Iterative Linear Quadratic Regulator Design for Nonlinear Biological Movement Systems

Weiwei Li
Department of Mechanical and Aerospace Engineering
University of California San Diego
La Jolla, CA 92093-0411
wwli@mechanics.ucsd.edu

Emanuel Todorov
Department of Cognitive Science
University of California San Diego
La Jolla, CA 92093-0515
todorov@cogsci.ucsd.edu

Abstract

This paper presents an Iterative Linear Quadratic Regulator (ILQR) method for locally-optimal feedback control of nonlinear dynamical systems. The method is applied to a musculo-skeletal arm model with 10 state dimensions and 6 controls, and is used to compute energy-optimal reaching movements. Numerical comparisons with three existing methods demonstrate that the new method converges substantially faster and finds slightly better solutions.

Keywords: ILQR, Optimal control, Nonlinear system

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. It has also provided the most fruitful general framework for constructing models of biological movement[3, 8, 11]. 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 improve behavioral performance.

Producing even the simplest movement involves an enormous amount of information processing. When we move our hand to a target, there are infinitely many possible paths and velocity profiles that the multi-joint arm could take, and furthermore each trajectory can be generated by an infinite variety of muscle activation patterns (since we have many more muscles than joints). Biomechanical redundancy makes the motor system very flexible, but at the same time requires a very well designed controller that can choose intelligently among the many possible alternatives. Optimal control theory provides a principled approach to this problem – it postulates that the movement patterns being chosen are the ones optimal for the task at hand.

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[8].

In their present form, however, such models have a serious limitation – they rely on the Linear-Quadratic-Gaussian formalism, while in reality biomechanical systems are strongly non-linear. The goal of the present paper is to develop a new method, and compare its performance to existing methods[9] on a challenging biomechanical control problem. The new method uses iterative linearization of the nonlinear system around a nominal trajectory, and computes a locally optimal feedback control law via a modified LQR technique. This control law is then applied to the linearized system, and the result is used to improve the nominal trajectory incrementally.

The paper is organized as follows. The new ILQR method is derived in Section 2. In section 3 we present a realistic biomechanical model of the human arm moving in the horizontal plane, as well as two simpler dynamical systems used for numerical comparisons. Results of applying the new method are presented in section 4. Finally, section 5 compares our method to three existing methods, and demonstrates a superior rate of convergence.

The notation used here is fairly standard. The transpose of a real matrix A is denoted by A^T; for a symmetric matrix, the standard notation > 0 (≥ 0) is used to denote positive definite matrix (positive semi-definite matrix). $D_x f(\cdot)$ denotes the Jacobian of $f(\cdot)$ with respect to x.

2 ILQR approach to nonlinear systems

Consider a discrete time nonlinear dynamical system with state variable $x_k \in R^{n_x}$ and control $u_k \in R^{n_u}$

$$x_{k+1} = f(x_k, u_k). \tag{1}$$

The cost function is written in the quadratic form

$$J_0 = \frac{1}{2}(x_N - x^*)^T Q_f (x_N - x^*) + \frac{1}{2}\sum_{k=0}^{N-1}(x_k^T Q x_k + u_k^T R u_k), \tag{2}$$

where x_N describes the final state (each movement lasts N steps), x^* is the given target. The state cost-weighting matrices Q and Q_f are symmetric positive semi-definite, the control cost-weighting matrix R is positive definite. All these matrices are assumed to have proper dimensions. Note that when the true cost is not quadratic, we can still use a quadratic approximation to it around a nominal trajectory.

Our algorithm is iterative. Each iteration starts with a nominal control sequence u_k, and a corresponding nominal trajectory x_k obtained by applying u_k to the dynamical system in open loop. The initialization is $u_k = 0$. The iteration produces an improved sequence u_k, by linearizing the system dynamics around u_k, x_k and solving a modified LQR problem. The process is repeated until convergence. Let the deviations from the nominal u_k, x_k be $\delta u_k, \delta x_k$. The linearization is

$$\delta x_{k+1} = A_k \delta x_k + B_k \delta u_k, \tag{3}$$

where $A_k = D_x f(x_k, u_k), B_k = D_u f(x_k, u_k)$. D_x denotes the Jacobian of $f(\cdot)$ with respect to x, D_u denotes the Jacobian of $f(\cdot)$ with respect to u, and the Jacobians are evaluated along x_k and u_k.

