L_k)\Sigma^{\hat{\mathbf{x}}e}_k (A_k - K_k F_k)^T \\ &+ B_k l_k (m^e_k)^T (A_k - K_k F_k)^T \\ &+ K_k F_k \Sigma^e_k (A_k - K_k F_k)^T - K_k \mathcal{P}_k K_k^T.\end{aligned} \tag{72}$$

By substituting K_k (67) into the above equations and combining the term $K_k F_k \Sigma^e_k F_k^T K_k^T + K_k \mathcal{P}_k K_k^T$, we can rewrite the update equations (70)–(72) into forms (53)–(55) which are exactly the same as those we obtain in Lemma 3.

Furthermore, based on the definition given in (64) and (69), and all the definitions of the unconditional means and unconditional covariances, $\mathcal{P}_k$ and $\mathcal{M}_k$ can be computed using (56)–(57) which completes the proof.

The complete iteration algorithm is as follows:

(1) Apply the current control law and obtain the nominal state-control trajectory $\bar{\mathbf{x}}^p_k$, $\bar{\mathbf{u}}^p_k$. Discretize system using time step Δt, linearize the dynamics and quadratize the cost, we then build a local LQG approximation around $\bar{\mathbf{x}}^p_k$, $\bar{\mathbf{u}}^p_k$.
(2) Based on the current control law, in a forward pass through time, compute an improved suboptimal filter (49) evaluated by minimizing the magnitude of its estimation errors.
(3) In a backward pass through time, compute an improved affine feedback control law (48) in the form $\mathbf{u}_k = l_k + L_k \hat{\mathbf{x}}_k$, and update its value function in the quadratic form (16).
(4) Apply the new control law forward in time and repeat iteration (1), the new open-loop controls $\tilde{\mathbf{u}}^p_k = \bar{\mathbf{u}}^p_k + \mathbf{u}_k$ and the corresponding average state trajectory are computed. Iterate the above steps until convergence.

In the next section we will test the above algorithm on the reaching movements for a 2-link 6-muscle arm model, which has non-linear dynamics, non-quadratic costs and multiplicative noise.

6. Application to reaching movements

6.1 *2-link 6-muscle human arm model*

Consider an arm model with 2 joints (shoulder and elbow), moving in the horizontal plane (figure 1). The inverse dynamics is

$$\mathcal{M}(\theta)\ddot{\theta} + \mathcal{C}(\theta, \dot{\theta}) + \mathcal{B}\dot{\theta} = \tau, \tag{73}$$

where $\theta \in R^2$ is the joint angle vector (shoulder: θ_1, elbow: θ_2), $\mathcal{M}(\theta) \in R^{2\times 2}$ is a positive definite symmetric inertia matrix, $\mathcal{C}(\theta, \dot{\theta}) \in R^2$ is a vector centripetal and Coriolis forces, $\mathcal{B} \in R^{2\times 2}$ is the joint friction matrix, and $\tau \in R^2$ is the joint torque. In (73), the expressions of the different variables and parameters are given by

$$\begin{aligned}\mathcal{M} &= \begin{pmatrix} a_1 + 2a_2\cos\theta_2 & a_3 + a_2\cos\theta_2 \\ a_3 + a_2\cos\theta_2 & a_3 \end{pmatrix}, \\ \mathcal{C} &= \begin{pmatrix} -\dot{\theta}_2(2\dot{\theta}_1 + \dot{\theta}_2) \\ \dot{\theta}_1^2 \end{pmatrix} a_2 \sin\theta_2, \quad \mathcal{B} = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix},\end{aligned} \tag{74}$$

$$a_1 = I_1 + I_2 + m_2 l_1^2, \quad a_2 = m_2 l_1 s_2, \quad a_3 = I_2, \tag{75}$$

where $b_{11} = b_{22} = 0.05$, $b_{12} = b_{21} = 0.025$, m_i is the mass (1.4 kg, 1 kg), l_i is the length of link i (30 cm, 33 cm), s_i is the distance from the joint centre to the centre of the mass for link i (11 cm, 16 cm), and I_i is the moment of inertia (0.025 kgm^2, 0.045 kgm^2). Based on equations (73)–(75), we can compute the forward dynamics

$$\ddot{\theta} = \mathcal{M}(\theta)^{-1}(\tau - \mathcal{C}(\theta, \dot{\theta}) - \mathcal{B}\dot{\theta}). \tag{76}$$

