2005 American Control Conference
June 8-10, 2005. Portland, OR, USA

An Efficient Sequential Linear Quadratic Algorithm for Solving Nonlinear Optimal Control Problems

Athanasios Sideris and James E. Bobrow
Department of Mechanical and Aerospace Engineering
University of California, Irvine
Irvine, CA 92697
{asideris, jebobrow}@uci.edu

Abstract— **We develop a numerically efficient algorithm for computing controls for nonlinear systems that minimize a quadratic performance measure. We formulate the optimal control problem in discrete-time, but many continuous-time problems can be also solved after discretization. Our approach is similar to sequential quadratic programming for finite-dimensional optimization problems in that we solve the nonlinear optimal control problem using sequence of linear quadratic subproblems. Each subproblem is solved efficiently using the Riccati difference equation. We show that each iteration produces a descent direction for the performance measure, and that the sequence of controls converges to a solution that satisfies the well-known necessary conditions for the optimal control.**

I. Introduction

As the complexity of nonlinear systems such as robots increases, it is becoming more important to find controls for them that minimize a performance measure such as power consumption. Although the Maximum Principle [1] provides the optimality conditions for minimizing a given cost function, it does not provide a method for its numerical computation. Because of the importance in solving these problems, many numerical algorithms and commercial software packages have been developed to solve them since the 1960's [2]. The various approaches taken can be classified as either indirect or direct methods. Indirect methods explicitly solve the optimality conditions stated in terms of the maximum principle, the adjoint equations, and the tranversality (boundary) conditions. Direct methods such as collocation [3] techniques, or direct transcription, replace the ODE's with algebraic constraints using a large set of unknowns. These collocation methods are powerful for solving trajectory optimization problems [4], [5] and problems with inequality constraints. However, due to the large-scale nature of these problems, convergence can be slow. Furthermore, optimal control problems can be difficult to solve numerically for a variety of reasons. For instance, the nonlinear system dynamics may create an unfavorable eigenvalue structure of the Hessian of the ensuing optimization problem, and gradient-based descent methods will be inefficient.

In this paper, we focus on the case of finding optimal controls for nonlinear dynamic systems with general linear quadratic performance indexes (including tracking and regulation). The discrete-time case is considered but continuous-time optimal control problems can handled after discretization. We are interested primarily in systems for which the derivatives of the dynamic equations with respect to the state and the control are available, but whose second derivatives with respect to the states and the controls are too complex to compute analytically. Our algorithm is based on linearizing the system dynamics about a input/state trajectory and solving a corresponding linear quadratic optimal control problem. From the solution of the latter problem, we obtain a search direction along which we minimize the performance index by direct simulation of the system dynamics. Given the structure of the proposed algorithm, we refer to it in the sequel as the *Sequential Linear Quadratic* (SLQ) algorithm. We prove that search directions produced in this manner are descent directions, and that this algorithm converges to a control that locally minimizes the cost function. Solution of the linear quadratic problem is well-known, and can be reliably obtained by solving a Riccati difference equation.

Algorithms similar in spirit are reported in [6], [7], [8], [9]. Such algorithms implement a Newton search, or asymptotically Newton search, but require that 2^{nd}-order system derivatives be available. Newton algorithms can achieve quadratic local convergence under favorable circumstances such as the existence of continuous 3^{rd}-order or Lipschiz continuous 2^{nd}-order derivatives, but cannot guarantee global convergence (that is convergence from any starting point to a local minimum) unless properly modified. Many systems are too complex for 2^{nd}-order derivatives to be available or even do not satisfy such strong continuity assumptions (e.g. see Section IV for examples.) In such cases, Newton's method cannot be applied, or the quadratic convergence rate of Newton's method does not materialize. We have found that our approach efficiently solves optimal control problems that are difficult to solve with other popular algorithms such as collocation methods (see Example, Section IV.) More specifically, we have observed that our algorithm exhibits near-quadratic convergence in many of the problems that we have tested. Indeed, it is shown in the full version of the paper [10], that the proposed algorithm can be interpreted as a Gauss-Newton method, thus explaining its excellent rate of convergence properties observed in simulations. Thus although many of the alternative methods ([2], [12], [11], [8]) can be applied to a broader class of problems, our SLQ algorithm provides a fast and reliable alternative to such algorithms for the important class of optimal control problems with quadratic cost under general nonlinear dynamics, while relying only on first derivative information.