Based on the linearized model (3), we can solve the following LQR problem with the cost function

$$\begin{aligned} J &= \frac{1}{2}(x_N + \delta x_N - x^*)^T Q_f (x_N + \delta x_N - x^*) \\ &\quad + \frac{1}{2}\sum_{k=0}^{N-1}\{(x_k + \delta x_k)^T Q (x_k + \delta x_k) \\ &\quad + (u_k + \delta u_k)^T R (u_k + \delta u_k)\}. \end{aligned} \tag{4}$$

We begin with the Hamiltonian function

$$\begin{aligned} H_k &= \frac{1}{2}(x_k + \delta x_k)^T Q (x_k + \delta x_k) \\ &\quad + \frac{1}{2}(u_k + \delta u_k)^T R (u_k + \delta u_k) \\ &\quad + \delta\lambda_{k+1}^T (A_k \delta x_k + B_k \delta u_k), \end{aligned} \tag{5}$$

where $\delta\lambda_{k+1}$ is Lagrange multiplier.

The optimal control improvement δu_k is given by solving the state equation (3), the costate equation

$$\delta\lambda_k = A_k^T \delta\lambda_{k+1} + Q(\delta x_k + x_k), \tag{6}$$

and the stationary condition which can be obtained by setting the derivative of Hamiltonian function with respect to δu_k to zero

$$0 = R(u_k + \delta u_k) + B_k^T \delta\lambda_{k+1} \tag{7}$$

with the boundary condition

$$\delta\lambda_N = Q_f (x_N + \delta x_N - x^*). \tag{8}$$

Solving for (7) yields

$$\delta u_k = -R^{-1} B_k^T \delta\lambda_{k+1} - u_k. \tag{9}$$

Hence, substituting (9) into (3) and combining it with (6), the resulting Hamiltonian system is

$$\begin{pmatrix} \delta x_{k+1} \\ \delta\lambda_k \end{pmatrix} = \begin{pmatrix} A_k & -B_k R^{-1} B_k^T \\ Q & A_k^T \end{pmatrix} \begin{pmatrix} \delta x_k \\ \delta\lambda_{k+1} \end{pmatrix} + \begin{pmatrix} -B_k u_k \\ Q x_k \end{pmatrix}. \tag{10}$$

It is clear that the Hamiltonian system is not homogeneous, but is driven by a forcing term dependent on the current trajectory x_k and u_k. Because of the forcing term, it is not possible to express the optimal control law in linear state feedback form (as in the classic LQR case). However, we can express δu_k as a combination of a linear state feedback plus additional terms, which depend on the forcing function.

Based on the boundary condition (8), we assume

$$\delta\lambda_k = S_k \delta x_k + v_k \tag{11}$$

for some unknown sequences S_k and v_k. Substituting the above assumption into the state and costate equation, and applying the matrix inversion lemma yields the optimal controller

$$\begin{aligned} \delta u_k &= -K\delta x_k - K_v v_{k+1} - K_u u_k, && (12) \\ K &= (B_k^T S_{k+1} B_k + R)^{-1} B_k^T S_{k+1} A_k, && (13) \\ K_v &= (B_k^T S_{k+1} B_k + R)^{-1} B_k^T, && (14) \\ K_u &= (B_k^T S_{k+1} B_k + R)^{-1} R, && (15) \\ S_k &= A_k^T S_{k+1}(A_k - B_k K) + Q, && (16) \\ v_k &= (A_k - B_k K)^T v_{k+1} - K^T R u_k + Q x_k, && (17) \end{aligned}$$

with boundary conditions

$$S_N = Q_f, \qquad v_N = Q_f (x_N - x^*). \tag{18}$$

In order to find the equations (12)-(18), use (11) in the state equation (3) to yield

$$\begin{aligned} \delta x_{k+1} = (I + B_k R^{-1} B_k^T S_{k+1})^{-1}(A_k \delta x_k \\ - B_k R^{-1} B_k^T v_{k+1} - B_k u_k). \end{aligned} \tag{19}$$

Substituting (11) and the above equation into the costate equation (6) gives

$$S_k \delta x_k + v_k = Q\delta x_k + A_k^T S_{k+1}(I + B_k R^{-1} B_k^T S_{k+1})^{-1} (A_k \delta x_k - B_k R^{-1} B_k^T v_{k+1} - B_k u_k) + A_k^T v_{k+1} + Q x_k.$$