As we see in figure 1a, there are a large number of muscles that act on the arm in the horizontal plane. But since we have only 2 degrees of freedom, these muscles can be organized into 6 actuator groups: elbow flexors (1), elbow extensors (2), shoulder flexors (3), shoulder extensors (4), biarticulate flexors (5), and biarticulate extensors (6). The joint torques produced by a muscle are a function of its moment arms (figure 1b), length-velocity-tension curve (figure 1c), and activation dynamics (figure 1d), which is given by

$$\tau = M(\theta)T(a, l(\theta), v(\theta, \dot{\theta})). \tag{77}$$

The moment arm $M(\theta) \in R^{2\times 6}$ is defined as the perpendicular distance from the muscle's line of action to the joint's center of rotation. Moment arms are roughly constant for extensor muscles, but vary considerably with joint angle for flexor muscles. For each flexor group, this variation is modelled with a function of the form $a + b\cos(c\,\theta)$, where the constants have been adjusted to match experimental data. This function provides a good fit to data—not surprising, since moment arm variations are due to geometric

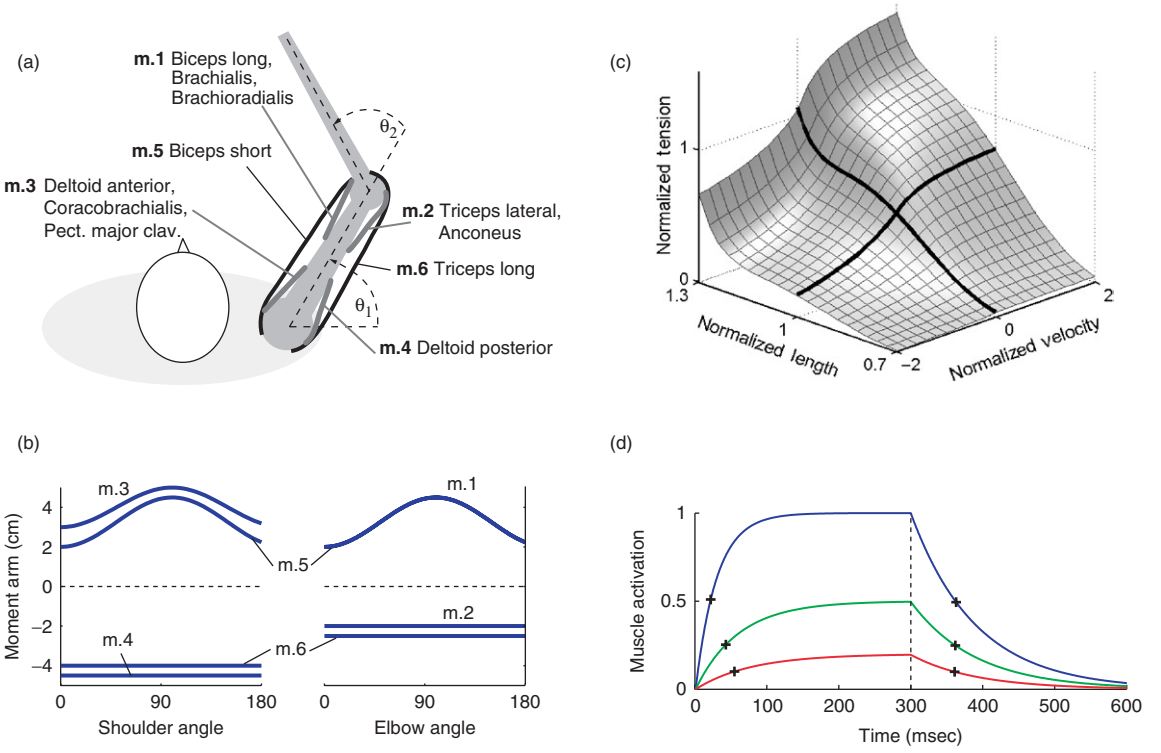

Figure 1. (a) 2-link 6-muscle arm; (b) joint torques; (c) length-velocity-tension curve; (d) muscle activation dynamics.

factors related to the cosine of the joint angle. It can also be integrated analytically, which is convenient since all muscle lengths need to be computed at each point in time.