II. Problem Formulation and Background Results

We consider the following general formulation of discrete-time optimal control problems.

$$\underset{u(n),\ x(n)}{\text{Minimize}} \quad J = \phi(x(N)) + \sum_{n=0}^{N-1} L(x(n), u(n), n) \tag{1}$$

$$\text{subject to} \quad x(n+1) = f(x(n), u(n)); \quad x(0) = x_0 \tag{2}$$

In the formulation above we assume a quadratic performance index, namely:

$$L(x(n), u(n), n) = \frac{1}{2}[x(n) - x^o(n)]^T Q(n)[x(n) - x^o(n)] + [u(n) - u^o(n)]^T R(n)[u(n) - u^o(n)] \tag{3}$$

and

$$\phi(x) = \frac{1}{2}[x - x^o(N)]^T Q(N)[x - x^o(N)] \tag{4}$$

In (3) and (4), $u^o(n)$, $x^o(n)$, $n = 1, \ldots\ N$ are given control input and state target sequences. In standard optimal regulator control problem formulations, $u^o(n)$, $x^o(n)$ are usually taken to be zero with the exception perhaps of $x^o(N)$, the desired final value for the state. The formulation considered here addresses the more general optimal tracking control problem and is required for the linear quadratic step in the proposed algorithm presented in Section III.1.

A. First Order Optimality Conditions

We next briefly review the first order optimality conditions for the optimal control problem of (1) and (2), in a manner that brings out certain important interpretations of the adjoint dynamical equations encountered in a control theoretic approach and Lagrange Multipliers found in a pure optimization theory approach.

Let us consider the *cost-to-go*:

$$J(n) \equiv \sum_{n=k}^{N-1} L(x(k), u(k), k) + \phi(x(N)) \tag{5}$$

with L and ϕ as defined in (3) and (4) respectively. We remark that $J(n)$ is a function of $x(n)$, and $u(k)$, $k = n, \ldots, N-1$ and introduce the sensitivity of the cost to go with respect to the current state:

$$\lambda^T(n) = \frac{\partial J(n)}{\partial x(n)} \tag{6}$$

From

$$J(n) = L(x(n), u(n), n) + J(n+1), \tag{7}$$

and since

$$\tfrac{\partial J(n+1)}{\partial x(n)} = \tfrac{\partial J(n+1)}{\partial x(n+1)} \cdot \tfrac{\partial x(n+1)}{\partial x(n)} = \lambda^T(n+1) f_x(x(n), u(n)),$$

we have the recursion:

$$\begin{aligned}\lambda^T(n) &= L_x(x(n), u(n), n) + \lambda^T(n+1) f_x(x(n), u(n)) \\ &= [x(n) - x^o(n)]^T Q(n) + \lambda^T(n+1) f_x(x(n), u(n))\end{aligned} \tag{8}$$

by using (3) and where L_x and f_x denote the partials of L and f respectively with respect to the state variables. The previous recursion can be solved backward in time ($n = N-1, \ldots, 0$) given the control and state trajectories and it can be started with the final value:

$$\lambda^T(N) = \frac{\partial L(N)}{\partial x(N)} = [x(N) - x^o(N)]^T Q(N) \tag{9}$$

derived from (4). In a similar manner, we compute the sensitivity of $J(n)$ with respect to the current control $u(n)$. Clearly from (7),

$$\begin{aligned}\frac{\partial J(n)}{\partial u(n)} &= L_u(x(n), u(n), n) + \lambda^T(n+1) f_u(x(n), u(n)) \\ &= [u(n) - u^o(n)]^T R(n) + \lambda^T(n+1) f_u(x(n), u(n)).\end{aligned} \tag{10}$$

In (10), L_u and f_u denote the partials of L and f respectively with respect to the control variables and (3) is used.

Next note that $\frac{\partial J}{\partial u(n)} = \frac{\partial J(n)}{\partial u(n)}$ since the first n terms in J do not depend on $u(n)$. We have then obtained the gradient of the cost with respect to the control variables, namely:

$$\nabla_u J = \begin{bmatrix} \frac{\partial J(0)}{\partial u(0)} & \frac{\partial J(1)}{\partial u(1)} & \ldots & \frac{\partial J(N-1)}{\partial u(N-1)} \end{bmatrix}. \tag{11}$$