By applying the matrix inversion lemma,[1] we obtain

$$S_k = A_k^T S_{k+1}[I - B_k(B_k^T S_{k+1} B_k + R)^{-1} B_k^T S_{k+1}]A_k + Q,$$

and

$$\begin{aligned} v_k =& A_k^T v_{k+1} - A_k^T S_{k+1}[I - B_k(B_k^T S_{k+1} B_k + R)^{-1} \\ & B_k^T S_{k+1}] B_k R^{-1} B_k^T v_{k+1} - A_k^T S_{k+1}[I - \\ & B_k(B_k^T S_{k+1} B_k + R)^{-1} B_k^T S_{k+1}] B_k u_k + Q x_k. \end{aligned}$$

By using $(R + B_k^T S_{k+1} B_k)^{-1} = R^{-1} - (R + B_k^T S_{k+1} B_k)^{-1} B_k^T S_{k+1} B_k R^{-1}$, the second term in v_k becomes

$$-A_k^T S_{k+1} B_k (R + B_k^T S_{k+1} B_k)^{-1} B_k^T v_{k+1},$$

and the third term in v_k can be written as

$$-A_k^T S_{k+1} B_k (R + B_k^T S_{k+1} B_k)^{-1} R u_k.$$

Therefore, with the definition of K in (13), the above S_k, v_k can be written into the forms as given in (16) and (17).

Furthermore, substituting (11) and (19) into (9) yields

$$\begin{aligned} \delta u_k = & -(R + B_k^T S_{k+1} B_k)^{-1} B_k^T S_{k+1} A_k \delta x_k \\ & - (R + B_k^T S_{k+1} B_k)^{-1} B_k^T v_{k+1} \\ & - (R + B_k^T S_{k+1} B_k)^{-1} R u_k. \end{aligned}$$

By the definition of K, K_v and K_u in (13)-(15), we can rewrite above δu_k as the form in (12).

With the boundary condition S_N given as the final state weighting matrix in the cost function (4), we can solve for an entire sequence of S_k by the backward recursion (16). It is interesting to note that the control law δu_k consists of three terms: a term linear in δx_k whose gain is dependent on the solution to the Riccati equation; a second term dependent on an auxiliary sequence v_k which is derived from the auxiliary difference equation (17); and a third term dependent on the nominal control u_k whose gain also relied on the Riccati equation solution. Once the modified LQR problem is solved, an improved nominal control sequence can be found: $u_k^* = u_k + \delta u_k$.

3 Optimal control problems to be studied

We first describe the dynamics of a 2-link arm moving in the horizontal plane. We then add realistic muscle actuators to it. We also define an inverted pendulum problem often used for numerical comparisons.

[1] $(A + BCD)^{-1} = A^{-1} - A^{-1}B(DA^{-1}B + C^{-1})^{-1}DA^{-1}$.

3.1 The dynamics of a 2-link arm

Consider an arm model [1, 10], with 2 joints (shoulder and elbow), moving in the horizontal plane (Fig 1). The inverse dynamics is

$$\mathcal{M}(\theta)\ddot{\theta} + \mathcal{C}(\theta, \dot{\theta}) + \mathcal{B}\dot{\theta} = \tau, \tag{20}$$

where $\theta \in R^2$ is the joint angle vector (shoulder: θ_1, elbow: θ_2), $\mathcal{M}(\theta) \in R^{2\times2}$ is a positive definite symmetric inertia matrix, $\mathcal{C}(\theta, \dot{\theta}) \in R^2$ is a vector centripetal and Coriolis forces, $\mathcal{B} \in R^{2\times2}$ is the joint friction matrix, and $\tau \in R^2$ is the joint torque. Here we consider direct torque control (i.e. τ is the control signal) which will later be replaced with muscle control. In (20), the expressions of the different variables and parameters are given by

$$\begin{aligned} \mathcal{M} &= \begin{pmatrix} a_1 + 2a_2 cos\theta_2 & a_3 + a_2 cos\theta_2 \\ a_3 + a_2 cos\theta_2 & a_3 \end{pmatrix}, && (21) \\ \mathcal{C} &= \begin{pmatrix} -\dot{\theta}_2(2\dot{\theta}_1 + \dot{\theta}_2) \\ \dot{\theta}_1^{\,2} \end{pmatrix} a_2 sin\theta_2, && (22) \\ \mathcal{B} &= \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}, && (23) \\ a_1 &= I_1 + I_2 + m_2 l_1^2, && (24) \\ a_2 &= m_2 l_1 s_2, && (25) \\ a_3 &= I_2, && (26) \end{aligned}$$