The tension produced by a muscle obviously depends on the muscle activation a, but also varies substantially with the length l and velocity v of that muscle. Figure 1c, based on the publicly available virtual muscle model (Brown *et al.* 1999), illustrates that dependence for maximal activation. We will denote this function with $T(a, l, v)$.

$$T(a,l,v) = A(a,l)(F_L(l)V_V(l,v) + F_P(l))$$

$$A(a,l) = 1 - \exp\left(-\left(\frac{a}{0.56N_f(l)}\right)^{N_f(l)}\right)$$

$$N_f(l) = 2.11 + 4.16\left(\frac{1}{l} - 1\right)$$

$$F_L(l) = \exp\left(-\left|\frac{l^{1.93} - 1}{1.03}\right|^{1.87}\right)$$

$$F_V(l,v) = \begin{cases} \dfrac{-5.72 - v}{-5.72 + (1.38 + 2.09l)v}, & v \le 0 \\ \dfrac{0.62 - (-3.12 + 4.21l - 2.67l^2)v}{0.62 + v} \end{cases}$$

$$F_P(l) = -0.02\exp(13.8 - 18.7l).$$

Mammalian muscles are known to have remarkable scaling properties, meaning that they are all similar after proper normalization: length is expressed in units of L_0 (the length at which maximal isometric force is generated), and velocity is expressed in units of L_0/sec. The unitless tension in figure 1c is scaled by 31.8 N per square centimeter of physiological cross-sectional area (PCSA) to yield physical tension T. The PCSA parameters used in the model are the sums of the corresponding parameters for all muscles in each group (1: 18 cm^2; 2: 14 cm^2; 3: 22 cm^2; 4: 12 cm^2; 5: 5 cm^2; 6: 10 cm^2). Muscle length (and velocity) are converted into normalized units of L_0 using information about the operating range of each muscle group (1: 0.6 − 1.1; 2: 0.8 − 1.25; 3: 0.7 − 1.2; 4: 0.7 − 1.1; 5: 0.6 − 1.1; 6: 0.85 − 1.2).

Muscle activation a_i $(i = 1, \ldots, 6)$ is not equal to instantaneous neural input u_i, but is generated by passing u_i through a filter that describes calcium dynamics. This is reasonably well modelled with a first order non-linear filter of the form

$$\dot{a}_i = \frac{((1 + \sigma_u\varepsilon)u_i - a_i)}{t(u_i, a_i)}, \tag{78}$$

where

$$t(u_i, a_i) = \begin{cases} t_{\text{deact}} + u_i(t_{\text{act}} - t_{\text{deact}}) & u_i > a_i, \\ t_{\text{deact}} & \text{otherwise.} \end{cases}$$

The input-dependent activation dynamics $t_{act} = 50$ msec is faster than the constant deactivation dynamics $t_{deact} = 66$ msec. Figure 1d illustrates the response of this filter to step inputs that last 300 msec. Note that the half-rise times are input-dependent, while the half-fall times are constant (crosses in figure 1d). The neural inputs u_i is disturbed by the multiplicative noise, whose standard deviation is 20% of its magnitude—which means $\sigma_u = 0.2$ in (78), while ε is a zero-mean Gaussian white noise with unity covariance.

We notice that adding muscles to the dynamical system results in 6 new state variables. Combining the forward dynamics (76) and muscle actuator model (78), we could write the system into a state space form

$$\dot{x} = F(x) + G(x)(1 + \sigma_u \varepsilon)u, \tag{79}$$

where the state and control are given by $x = (\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2, a_1, \ldots, a_6)^T$ and $u = (u_1, \ldots, u_6)^T$ respectively.

The sensory feedback carries the information about position and velocity

$$y = (\theta_1 \quad \theta_2 \quad \dot{\theta}_1 \quad \dot{\theta}_2)^T + v. \tag{80}$$

where the sensory noise v has zero-mean Gaussian distribution with unity covariance.

The first task we study is reaching movement, where the arm has to start at some initial position and move to a target in a specified time interval. It also has to stop at the target, and do all that with minimal energy consumption. There are good reasons to believe that such costs are indeed relevant to the neural control of movement (Todorov and Jordan, 2002). The cost function is defined as

$$J_1 = \left\| e(\theta(T)) - e^* \right\|^2 + 0.001 \left\| \dot{e}\left(\theta(T), \theta(\dot{T})\right) \right\|^2 + \frac{1}{2} \int_0^T 0.0001 \|u\|^2 \, \mathrm{d}t, \tag{81}$$

where $e(\theta)$ is the forward kinematics transformation from joint coordinates to end-point coordinates, and the desired target e^* is defined in end-point coordinates.

