SAND 97-2147 C
SAND-97-2147C

A Multi-Objective Dynamic Programming Approach to Constrained Discrete-Time Optimal Control

CONF-980623--

Brian J. Driessen
Kwan S. Kwok
(Sandia National Laboratories, Albuquerque, NM)

RECEIVED

SEP 23 1997

OSTI

Abstract

This work presents a multi-objective differential dynamic programming approach to constrained discrete-time optimal control. In the backward sweep of the dynamic programming in the quadratic sub problem, the sub problem input at a stage or time step is solved for in terms of the sub problem state entering that stage so as to minimize the summed immediate and future cost subject to minimizing the summed immediate and future constraint violations, for all such entering states. The method differs from previous dynamic programming methods, which used penalty methods, in that the constraints of the sub problem, which may include terminal constraints and path constraints, are solved exactly if they are solvable; otherwise, their total violation is minimized. Again, the resulting solution of the sub problem is an input history that minimizes the quadratic cost function subject to being a minimizer of the total constraint violation. The expected quadratic convergence of the proposed algorithm is demonstrated on a numerical example.

1. Introduction

The work of this paper presents a new approach to solving constrained optimal control quadratic sub problems. The resulting sub problem solution can be used in an iterative procedure for solving a constrained nonlinear optimal control problem. The approach to the sub problem differs from most previous work, which use penalty methods to incorporate constraints. The proposed multi-objective dynamic programming approach presented herein instead finds a sub problem input history that minimizes the cost function subject to being a minimizer of the total constraint violation. Thus, if the (linear) constraints of the sub problem are solvable, the proposed method solves them exactly,

19980423 093

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DTIC QUALITY INSPECTED

MASTER

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

while if they are not solvable, the method minimizes the objective function over the set of input histories that minimize the total constraint violation.

We will now briefly review some of the relevant optimal control work, while restricting ourselves to those whose algorithm's exhibit Order(N) complexity, where N is the number of time steps. Wright ([1] and [2]) considered optimal control problems with inequality constraints. He used a banded matrix approach and an interior point (barrier) algorithm to avoid the Order(N^2) or greater complexity associated with active set methods for handling inequality constraints. Ruxton [3] also considered inequality constrained problems, and used a differential dynamic programming approach, in which the cost function included a penalty term for the constraint violations. Murray and Yakowitz [4] used a quadratically convergent differential dynamic programming algorithm for optimal control problems. The DDP (differential dynamic programming) algorithm avoids the need to calculate Lagrange multipliers associated with the state equation constraints. DDP also provides a convenient means of Hessian modification during its backward sweep.

In numerous dynamic system trajectory optimization problems, the only inequality constraints are bounds on the input variables. Bertsekas [6] utilized this fact in the design of a semi-Newton method in either a differential dynamic programming or Ricatti method approach. He correctly claimed that in many dynamic system trajectory optimization problems, the use of a simplex pivoting algorithm (such as those which are often used to solve quadratic and linear programs (see Bazaraa et al [8])) would be impracticable, especially in those problems in which there are many time steps (N large). Also, Gawande and Dunn [7] applied such a projected Newton algorithm for input-constrained optimal control problems.

Dohrmann and Robinett [9] proposed a DDP approach that solved the imposed equality constraints (such as terminal state constraints and path constraints) of the quadratic sub problem exactly. This was accomplished by stepping back enough time

steps until the immediate and future constraints could be satisfied for any incoming state to that time step. They then performed a backward sweep based on Bellman's Principle, followed by the forward sweep. To this author's knowledge, this work of Dohrmann and Robinett [9] is the first reasonable published DDP method that solves the sub problem constraints exactly, as opposed to incorporating them into a penalty function. The work presented in the present paper is an alternative approach for solving the constrained sub problem exactly. Instead of stepping back enough time steps until the immediate and future constraints can be satisfied for any incoming state to that time step, the method proposed herein uses a multi-objective approach in which the input at *any* time step minimizes the summed immediate and future cost subject to being a minimizer of the summed immediate and future constraint violations, for any state coming into that stage.