Assuming u is unconstrained, the first order optimality conditions require that

$$\nabla_u J = 0. \tag{12}$$

We remark that by considering the Hamiltonian

$$H(x, u, \lambda, n) \equiv L(x, u, n) + \lambda^T f(x, u), \tag{13}$$

we have that $H_u(x(n), u(n), \lambda(n+1), n) \equiv \frac{\partial J}{\partial u(n)}$, i.e. we uncover the generally known but frequently overlooked fact that the partial of the Hamiltonian with respect to the control variables u is the gradient of the cost function with respect to u. We emphasize here that in our approach for solving the optimal control problem, we take the viewpoint of the control variables $u(n)$ being the independent variables of the problem since the dynamical equations express (recursively) the state variables in terms of the controls and thus can be eliminated from the cost function. Thus in taking the partials of J with respect to u, J is considered as a function $u(n)$, $n = 0, \ldots, N-1$ alone, assuming that $x(0)$ is given. With this perspective, the problem becomes one of unconstrained minimization, and having computed $\nabla_u J$, Steepest Descent, Quasi-Newton, and other first derivative methods can be brought to bear to solve it. However, due to the large-scale character of the problem, only methods that take advantage of the special structure of the problem become viable. The Linear Quadratic Regulator algorithm is such an approach in case of linear dynamics. We briefly review it next.

B. Linear Quadratic Tracking Problem

We next consider the case of linear dynamics in the optimal control problem of (1) and (2). In the following, we distinguish all variables corresponding to the linear optimal control problem that may have different values in the nonlinear optimal control problem by using an over-bar. When the cost is quadratic as in (3) we have the well-known Linear Quadratic Tracking problem. The control theoretic approach to this problem is based on solving the first order necessary optimality conditions (also sufficient in this case) in an efficient manner by introducing the Riccati equation. We briefly elaborate on this derivation next, for completeness and also since most references assume that the target sequences $x^o(n)$ and $u^o(n)$ are zero. First, we summarize the first order necessary optimality conditions for this problem.

$$\begin{aligned} \bar{x}(n+1) &= A(n)\bar{x}(n) + B(n)\bar{u}(n) && (14)\\ \bar{\lambda}^T(n) &= [\bar{x}(n) - \bar{x}^o(n)]^T Q(n) + \\ &\quad +\bar{\lambda}^T(n+1)A(n) && (15)\\ \partial\bar{J}(n)/\partial\bar{u}(n) &= [\bar{u}(n) - \bar{u}^o(n)]^T R(n) + \\ &\quad +\bar{\lambda}^T(n+1)B(n) = 0 && (16) \end{aligned}$$

Note that the system dynamical equations (14) run forward in time $n = 0, \ldots, N-1$ with initial conditions $\bar{x}(0) = \bar{x}_0$ given, while the adjoint dynamical equations (15) run backward in time, $n = N-1, \ldots, 0$ with final conditions $\bar{\lambda}^T(N) = [\bar{x}(N) - \bar{x}^o(N)]^T Q(N)$. From (16), we obtain

$$\bar{u}(n) = \bar{u}^o(n) - R^{-1}(n)B^T(n)\bar{\lambda}(n+1) \qquad (17)$$

and by substituting in (14) and (15), we obtain the classical two-point boundary system but with additional forcing terms due to the $\bar{x}^o(n)$ and $\bar{u}^o(n)$ sequences.

$$\begin{aligned} \bar{x}(n+1) &= A(n)\bar{x}(n) - B(n)R^{-1}(n)B^T(n)\bar{\lambda}(n+1) + \\ &\quad +B(n)\bar{u}^o(n) && (18)\\ \bar{\lambda}^T(n) &= Q(n)\bar{x}(n) + A^T(n)\bar{\lambda}(n+1) - \\ &\quad -Q(n)\bar{x}^o(n). && (19) \end{aligned}$$

The system of (18) and (19) can be solved by the *sweep method* [1], based on the postulated relation

$$\bar{\lambda}(n) = P(n)\bar{x}(n) + s(n) \qquad (20)$$

where $P(n)$ and $s(n)$ are appropriate matrices that can be found as follows. For $n = N$, (20) holds with

$$P(N) = Q(N), \qquad s(N) = -Q(N)\bar{x}^o(N). \qquad (21)$$

We now substitute (20) in (18) and after some algebra we obtain

$$\bar{x}(n+1) = M(n)A(n)\bar{x}(n) + v(n) \qquad (22)$$

where we defined

$$\begin{aligned} M(n) &= \left[I + B(n)R^{-1}(n)B^T(n)P(n+1)\right]^{-1} && (23)\\ v(n) &= M(n)B(n)[\bar{u}^o(n) - R^{-1}(n)B^T(n)s(n+1)]. && (24) \end{aligned}$$