6.2 *Numerical results*

In order to demonstrate the effectiveness of our design, we applied ILQG method to the human arm model described above. Note that this model is stochastic: we include multiplicative noise in the control signal, with standard deviation equal to 20% of the control signal. Here we use the center-out reaching task which is commonly studied in the motor control—the targets are arranged in a circle with 0.1 m radius around the starting position. Figure 2 shows average behavior for the fully observable case: hand paths in (a), tangential speed profiles in (b), and muscle activations in (c). We found out that both the movement kinematics and the muscle activations share many features with experimental data on human arm movements, but a detailed discussion of the relevance to motor control is beyond the scope of this paper. Another encouraging result is the CPU time. On average, the algorithm can find a locally-optimal time-varying feedback control law in about 10 seconds (on a 2.8 GHz Pentium 4 machine, in Matlab), for reaching in 16 different directions.

Figure 3 illustrates the robustness to noise: open-loop control in (a), closed-loop control in (b), and closed-loop control optimized for a deterministic system in (c). Closed-loop control is based on the time-varying feedback gain matrix L generated by the ILQG method, while open-loop control only uses the final u constructed by the algorithm. As the endpoint error ellipses show, the feedback control scheme substantially reduces the effects of the noise, and benefits from being optimized for the correct multiplicative noise model.

Now we look at the partial observable case where the states of system are obtained by the estimator. Although the state of the controlled plant are only available through noisy measurement, figure 4(a) shows that the hand could still arrive to the desired target position (shown as red stars in figure 4a) as accurately as possible. Figure 4(b) shows tangential speed profiles, for reaching in 16 different directions, and they remain in bell shapes. Figure 4(c) shows the muscle activation patterns for elbow flexor, elbow extensor, shoulder flexor, shoulder extensor, biarticulate flexor and biarticulate extensor.

Trajectory-based algorithms related to Pontryagins Maximum Principle in general find locally-optimal solutions, and complex control problems may exhibit many local minima. To explore the issue of local minima for the arm control problem, we used 50 different initializations, for each of 8 movement directions. The final trajectories are given in figure 5, where figure 5(a) shows that, for the fully observable case, all the optimization runs converged to a solution very similar to the best solution we found for the corresponding target direction. Figure 5(b) shows how the cloud of 50 randomly initialized trajectories gradually converge for the partial observable case by using ILQG method. There are local minima, but half the time the algorithm converges to the global minimum. Therefore, a small number of restarts of ILQG are sufficient to discover what appears to be the global minimum in a relatively complex control problem.

Finally we studied the convergence properties of the algorithm. The complete algorithm is that we initialize $K_0, \ldots, K_{N-1}$, and iterate (48) and (49)–(57) until convergence. Convergence is guaranteed, because the expected cost is non-negative by definition, and we are