where $b_{11} = b_{22} = 0.05, b_{12} = b_{21} = 0.025$, m_i is the mass (1.4kg, 1kg), l_i is the length of link i (30cm, 33cm), s_i is the distance from the joint center to the center of the mass for link i (11cm, 16cm), and I_i is the moment of inertia $(0.025kgm^2, 0.045kgm^2)$.

Based on equations (20)-(26), we can compute the forward dynamics

$$\ddot{\theta} = \mathcal{M}(\theta)^{-1}(\tau - \mathcal{C}(\theta, \dot{\theta}) - \mathcal{B}\dot{\theta}), \tag{27}$$

and write the system in state space form

$$\dot{x} = F(x) + G(x)u, \tag{28}$$

where the state and control are given by

$$x = (\theta_1\ \theta_2\ \dot{\theta}_1\ \dot{\theta}_2)^T, \qquad u = \tau = (\tau_1\ \tau_2)^T.$$

The cost function is

$$J_0 = \frac{1}{2}(\theta(T) - \theta^*)^T(\theta(T) - \theta^*) + \frac{1}{2}\int_0^T \tau^T R \tau\, dt, \tag{29}$$

where $R = \begin{pmatrix} 0.0001 & 0 \\ 0 & 0.0001 \end{pmatrix} > 0$ and θ^* is the desired final posture. In the definition of the cost function, the first term means that the joint angle is going to the target θ^* which represents the reaching movement; the second term illustrates the energy efficiency.

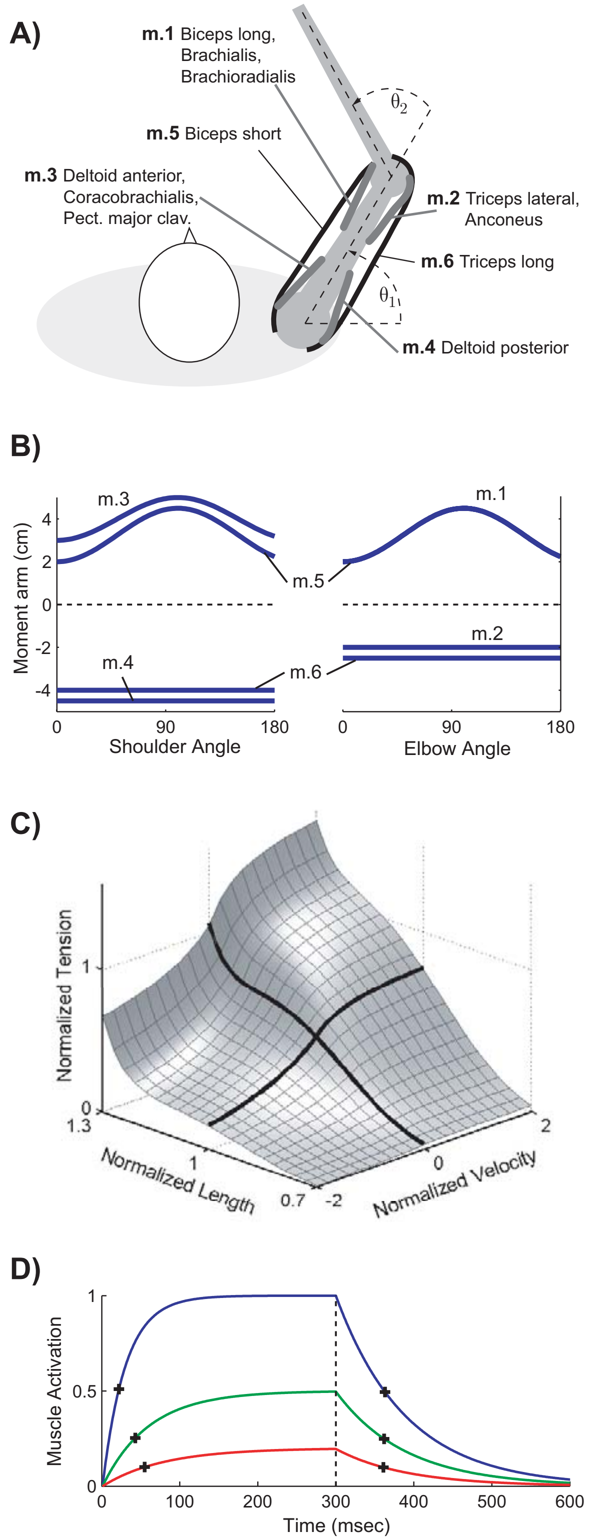

Figure 1: (A) 2-link 6-muscle arm; (B) Joint torques; (C) Length-velocity-tension curve; (D) Muscle activation dynamics.