By replacing $\bar{\lambda}(n)$ and $\bar{\lambda}(n+1)$ in (19) in terms of $\bar{x}(n)$ and $\bar{x}(n+1)$ from (20), we obtain

$$\begin{aligned} P(n)\bar{x}(n) + s(n) &= \\ &= Q(n)\bar{x}(n) + A^T(n)\left[P(n+1)\bar{x}(n+1) + \right.\\ &\quad \left. +s(n+1)\right] - Q(n)\bar{x}^o(n), \end{aligned}$$

and by expressing $\bar{x}(n+1)$ from (22) and (24) above, we get

$$\begin{aligned} P(n)\bar{x}(n) + s(n) &= \\ &= Q(n)\bar{x}(n) + A^T(n)P(n+1)M(n)A(n)\bar{x}(n) - \\ &\quad -A^T(n)P(n+1)M(n)B(n)R^{-1}(n)B^T(n)s(n+1) \\ &\quad +A^T(n)P(n+1)M(n)B(n)\bar{u}^o(n) + \\ &\quad +A^T(n)s(n+1) - Q(n)\bar{x}^o(n). \end{aligned}$$

The above equation is satisfied by taking

$$\begin{aligned} P(n) &= Q(n) + A^T(n)P(n+1)M(n)A(n) && (25)\\ s(n) &= A^T(n)\left[I - P(n+1)M(n)B(n)R^{-1}(n)\cdot \right.\\ &\quad \left. \cdot B^T(n)\right] s(n+1) + \\ &\quad +A^T(n)P(n+1)M(n)B(n)\bar{u}^o(n) - \\ &\quad -Q(n)\bar{x}^o(n) && (26) \end{aligned}$$

Equation (25) is the well-known Riccati difference equation and together with the auxiliary equation (26), which is unnecessary if $\bar{x}^o(n)$ and $\bar{u}^o(n)$ are zero, are solved backward in time ($n = N-1, \ldots, 1$), with final values given by (21) and together with (23) and (24). The resulting values $P(n)$ and $s(n)$ are stored and used to solve forward in time (22) and (17) for the optimal control and state trajectories.

III. Main Results

A. Formulation of the SLQ Algorithm

In the proposed SLQ algorithm, the control at stage $k+1$ is found by performing a one-dimensional search from the control at stage k and along a search direction that is found by solving an Linear Quadratic (LQ) optimal control problem. Specifically, let $U_k = [u^T(0)\ u^T(1) \ldots u^T(N-1)]^T$ be the optimal solution candidate at step k, and $X_k = [x^T(1)\ x^T(2) \ldots x^T(N)]^T$ the corresponding state trajectory obtained by solving the dynamical equations (2) using U_k and with the initial conditions $x(0)$. We next linearize the state equations (2) about the nominal trajectory of U_k and X_k. The linearized equations are

$$\bar{x}(n+1) = f_x(x(n),\ u(n))\bar{x}(n) + f_u(x(n),\ u(n))\bar{u}(n) \qquad (27)$$

with initial conditions $\bar{x}(0) = 0$. We then minimize the cost index (1) with respect to $\bar{u}(n)$. The solution of this LQ problem gives $\bar{U}_k = [\bar{u}^T(0)\ \bar{u}^T(1) \ldots \bar{u}^T(N-1)]^T$, the proposed search direction. Thus, the control variables at stage $k+1$ of the algorithm are obtained from

$$U_{k+1} = U_k + \alpha_k \cdot \bar{U}_k \qquad (28)$$

where $\alpha_k \in \mathbb{R}^+$ is appropriate stepsize the selection of which is discussed later in the paper. Note again our perspective of considering the optimal control problem as an unconstrained finite-dimensional optimization problem in U.

We emphasize that $\bar{U}_k$ as computed above is *not* the steepest descent direction. It is the solution to a linear quadratic tracking problem for a nonlinear system that has been linearized about U_k. Note that the objective function is not linearized for this solution. Our algorithm is different than standard *Quasilinearization* [1] and *Neighboring Extremal* [13] methods where the adjoint equations are also linearized and two-point boundary problems are solved. As it was mentioned earlier, our algorithm corresponds to the Gauss-Newton method for solving the optimal control problem of (1) and (2).

B. Properties of the SQL Algorithm

In this section, we prove two important properties of the proposed algorithm. First, we show that search direction $\bar{U}$ is a descent direction.