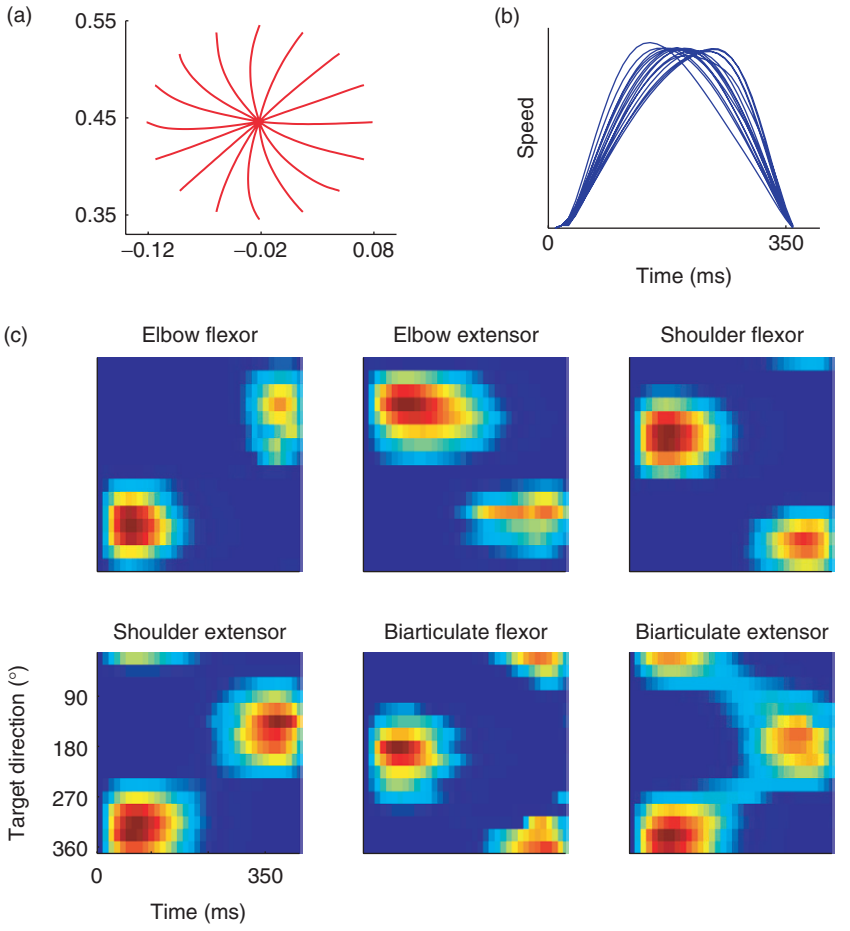

Figure 2. Fully observable case: average behaviour of the ILQG controller for reaching movements, using a 2-link 6-muscle human arm model: (a) hand paths for movement in 16 directions; (b) speed profiles; (c) muscle activations.

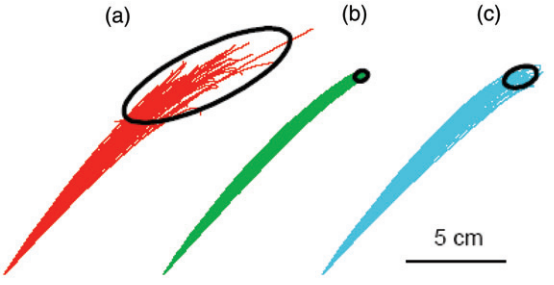

Figure 3. Effects of control-dependent noise on hand reaching trajectories, under different control laws: (a) open-loop control; (b) closed-loop control; (c) closed-loop controller optimized for deterministic system.

using a coordinate-descent algorithm, which decreases the expected cost in each step. Figure 6 shows how the cost decreases with the number of iterations, obtained by averaging 50 random conditions over each of 8 movement directions. In all cases convergence is very rapid, with the relative change in expected cost decreasing exponentially.

7. Conclusion

Optimal control theory plays a very important role in the study of biological movement. Further progress in the field depends on the availability of efficient methods for solving non-linear optimal control problems. For the real control system design, feedback is based on delayed and noisy sensors that may not measure all the state variables, hence we extend the algorithm to the partially observable case by combining it with an extended Kalman filter. This results in a coupled estimation-control problem, which is complicated in the presence of multiplicative noise. This paper developed a new iterative local method for optimal feedback control and estimation design of non-linear stochastic dynamical systems. It provided an iterative coordinate-descent algorithm, which is guaranteed to converge to a filter and a control law optimal with respect to each other. We illustrated its application to a biomechanical model of the human arm. The simulation results numerically demonstrated that the solutions were close to the global minimum.