There has been much work with shooting methods. Such methods can be found in Ho and Bryson [10] and in Lewis [11]. Wie and Sunkel [12] solved a minimum time pointing control problem using such an approach. They noted the fact that this approach requires an extremely good initial guess for the value of the initial costate. Li and Bainum [13] also approached a minimum time problem for rigid spacecraft maneuvers via a shooting method, and they treated a quadratic function of the controls as their cost and sequentially shortened the slewing time. Geering et al [14] is yet another example of the use of shooting methods for minimum time problems, and they reported that they were unable to obtain solutions for an IBM robot problem using the shooting method alone. The shooting method (or two-point boundary value problem method) is known to be much more sensitive to initial guess than methods that use Order(N) variables that span the total time interval, as do the mathematical programming based approaches such as SQP (sequential quadratic programming) (see Bazaraa et al [8] and Gill et al [15]) methods, which include DDP. This difference in robustness is especially true for minimum time problems.

This paper will first present some preliminary theory that will be used throughout the paper. Then, the recursive details of the Order(N) multi-objective dynamic programming approach to the quadratic sub problem will be presented. A numerical example will be presented, and finally conclusions will be given.

2. Preliminary Theory

In this section we will introduce a technical lemma that will be used throughout this paper. Consider the following Problem I

$$\text{Min}_{x_1, x_2} f(x_1, x_2) \quad (\text{Problem I})$$

subject to

$$g(x_1, x_2) \leq g(x, y) \quad \forall x, y \quad (2.1)$$

where g is bounded below and f is bounded below on the set S defined below

$$S \equiv \{(x_1, x_2) : g(x_1, x_2) \leq g(x, y) \quad \forall x, y\} \quad (2.2)$$

and where f and g both map from R^n to R^1 .

Consider also Problem II below

$$\text{Min}_{x_2} f(x_1, x_2) \quad (\text{Problem II})$$

$$\text{subject to } g(x_1, x_2) \leq g(x_1, y) \quad \forall x_1, y \quad (2.3)$$

Suppose we have a function h such that $x_2 = h(x_1)$ solves Problem II for *any* fixed or given x_1 .

Consider finally Problem III below

$$\text{Min}_{x_1} f(x_1, h(x_1)) \quad (\text{Problem III})$$

subject to

$$g(x_1, h(x_1)) \leq g(x, h(x)) \quad \forall x \quad (2.4)$$

Suppose x_1^* solves Problem III.

Lemma 1. $(x_1^*, x_2^*) \equiv (x_1^*, h(x_1^*))$ solves Problem I.

Although Lemma 1 is rather intuitively obvious, we present a rigorous proof below.

Proof of Lemma 1. Let $g_{\min}$ be the minimum value of $g(x_1, x_2)$. Consider any (x_1, x_2) . By the definition of the function h ,

$$g(x_1, h(x_1)) \leq g(x_1, x_2) \quad \forall x_1, x_2 \quad (2.5)$$

By the definition of x_1^* ,

$$g(x_1^*, h(x_1^*)) \leq g(x_1, x_2) \quad \forall x_1, x_2 \quad (2.6)$$

Combining (2.5) and (2.6), we have,

$$g(x_1^*, h(x_1^*)) \leq g(x_1, x_2) \quad (2.7)$$

Thus,

$$g(x_1^*, x_2^*) \leq g(x, y) \quad \forall x, y \quad (2.8)$$

which is exactly equation (2.1). Thus, $(x_1^*, x_2^*) \in S$ (from equation (2.2)). Also, therefore,

$$g(x_1^*, x_2^*) = g_{\min}.$$

By the definition of the function h , certainly

$$f(x_1, h(x_1)) \leq f(x_1, x_2) \quad (2.9)$$

$$\text{for all } (x_1, x_2) \in S_2 \equiv \{(x_1, x_2): x_2 = h(x_1)\} \quad (2.10)$$

(Agreeably this is almost too obvious to state.) By the definition of x_1^* ,

$$f(x_1^*, h(x_1^*)) \leq f(x_1, x_2) \quad (2.11)$$

$$\text{for all } (x_1, x_2) \in S_3 \equiv \{(x_1, x_2): x_2 = h(x_1) \text{ and } g(x_1, x_2) \leq g(x, h(x)) \forall x\}. \quad (2.12)$$

Importantly, S_3 is a subset of S_2 , and S_3 is a subset of S since any point in S_3 has $g = g_{\min}$.

Combining the fact that $S_3 \subset S_2$ and (2.10) through (2.12), we have,

$$f(x_1^*, h(x_1^*)) \leq f(x_1, h(x_1)) \text{ and } f(x_1, h(x_1)) \leq f(x_1, x_2) \quad (2.13)$$

for all $(x_1, x_2) \in S_3$.