Theorem 1: Consider the discrete-time nonlinear optimal control problem of (1) and (2), and assume a quadratic cost function as in (3) and (4) with $R(n) = R^T(n) > 0$, $Q(n) = Q^T(n) \geq 0$, $n = 0, 1, \ldots, N-1$, and $Q(0) = 0$, $Q(N) = Q^T(N) \geq 0$. Also consider a control sequence $U \equiv [u^T(0) \ldots u^T(N-1)]^T$ and the corresponding state trajectory $X \equiv [x^T(1) \ldots x^T(N)]^T$. Next, linearize system (2) about U and X and solve the following linear quadratic problem:

$$\begin{aligned} \underset{\bar{u}(n),\bar{x}(n)}{\text{Minimize}} \ \bar{J} \ = \ & \frac{1}{2}[\bar{x}(N) - \bar{x}^o(N)]^T Q(N)[\bar{x}(N) + \bar{x}^o(N)] + \\ & + \frac{1}{2}\sum_{n=0}^{N-1}\Big\{[\bar{x}(n) - \bar{x}^o(n)]^T Q(n)[\bar{x}(n) - \bar{x}^o(n)] + \\ & + [\bar{u}(n) - \bar{u}^o(n)]^T R(n)[\bar{u}(n) - \bar{u}^o(n)]\Big\} \quad (29) \end{aligned}$$

subject to

$$\begin{aligned} \bar{x}(n+1) = & f_x(x(n), u(n))\bar{x}(n) + \\ & + f_u(x(n), u(n))\bar{u}(n); \quad (30) \\ \bar{x}(0) = & 0, \end{aligned}$$

where $\bar{x}^o(n) \equiv x^o(n) - x(n)$, $\bar{u}^o(n) \equiv u^o(n) - u(n)$. Then if $\bar{U} \equiv [\bar{u}^T(0) \ldots \bar{u}^T(N-1)]^T$ is not zero, it is a descent direction for the cost function (1), i.e. $J(U + \alpha \cdot \bar{U}) < J(U)$ for some $\alpha > 0$.

Proof: We establish that $\bar{U}$ is a descent direction by showing that:

$$\nabla_u J \cdot \bar{U} = \sum_{n=0}^{N-1} \frac{\partial J(n)}{\partial u(n)} \bar{u}(n) < 0, \quad (31)$$

since $\nabla_u J$ in (11) is the gradient of the cost function with respect to the control variables. Now, the components of $\nabla_u J$ are expressed in (10) in terms of the adjoint variables $\lambda(n)$ that satisfy recursion (8) with final values given by (9). On the other hand, $\bar{x}(n)$ and $\bar{u}(n)$ together with adjoint variables $\bar{\lambda}(n)$ satisfy the first order optimality conditions for the linear quadratic problem given in (14), (15) and (16), where $A(n) = f_x(x(n), u(n))$ and $B(n) = f_u(x(n), u(n))$. Let us define

$$\tilde{\lambda}(n) = \bar{\lambda}(n) - \lambda(n) \quad (32)$$

and note from (8) and (15) that

$$\begin{aligned} \tilde{\lambda}^T(n) \ &= \ \bar{x}^T(n)Q(n) + \tilde{\lambda}^T(n+1)A(n) \\ \tilde{\lambda}(N) \ &= \ Q(N)\bar{x}(N). \end{aligned} \quad (33)$$

Next through the indicated algebra, we can establish the following relation:

$$\begin{aligned} & \frac{\partial J(n)}{\partial u(n)} \cdot \bar{u}(n) = \\ & = \big([u(n) - u^o(n)]^T R(n) + \lambda^T(n+1)B(n)\big)\bar{u}(n) \\ & \quad (\text{using } (10)) \\ & = -\tilde{\lambda}^T(n+1)B(n)\bar{u}(n) - \bar{u}^T(n)R(n)\bar{u}(n) \\ & \quad (\text{using } (16)) \\ & = -\tilde{\lambda}^T(n+1)\bar{x}(n+1) + \tilde{\lambda}^T(n+1)A(n)\bar{x}(n) - \\ & \quad -\bar{u}^T(n)R(n)\bar{u}(n) \quad (\text{using } (30)) \\ & = -\tilde{\lambda}^T(n+1)\bar{x}(n+1) + \tilde{\lambda}^T(n)\bar{x}(n) - \\ & \quad -\bar{x}^T(n)Q(n)\bar{x}(n) - \bar{u}^T(n)R(n)\bar{u}(n). \\ & \quad (\text{using } (33)) \end{aligned}$$

Finally, summing up the above equation from $n = 0$ to $n = N-1$ and noting that $\bar{x}(0) = 0$ and from (21) that $\bar{\lambda}(N) = Q(N)\bar{x}(N)$, gives:

$$\begin{aligned} \nabla_u J \cdot \bar{U} \ &= \ \sum_{n=0}^{N-1} \frac{\partial J(n)}{\partial u(n)} \cdot \bar{u}(n) \\ &= \ -\sum_{n=0}^{N-1} [\bar{x}^T(n)Q(n)\bar{x}(n) + \bar{u}^T(n)R(n)\bar{u}(n)] - \\ & \quad -\bar{x}^T(N)Q(N)\bar{x}(N) \ < \ 0 \end{aligned} \quad (34)$$

and the proof of the theorem is complete. ■

We remark that the search direction $\bar{U}$ can be found by the LQ algorithm of Section II.B with $A(n) \equiv f_x(x(n), u(n))$ and $B(n) \equiv f_u(x(n), u(n))$.