3.2 A model of muscle actuators

There are a large number of muscles that act on the arm in the horizontal plane (see Fig 1A). 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 (Fig 1B), length-velocity-tension curve (Fig 1C), and activation dynamics (Fig 1D).

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 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. We will denote the 2 by 6 matrix of muscle moment arms with $M(\theta)$.

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. Fig 1C, based on the publicly available Virtual Muscle model, illustrates that dependence for maximal activation. We will denote this function with $T_0(a, l, v)$.

$$T_0\left(a, l, v\right) = A\left(a, l\right)\left(F_{\mathrm{L}}\left(l\right) F_{\mathrm{V}}\left(l, v\right) + F_{\mathrm{P}}\left(l\right)\right)$$

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

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

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

$$F_{\mathrm{V}}\left(l, v\right) = \begin{cases} \dfrac{-5.72 - v}{-5.72 + \left(1.38 + 2.09\ l\right)\ v}, & v \leq 0 \\ \dfrac{0.62 - \left(-3.12 + 4.21\ l - 2.67\ l^2\right)\ v}{0.62 + v}, & v > 0 \end{cases}$$

$$F_{\mathrm{P}}\left(l\right) = -0.02 \exp\left(13.8 - 18.7\ l\right)$$

Mammalian muscles are known to have remarkable scaling properties, meaning that they are all very 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 Fig 1C is scaled by $31.8N$ 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: $18cm^2$; 2: $14cm^2$; 3: $22cm^2$; 4: $12cm^2$; 5: $5cm^2$;

6: $10cm^2$). 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 is not equal to instantaneous neural input u, but is generated by passing u through a filter that describes calcium dynamics. This is reasonably well modelled with a first order nonlinear filter of the form $\dot{a} = (u-a)/t(u,a)$, where $t = t_{deact} + u(t_{act} - t_{deact})$ when $u > a$, and $t = t_{deact}$ otherwise. The input-dependent activation dynamics $t_{act} = 30msec$ is faster than the constant deactivation dynamics $t_{deact} = 60msec$. Fig 1D illustrates the response of this filter to step inputs that last $300msec$. Note that the half-rise times are input-dependent, while the half-fall times are constant (crosses in Fig 1D).

To summarize, adding muscles to the dynamical system results in 6 new state variables, with dynamics

$$\dot{a} = (u-a)/t(u,a). \tag{30}$$

The joint torque vector generated by the muscles is given by

$$\tau = M(\theta)\, T(a, l(\theta), v(\theta, \dot{\theta})). \tag{31}$$

The cost function is defined as

$$J_0 = \frac{1}{2}(\theta(T)-\theta^*)^T(\theta(T)-\theta^*) + \frac{1}{2}\int_0^T ru^T u\, dt, \tag{32}$$

where $r = 0.001$ and θ^* is the desired final posture.

3.3 Inverted Pendulum

Consider a simple pendulum where m denotes the mass of the bob, l denotes the length of the rod, θ describes the angle subtended by the vertical axis and the rod, and μ is the friction coefficient. For this example, we assume that $m = l = 1, g = 9.8, \mu = 0.01$.The state space equation of the pendulum is

$$\dot{x}_1 = x_2, \tag{33}$$

$$\dot{x}_2 = \frac{g}{l} sin x_1 - \frac{\mu}{ml^2}x_2 + \frac{1}{ml^2}u, \tag{34}$$

where the state variables are $x_1 = \theta$, $x_2 = \dot{\theta}$. The goal is to make the pendulum swing up. The control objective is to find the control $u(t)$, $0 < t < T$ and minimize the performance index

$$J_0 = \frac{1}{2}(x_1(T)^2 + x_2(T)^2) + \frac{1}{2}\int_0^T ru^2 dt, \tag{35}$$

where $r = 1e-5$.