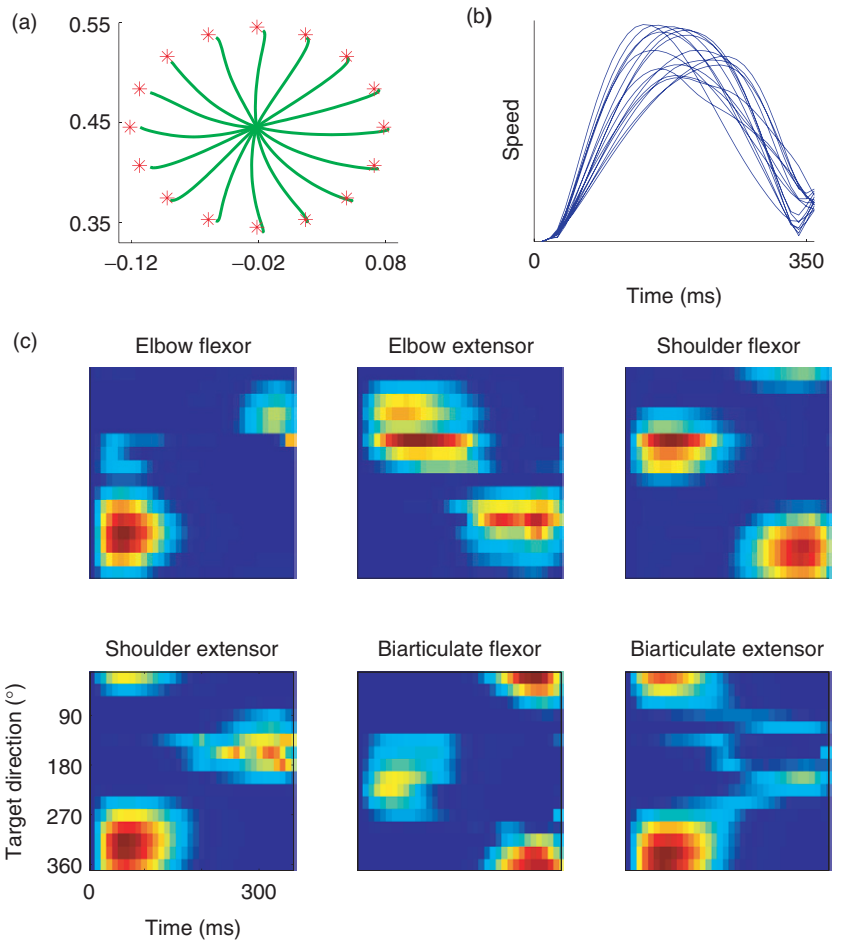

Figure 4. Partial observable case: average behavior of the ILQG controller and estimator for reaching movements, using a 2-link 6-muscle human arm model: (a) Hand paths for movement in 16 directions; (b) Speed profiles; (c) Muscle activations.

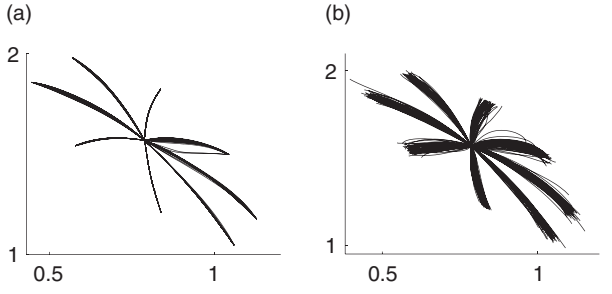

Figure 5. The optimized hand paths obtained by using 50 different initial conditions for each of 8 movement directions. (a) fully observable case; (b) partial observable case.

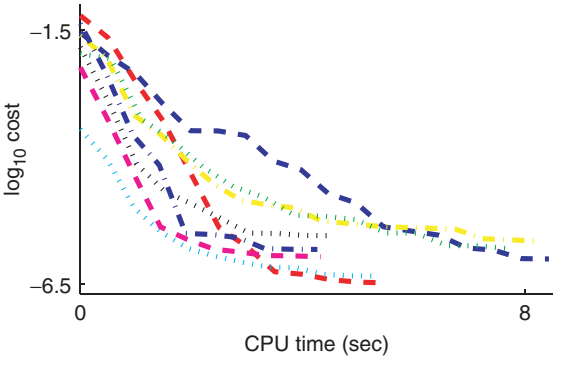

Figure 6. Cost over iterations for each of 8 movement directions.

Finally there are several extensions to the work we presented here. While we assumed a specified final time T, the algorithm can be applied in model-predictive mode, using a fixed time horizon rather than a fixed final time. The final cost $h(x)$ will have to be replaced with some approximation to the optimal cost-to-go, but that has to be done whenever fixed-horizon model-predictive control is used. Additional work is needed to ascertain the properties of the algorithm in more complex problems, where we cannot use global methods for validation. We are going to implement a very detailed model of the human arm, including 4 degrees of freedom and around 20 muscles; it will be interesting to see if the algorithm can handle such a complex system in a reasonable amount of time.

Acknowledgments

This work is supported by the National Institutes of Health Grant R01-NS045915.

References

M. Abu-Khalaf and F.L. Lewis, "Nearly optimal control laws for non-linear systems with saturating actuators using a neural network HJB approach", *Automatica*, 41, pp. 779–791, 2005.

A. Beghi and D. D'Alessandro, "Discrete-time optimal control with control-dependent noise and generalized Riccati difference equations", *Automatica*, 34, pp. 1031–1034, 1998.

D.P. Bertsekas, *Dynamic Programming and Optimal Control*, Belmont, Massachusetts, USA: Athena Scientific, 2000.