The next result shows that the proposed SLQ algorithm does in fact converge to a control locally minimizing the cost function (1). We denote by $J[U]$ the cost associated with the control $U = [u^T(0) \ldots u^T(N-1)]^T$ (and the given initial conditions $x(0)$.)

Theorem 2: Starting with an arbitrary control sequence U_0, compute recursively new controls from (28) where the direction $\bar{U}_k$ is obtained as in Theorem 1 by solving the LQ problem of (29) and the linearized system (30) about the current solution candidate U_k and corresponding state trajectory X_k; also α_k is obtained by minimizing $J[U_k + \alpha \bar{U}_k]$ over $\alpha > 0$. Then every limit point of U_k gives a control that satisfies the first order optimality conditions for the cost function (1) subject to the system equations (2).

Proof: Let us define $D_k \equiv \nabla_U J[U_k] \cdot \bar{U}_k < 0$, the derivative of $J[U_k + \alpha\bar{U}_k]$ with respect to α at $\alpha = 0$. We first show that under the exact line search assumption, D_k has a subsequence that converges to zero as $k \to \infty$. Notice that for any fixed ϵ such that $0 < \epsilon < 1$ and α sufficiently small, it holds

$$J[U_k + \alpha\bar{U}_k] \leq J[U_k] + \epsilon\alpha D_k. \quad (35)$$

Let $\bar{\alpha}_k$ be such that (35) is satisfied with equality (such $\bar{\alpha}_k < \infty$ exists since otherwise $J[U_k + \alpha\bar{U}_k] \to -\infty$ as $\alpha \to \infty$, contradicting the fact that J is bounded below by zero.) From the mean value theorem, there exists β_k with $0 \leq \beta_k \leq \bar{\alpha}_k$ such that

$$J[U_k + \beta_k\bar{U}_k] \cdot \bar{U}_k = \epsilon D_k. \quad (36)$$

Then, it holds

$$J[U_k] - J[U_{k+1}] \geq J[U_k] - J[U_k + \bar{\alpha}_k\bar{U}_k] = \epsilon\bar{\alpha}_k D_k. \quad (37)$$

Since $J[U_k]$ is monotonically decreasing and bounded below (by zero), it converges to some value and (37) implies that $\bar{\alpha}_k D_k \to 0$. Let us assume that $\|D_k\| \geq \eta > 0$ for all k. Then, it must be that $\bar{\alpha}_k \to 0$ and thus $\beta_k \to 0$. Now divide both sides of (36) by $\|D_k\|$ and note that since $\bar{D}_k \equiv D_k/\|D_k\|$ belongs in the closed unit ball of $\mathbb{R}$, a compact set, it has at least one limit point $\bar{D}^*$ that satisfies from (36), $\bar{D}^* = \epsilon\bar{D}^*$, or $\bar{D}^* = 0$, which a contradiction. Therefore, we conclude that there is a subsequence of D_k that converges to zero.

Next, from (34), we obtain:

$$D_k = \nabla_U J[U_k] \cdot \bar{U}_k \leq -\rho\|\bar{U}_k\|^2 \leq 0, \quad (38)$$

where $\rho > 0$ is a uniform lower bound on the minimum eigenvalue of $R(n),\ n = 0, \ldots, N-1$. Then (38) implies that there is a corresponding subsequence of $\bar{U}_k$ that converges to zero. For notational convenience, we identify this subsequence of $\bar{U}_k$ (and corresponding subsequences of other sequences) with the whole sequence. But $\bar{U}_k \to 0$ implies from (14) that $\bar{X}_k \to 0$ (note that $\bar{x}(0) = 0$ and that $A(n)$ and $B(n)$ can be assumed to be uniformly bounded since $U_{k+1} - U_k = \alpha_k\bar{U}_k \to 0$.) Consequently, (33) implies that $\bar{\lambda}(n) \to \lambda(n)$ for $n = 0, \ldots, N-1$, and that the right-hand-side of (10) converges to the right-hand-side of (16) which is zero by the optimality of $\bar{U}_k$ for the LQ problem. This shows that the first order optimality conditions (10) for the nonlinear control problem are asymptotically satisfied, i.e. $\nabla_U J[U_k] \to 0$ as $k \to \infty$. We remark that the last conclusion follows for a subsequence of U_k, but since the cost $J[U_k]$ is monotonically decreasing and clearly converges to a locally minimum value, any limit point of U_k must be a stationary point. ■

We remark, that the exact line search in Theorem 2 can be replaced with an inaccurate line search that satisfies some standard conditions such as in the Armijo, Goldstein, or Wolfe stepsize selection rules [14], [15]. Similar arguments as in previous proof can be used to show that $D_k \to 0$ still follows and the proof is completed as above.