4 Optimal trajectories

Here we illustrate the optimal trajectories found by iterating equations (12)-(18) on each of the three control problems. Fig 2A and Fig 2B show the optimal trajectory of the arm joint angles θ_1 (shoulder angle) and θ_2 (elbow angle). We find that the shoulder angle and the elbow angle arrive to the desired posture $\theta_1 = 1, \theta_2 = 1.5$ respectively. Fig 2C shows a set of optimal trajectories in the phase space, for a pendulum being driven from different starting points to the goal point. For example, S2 describes a starting point where the pendulum is hanging straight down; trajectory 2 shows that the pendulum swing directly up to the goal.

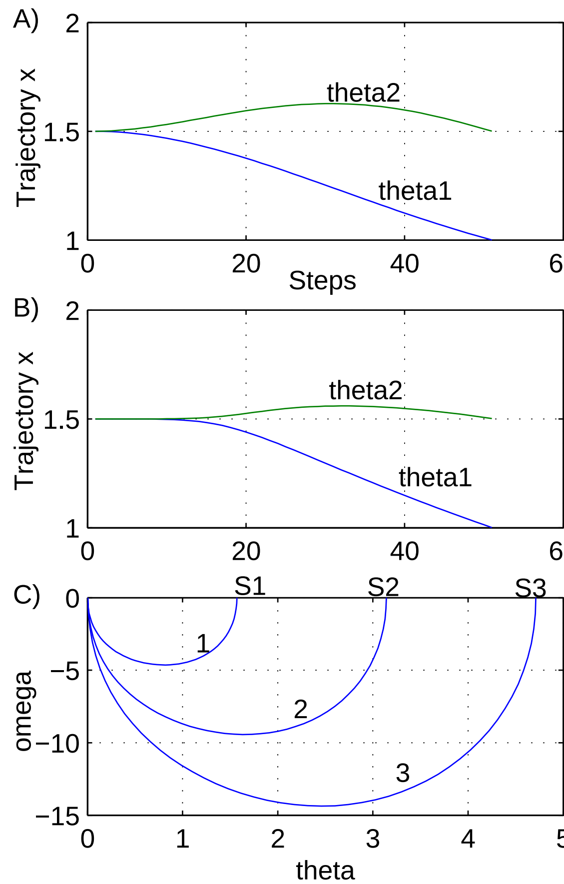

Figure 2: Optimal trajectories. (A) Torque-controlled arm; (B) Muscle-controlled arm; (C) Inverted pendulum.

Especially for the muscle-controlled arm model, Fig3 illustrates how the current trajectory converges with the number of iterations. After 11 iterations, the shoulder angle θ_1 and the elbow angle θ_2 arrive to the desired posture.

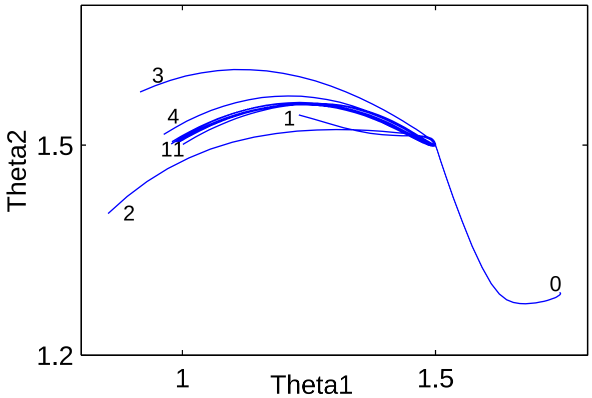

Figure 3: Trajectories of 2-link 6-muscle arm for different iterations

5 Comparison with existing algorithms

Existing algorithms for nonlinear optimal control can be classified in two groups, based respectively on Bellman's Optimality Principle and Pontryagin's Maximum Principle.

The former yields globally optimal solutions, but involves a partial differential equation (the Hamilton-Jacobi-Bellman equation) which is only solvable for low-dimensional systems. While various forms of function approximation have been explored, presently there is no known cure for the curse of dimensionality. Since the biological control problems we are interested in tend to have very high dimensionality (the 10 dim arm model is a relatively simple one), we do not believe that global methods will be applicable to such problems in the near future.