R.R. Bitmead, "Iterative control design approaches", *IFAC World Congress*, Sydney, Invited plenary paper, 9, pp. 381–384, 1993.

R.R. Bitmead, Iterative Optimal Control, in *Lecture Notes on Iterative Identification and Control Design*, P. Albertos and A. Sala (Eds), Strasbourg, France: European Science Foundation, 2000, pp. 153–166.

I.E. Brown, E.J. Cheng and G.E. Leob, "Measured and modeled properties of mammalian skeletal muscle. II. The effects of stimulus frequency on force-length and force-velocity relationships", *J. Muscle Res. Cell Motil.*, 20, pp. 627–643, 1999.

A.E. Bryson and Y.-C. Ho, *Applied Optimal Control*, Massachusetts, USA: Blaisdell Publishing Company, 1969.

B.S. Chow and W.P. Birkemeier, "A new recursive filter for systems with multiplicative noise", *IEEE Trans. Informat. Theory*, 36, pp. 1430–1435, 1990.

M.C. De Oliveria and R.E. Skelton, "State feedback control of linear systems in the presence of devices with finite signal-to-noise ratio", *Int. J. Cont.*, 74, pp. 1501–1509, 2001.

L. El Ghaoui, "State-feedback control of systems of multiplicative noise via linear matrix inequalities", *Syst. Con. Lett.*, 24, pp. 223–228, 1995.

E. Gershon, U. Shaked and I. Yaesh, "H_∞ control and filtering of discrete-time stochastic systems with multiplicative noise", *Automatica*, 37, pp. 409–417, 2001.

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

B. Hoff, "A computational description of the organization of human reaching and prehension". PhD thesis, University of Southern California, 1992.

A. Isidori, *Non-linear Control Systems*, New York, USA: Springer, 1995.

J.C. Jimenez and T. Ozaki, "Linear estimation of continuous-discrete linear state space models with multiplicative noise", *Syst. Con. Lett.*, 47, pp. 91–101, 2002.

H. Kushner and P. Dupuis, *Numerical Methods For Stochastic Control Problems in Continuous Time,* , 2nd ed., New York, USA: Spinger, 2001.

W. Li and E. Todorov, "Iterative linear quadratic regulator design for non-linear biological movement systems". in *Proceedings of the 1st International Conference on Informatics in Control, Automation and Robotics*, Portugal, 1, 2004, pp. 222–229.

P. McLane, "Optimal stochastic control of linear systems with state- and control-dependent disturbances", *IEEE Trans. Automat. Cont.*, 16, pp. 793–798, 1971.

J.B. Moore, X.Y. Zhou and E.B. Lim, "Discrete time LQG controls with control dependent noise", *Syst. Con. Lett.*, 36, pp. 199–206, 1999.

C.K. Ng, L.Z. Liao and D. Li, "A globally convergent and efficient method for unconstrained discrete-time optimal control", *J. Global Optim.*, 23, pp. 401–421, 2002.

Y.A. Phillis, "Controller design of systems with multiplicative noise", *IEEE Trans. Autom. Cont.*, 30, pp. 1017–1019, 1985.

Y.A. Phillis, "Estimation and control of systems with unknown covariance and multiplicative noise", *IEEE Trans. Autom. Cont.*, 34, pp. 1075–1078, 1989.

M.A. Rami, X. Chen, J.B. Moore and X. Zhou, "Solvability and asymptotic behavior of generalized Riccati equations arising in indefinite stochastic LQ controls", *IEEE Trans. Autom. Cont.*, 46, pp. 428–440, 2001.

E. Todorov and M. Jordan, "Optimal feedback control as a theory of motor coordination", *Nature Neuroscience*, 5, pp. 1226–1235, 2002.

E. Todorov and W. Li, "Optimal control methods suitable for biomechanical systems", in *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Cancun, Mexico, 2003, pp. 1758–1761.

E. Todorov, "Optimality principles in sensorimotor control", *Nature Neuroscience*, 7, pp. 907–915, 2004.

E. Todorov, "Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system", *Neural Computation*, 17, pp. 1084–1108, 2005.

Y. Uno, M. Kawato and R. Suzuki, "Formation and control of optimal trajectory in human multijoint arm movement: Minimum torque-change model", *Biol. Cybern.*, 61, pp. 89–101, 1989.

J.L. Willems and J.C. Willems, "Feedback stabilizability for stochastic systems with state and control dependent noise", *Automatica*, 12, pp. 277–283, 1976.