IV. Numerical Example: A Gas Actuated Hopping Machine

An interesting optimal control problem is that of creating motions for an autonomous hopping machine. A simple model for a gas actuated one-dimensional hopping system is shown on the left-hand side of Figure 1. This system is driven by a pneumatic actuator, with the location of the piston relative to the mass under no external loading defined as y_p. After contact occurs with the ground with $y \leq y_p$, the upward force on the mass from actuator can be approximated by a linear spring with $F = k(y_p - y)$, where k is the spring constant. The position y_p can be viewed as the unstretched spring length and it can be easily changed by pumping air into or out of either side of the cylinder. The equations of motion for the mass are $m\ddot{y} = F(y, y_p) - mg$, where mg is the force due to gravity, and $F(y, y_p) = \begin{cases} 0 & y > y_p \\ k(y_p - y) & \text{otherwise.} \end{cases}$ Note that in this case $F(y, y_p)$ is not differentiable at $y = y_p$, and the proposed algorithm can not be used on this problem. However, the discontinuity in the derivative can easily be smoothed. For instance, let the spring compression be $e = y_p - y$ and choose an $\alpha > 0$, then

$$F(e) = \begin{cases} 0 & 0 > e \\ \frac{k}{2\alpha}e^2 & 0 \leq e < \alpha \\ ke - \frac{k\alpha}{2} & \text{otherwise} \end{cases}$$

is C^1. The final equation of motion for this system relates the air flow into the cylinder, which is the control $u(t)$, to the equilibrium position y_p of the piston. Assume for the following that the equation $\dot{y}_p = u$ approximates this relationship.

When the hopping machine begins its operation, we are interested in starting from rest, and reaching a desired hop height y_N^o at time t_f. If we minimize

$$\begin{aligned} J(u) \quad = \quad & \frac{1}{2}q_{fin}(y(N) - y_N^o)^2 + \dot{y}(N)^2 \\ & + \frac{t_f}{2N}\sum_{n=0}^{N-1}\left[q\ y_p(n)^2 + r\ u(n)^2\right], \end{aligned} \quad (39)$$

the terms outside the summation reflect the desire to reach the height at time t_f with zero velocity, and the terms inside the summation reflect the desire to minimize the gas used to achieve this. The weighting on y_p is used to keep the piston motion within its bounds. We conducted numerical experiments on this system with the following parameters: $k/m = 500$, $g = 386.4$, $\alpha = 0.1$. We assumed that all states were initially zero, and that the initial control sequence was zero. The cost function parameters were selected as: $y_N^o = 20$, $t_f = 1$, $q_{fin} = q = 1000$, and $r = 1.0$. A simple Euler approximation was used to discretize the equations, with $N = 100$. Note that the algorithm produced an alternating sequence of stance phases and flight phases for

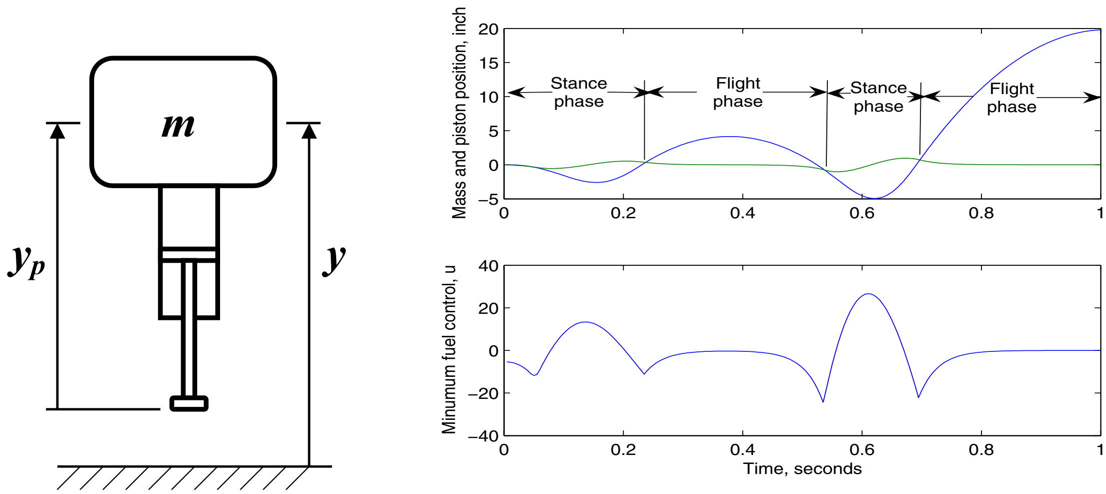

Fig. 1. Maximum height hopping motion and minimum fuel control.