So, if we can prove that all optimal solutions of Problem I are contained in S_3 , the proof of Lemma 1 will be complete as (2.12) will imply $f(x_1^*, h(x_1^*)) \leq f(x_1, x_2) \forall (x_1, x_2) \in S$. So, let us prove that all optimal solutions of Problem I are contained in S_3 . Proof that there exists no point $(\bar{x}_1, \bar{x}_2)$ in S but not in S_3 for which $f(\bar{x}_1, \bar{x}_2) < f(\bar{x}_1, h(\bar{x}_1))$: Proof by contradiction: Suppose $g(\bar{x}_1, \bar{x}_2) = g_{\min}$ (i.e.,

$(\bar{x}_1, \bar{x}_2) \in S$ and $f(\bar{x}_1, \bar{x}_2) < f(\bar{x}_1, h(\bar{x}_1))$. By the definition of h and the fact that $g(\bar{x}_1, \bar{x}_2) = g_{\min}$, we have $g(\bar{x}_1, h(\bar{x}_1)) = g_{\min}$. Also, by the definition of h , we have $f(\bar{x}_1, h(\bar{x}_1)) \leq f(\bar{x}_1, \bar{x}_2)$, a contradiction. QED.

So, for Problem I, we need only consider points in S_3 . Equation (2.13) states that $f(x_1^*, h(x_1^*)) \leq f(x_1, x_2)$ for all points (x_1, x_2) in this subset S_3 . This completes the proof of Lemma 1. QED.

Let us now consider the special case of Problem I in which f and g are quadratic functions, and where g is bounded below, and f is bounded below on the set of minimizers of g . We have

$$\begin{aligned} f &= \frac{1}{2} x^T H x + x^T L u + \frac{1}{2} u^T P u + q^T x + s^T u \\ g &= \frac{1}{2} x^T \hat{H} x + x^T \hat{L} u + \frac{1}{2} u^T \hat{P} u + \hat{q}^T x + \hat{s}^T u \end{aligned}$$

Let us first find the function h . By setting the derivative of g equal to zero, we have

$$g_u = \hat{P} u + \hat{L}^T x + \hat{s} = 0 \quad (2.14)$$

Note: Since g is a convex bounded-below function of u for any fixed or given x , the equation that expresses that $g_u = 0$ (equation (2.14)) is guaranteed to have a solution for any x . Equation (2.14) represents our constraint on u . The necessary conditions for minimizing f subject to (2.14) are (2.15) below:

$$P u + L^T x + s + \hat{P}^T \lambda = 0 \quad (2.15)$$

and (2.14) itself, where λ in (2.15) is a Lagrange multiplier. Note: If P were not positive definite (p.d.) on the null space of $\hat{P}$ ($N(\hat{P})$), we would replace P with $\tilde{P}$ where $\tilde{P}$ is p.d. on $N(\hat{P})$. (Since we are assuming that f is bounded below over the set of minimizers of g , P will be at least positive-semi-definite on $N(\hat{P})$.) In summary, the following two simultaneous equations in u and λ are guaranteed to have a solution, for any x :

$$P u + \hat{P}^T \lambda = -L^T x - s \quad (2.16)$$

$$\hat{P} u = -\hat{L}^T x - \hat{s} \quad (2.17)$$

Solving (2.16)/(2.17) gives us u in terms of x , i.e.

$$u = Gx + e \quad (2.18)$$

One way to solve (2.16)/(2.17) is as follows. To solve (2.16)/(2.17), we will first use the range space of $\hat{P}$ to solve (2.17), and then minimize f on the null space of $\hat{P}$, given our value for the range space component.

Let the decomposition of $\hat{P}$ be denoted by

$$\hat{P} = VDV^T \quad (2.19)$$

where D is a diagonal matrix of eigenvalues λ_i and V the eigenvectors. If $\lambda_i < \text{some tolerance}$, set $\lambda_i = 0$. Consider first the case of $x=0$. Let $y_i = (-V^T \hat{s})_i$ if $\lambda_i \neq 0$ and 0 otherwise. Let $u^1 = V^T y$. For the null space part of u , let N be a basis for the null space of $\hat{P}$ and solve

$$(N^T P N) y = -N^T s - N^T P u^1 \quad (2.20)$$

Let $u^2 = N^T y$. Finally, the total solution is $e = u^1 + u^2$. Then, we repeat this same procedure, replacing $\hat{s}$ with each column of $\hat{L}^T$ and s with each column of L^T , thus obtaining each respective column of G in equation (2.18).