Therefore we have chosen to pursue local trajectory-based methods related to the Maximum Principle. These methods iteratively improve their estimate of the extremal trajectory. Elsewhere [9] has compared three such methods: (1) ODE solves the system of state-costate ordinary differential equations resulting from the Maximum Principle, using the BVP4C boundary value problem solver in Matlab; (2) CGD is a gradient descent method, which uses the Maximum Principle to compute the gradient of the total cost with respect to the nominal control sequence, and then calls an optimized conjugate gradient descent routine; (3) differential dynamic programming (DDP) performs dynamic programming in the neighborhood of the the nominal trajectory, using second order approximations. See [9] for more detailed descriptions of these preexisting algorithms.

All algorithms were implemented in Matlab, and used the same dynamic simulation. Table 1 compares the CPU time, number of iterations, and final cost for all algorithms on all three problems. Arm1 is the torque-controlled arm, Arm2 is the muscle-controlled arm. Note that the time per iteration varies substantially (and in the case of ODE the number of iterations is not even defined) so the appropriate measure of efficiency is the CPU time.

Table 1: Comparison of Four Methods

	Method	Time (sec)	Iteration	Cost
Arm1	ODE	3.19	N/A	1.14e-04
	CGD	4.46	7	1.26e-04
	DDP	1.30	10	1.73e-04
	ILQR	**0.61**	**6**	**0.96e-04**
Arm2	ODE	>400	N/A	N/A
	CGD	70.09	10	5.16e-05
	DDP	184.84	15	8.35e-05
	ILQR	**6.62**	**10**	**2.37e-05**
Pendulum	ODE	1.04	N/A	1.0628
	CGD	4.18	8	0.0701
	DDP	2.28	30	2.6611
	ILQR	**0.48**	**5**	**0.0096**

On all problems studied, the new ILQR method converged faster than the three existing methods, and found a better solution. The speed advantage is most notable in the complex arm model, where ILQR outperformed the nearest competitor by more than a factor of 10.

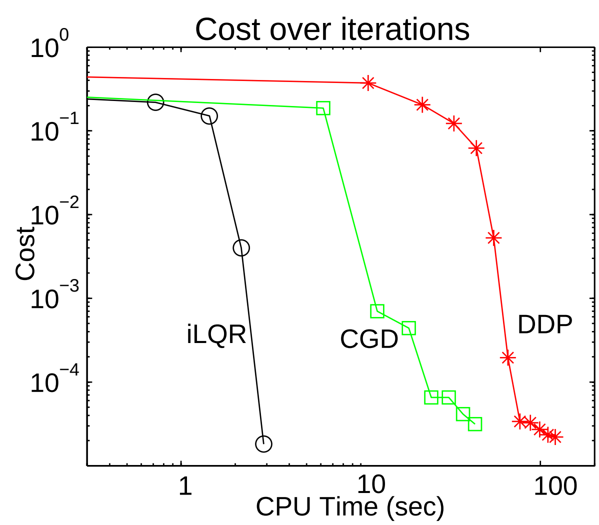

Figure 4: Cost vs. Iterations for 2-link 6-muscle Model

Fig 4 illustrates how the cost of the nominal trajectory decreases with the number of iterations. The new ILQR method converged faster than the other methods, and found a better solution. Also we have found