the hopping system and it naturally identified the times to switch between these phases. If one were to use collocation methods to solve this problem with explicit consideration of the different dynamics in the different phases, one would have to guess the number of switches between phases and would need to treat the times at which the switch occurs as variables in the optimization. In the full version of the paper [10] (omitted here due to space limitations), we show that the proposed SLQ algorithm can be interpreted as a Gauss-Newton algorithm and thus explain its excellent rate of convergence observed in simulations. Consistent with this interpretation, our algorithm converges much faster when the weighting on the control r is increased; also the number of iterations required for convergence in this problem increases for larger y_N^o, ranging from 3 for $y_N^o = 1$, to 166 for $y_N^o = 50$. In addition, the algorithm failed to converge for $\alpha < 1 \times 10^{-5}$, which demonstrates the need for the dynamics to be continuously differentiable.

V. Conclusion

We developed an algorithm for solving nonlinear optimal control problems with quadratic performance measures and unconstrained controls. Each subproblem in the course of the algorithm is a linear quadratic optimal control problem that can be efficiently solved by Riccati difference equations. We show that each search direction generated in the linear quadratic subproblem is a descent direction, and that the algorithm is convergent. Computational experience has demonstrated that the algorithm converges quickly to the optimal solution. The SLQ algorithm is proposed as a powerful alternative to Newton methods in situations that second order derivative information on the system dynamics is not available or expensive to obtain and a near real-time solver is required.

References

[1] A.E. Bryson and Y.C. Ho, *Applied Optimal Control,* Wiley, New York, 1995.

[2] J.T. Betts, "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control and Dynamics,* V. 21: (2) 193-207, 1999.

[3] O. von Stryk, "Numerical solution of optimal control problems by direct collocation," *Optimal Control*, Bulirsch R, Miele A, Stoer J, Well KH (eds), *International Series of Numerical Mathematics,* vol. 111. Birkshauser Verlag; Basel, 1993; 129143.

[4] C. R. Hargraves and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *Journal of Guidance, Control, and Dynamics,* 1987; 10:338342.

[5] P. J. Enright and B. A. Conway, "Optimal finite-thrust spacecraft trajectories using collocation and nonlinear programming," *Journal of Guidance, Control, and Dynamics,* 1991; 14:981 985.

[6] J. F. Pantoja, "Differential Dynamic Programming and Newton's Method," *International Journal on Control,* Vol. 47, No. 5, pp. 1539-1553, 1988.

[7] J. C. Dunn and D. P. Bertsekas, "Efficient Dynamic Programming Implementations of Newton's Method for Unconstrained Optimal Control Problems," *Journal of Optimization Theory and Applications,* Vol. 63, No. 1, pp. 23-38, 1989.

[8] J. F. Pantoja and D. Q. Mayne, "Sequential Quadratic Programming Algorithm for Discrete Optimal Control Problems with Control Inequality Constraints," *International Journal on Control,* Vol. 53, No. 4, pp. 823-836, 1991.

[9] L. Z. Liao and C. A. Shoemaker, "Convergence in Unconstrained Discrete- Time Differential Dynamic Programming," *IEEE Transactions on Automatic Control,* Vol. 36, No. 6, pp. 692-706, 1991.

[10] A. Sideris and J. E. Bobrow, "An Efficient Sequential Linear Quadratic Algorithm for Solving Nonlinear Optimal Control Problems," submitted to *IEEE Transactions on Automatic Control,* June 2004.

[11] S. J. Wright, "Interior-Point Methods for Optimal Control of Discrete-Time Systems," *Journal of Optimization Theory and Applications,* Vol. 77, No. 1, pp. 161-187, 1993.

[12] C. R. Dohrmann and R. D. Robinett, "Dynamic Programming Method for Constrained Discrete-Time Optimal Control," *Journal Of Optimization Theory And Applications,* Vol. 101, No. 2. pp. 259-283, 1999.

[13] T. Veeraklaew and S. K. Agrawal, "Neighboring optimal feedback law for higher-order dynamic systems," *ASME J. Dynamic Systems, Measurement, and Control,* 124 (3): 492-497 SEP 2002.

[14] R. Fletcher, *Practical Methods of Optimization*, Wiley, 2nd edition, 1987.

[15] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Harcourt Brace, 1981.