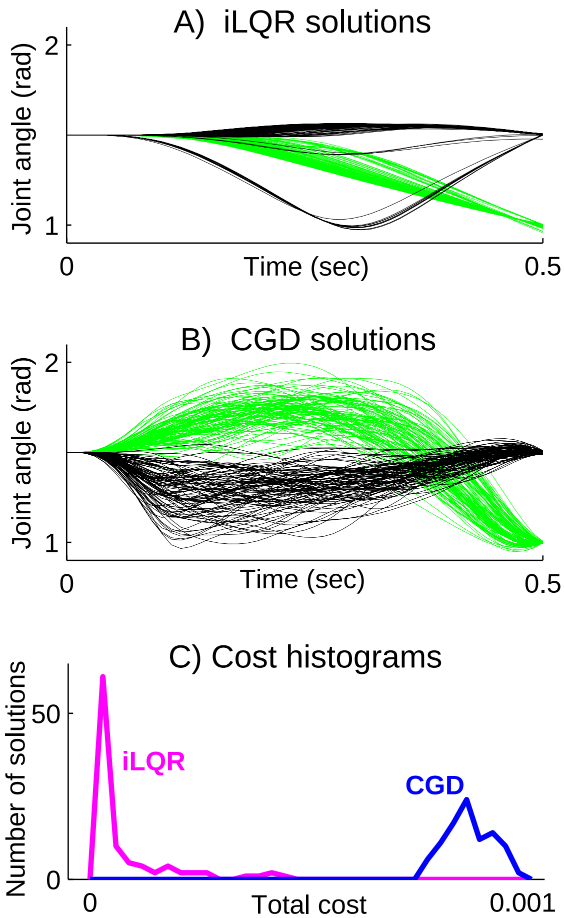

Figure 5: Trajectories of 2-link 6-muscle Model for different initial conditions (Light color describes the trajectories of the shoulder angle, black color describes the trajectories of the elbow angle)

that the amount of computation per iteration for ILQR method is much less than the other methods. This is because gradient descent requires a linesearch (without which it works poorly) and DDP uses a second-order approximation to the system dynamics – both of which take a significant amount of time to compute.

Fig 5A) and Fig 5B) shows how the cloud of 100 randomly initialized trajectories gradually converge for the muscle-controlled arm model by using ILQR and CGD method respectively.

6 Conclusions and future work

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 nonlinear optimal control problems. This paper developed a new Iterative Linear Quadratic Regulator (ILQR) algorithm for optimal feedback control of nonlinear dynamical systems. We illustrated its application to a realistic 2-link, 6-muscle arm model, as well as simpler control problems. The simulation results suggest that the new method is more effective compared to the three most efficient methods that we are aware of.

While the control problems studied here are deterministic, the variability of biological movements indicates the presence of large disturbances in the motor system. It is very important to take these disturbances into account when studying biological control. In particular, it is known that the motor noise is control-dependent, with standard deviation proportional to the mean of the control signal. Such noise has been modelled in the LQG setting before [8]. Since the present ILQR algorithm is an extension to the LQG setting, it should be possible to treat nonlinear systems with control-dependent noise using similar methods. Another issue that will need to be addressed is the presence of local minima. Trajectory-based algorithms related to Pontryagin's Maximum Principle in general find locally-optimal solutions, and complex control problems may exhibit many local minima. It may be necessary to combine ILQR with global search methods (e.g. using multiple restarts).

7 Acknowledgments

This work is supported by the National Institutes of Health Grant 1-R01-NS045915-01.

References

[1] 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: 627-643, 1999.

[2] Arthur E. Bryson and Yu-Chi Ho. *Applied Optimal Control*. Blaisdell Publishing Company, Massachusetts, USA, 1969.

[3] C.M. Harris and D.M. Wolpert. Signal-dependent Noise Determines Motor Planning. *Nature*, Vol. 394, August 1998.

[4] David H. Jacobson and David Q. Mayne. *Differential Dynamic Programming*. Elsevier Publishing Company, New York, USA, 1970.

[5] Tarek Lahdhiri and Hoda A. Elmaraghy. Design of Optimal Feedback Linearizing-Based Controller for an Experimental Flexible-Joint Robot Manipulator. *Optimal Control Applications and Methods*, Volume 20, 165-182, 1999.

[6] Frank L. Lewis and Vassilis L. Syrmos. *Optimal Control*. John Wiley and Sons, New York, USA, 1995.

[7] Aristide Santos and Robert Bitmead. Nonlinear control design for an autonomous underwater vehicle (AUV) preserving linear design capabilities, *Proceedings of the 34th IEEE Conference on Decision and Control*, New Orleans, 1995, 3817-3823.

[8] E. Todorov and M. Jordan. Optimal Feedback Control as a Theory of Motor Coordination. *Nature Neuroscience*, 5(11): 1226, 2002.

[9] E. Todorov and W. Li. Optimal Control Methods Suitable for Biomechanical Systems. *The 25th IEEE Conference on Engineering in Medicine and Biology Society*,Cancun, Mexico, September 2003, in press.

[10] E. Todorov. On the role of primary motor cortex in arm movement control. *In Progress in Motor Control III*, ch 6, pp 125-166, Latash and Levin (eds),Human Kinetics.

[11] Uno Y, Kawato M and Suzuki R. Formation and Control of Optimal Trajectory in Human Multijoint Arm Movement - Minimum Torque-Change Model. *Biological Cybernetics* 61, 89-101, 1989.