3. Problem Statement

The problem is the following equality-constrained optimal control problem, herein denoted by OCP.

$$\text{Minimize } \gamma = \sum_{k=1}^N \gamma_k(x_k, u_k) + \gamma_{N+1}(x_{N+1}) \quad (3.1)$$

subject to

$$x_{k+1} = f(x_k, u_k), (k=1, \dots, N) \quad (3.2)$$

$$g(x_k, u_k) = 0, (k=1, \dots, N) \quad (3.3)$$

$$g_{N+1}(x_{N+1}) = 0 \quad (3.4)$$

where $x_k \in R^n \forall k$ and $u_k \in R^r \forall k$.

4. Overview of the Solution Procedure

This section presents an overview of the solution procedure for solving the problem of section 2.

1. Given an initial guess for $u_k \forall k$ and for Lagrange multipliers associated with imposed constraints.
2. Calculate a search direction, $\Delta u_k \forall k$, based on a quadratic/linear approximation of OCP at the current point $(u_1, \dots, u_N)$ and update Lagrange multipliers, to be used in the next quadratic/linear sub problem.
3. Perform a line search along the search direction, based on a standard merit function (see Gill et al [15]), thereby obtaining $u_k = u_k + \alpha \Delta u_k$, $(k=1, \dots, N)$ for some $\alpha \in (0, 1]$.
4. Go to step 2.

This loop is repeated to convergence.

6. Recursive Solution to the Quadratic Approximation of the Problem

In this section we will present the sub problem details and the details of the recursive solution procedure that is applied to the sub problem. We are given a quadratic/linear approximation of the discrete-time state equation:

$$\begin{aligned} \Delta x_{k+1,i} \approx & \sum_{j=1}^n \frac{\partial f_{ki}}{\partial x_{kj}} \Delta x_{kj} + \frac{1}{2} \sum_{j=1}^n \sum_{m=1}^n \frac{\partial^2 f_{ki}}{\partial x_{kj} \partial x_{km}} \Delta x_{kj} \Delta x_{km} + \\ & \sum_{j=1}^n \sum_{m=1}^r \frac{\partial^2 f_{ki}}{\partial x_{kj} \partial u_{km}} \Delta x_{kj} \Delta u_{km} + \frac{1}{2} \sum_{j=1}^r \sum_{m=1}^r \frac{\partial^2 f_{ki}}{\partial u_{kj} \partial u_{km}} \Delta u_{kj} \Delta u_{km} + \\ & \sum_{j=1}^r \frac{\partial f_{ki}}{\partial u_{kj}} \Delta u_{kj}, \quad (k=1, \dots, N), \quad (i=1, \dots, n) \end{aligned} \quad (6.1)$$

(Note: f_{ki} denotes the i th (of n) component of the n -vector f_k . This notation convention will be used throughout.) Also, we have the following linear approximation of the imposed constraints, such as the terminal constraints and path constraints (3.3) and (3.4):

$$E_k \Delta x_k + F_k \Delta u_k - w_k = 0, \quad (k=1, \dots, N) \quad (6.2)$$

$$E_{N+1}\Delta x_{N+1} - w_{N+1} = 0 \quad (6.3)$$

Without loss of generality, let the number of constraints in (6.2) and (6.3) be n . And, we have the following quadratic/linear approximation of the cost function:

$$\bar{J} \equiv \sum_{k=1}^{N+1} J_k \quad (6.4)$$

$$\text{where } J_k = \sum_{k=1}^N \left[\frac{1}{2} \Delta u_k^T Q_k \Delta u_k + \frac{1}{2} \Delta x_k^T R_k \Delta x_k + \Delta x_k^T M_k \Delta u_k + c_k^T \Delta x_k + d_k^T \Delta u_k \right],$$

$$(k=1, \dots, N) \quad (6.5)$$

$$\text{and } J_{N+1} = \frac{1}{2} \Delta x_{N+1}^T Q_{N+1} \Delta x_{N+1} + c_{N+1}^T \Delta x_{N+1} \quad (6.6)$$

(where

$$(Q_k)_{ij} \equiv \frac{\partial^2 J_k}{\partial u_{ki} \partial u_{kj}} + \sum_{p=1}^n \lambda_{kp} \frac{\partial^2 g_{kp}}{\partial u_{ki} \partial u_{kj}},$$

$$(R_k)_{ij} \equiv \frac{\partial^2 J_k}{\partial x_{ki} \partial x_{kj}} + \sum_{p=1}^n \lambda_{kp} \frac{\partial^2 g_{kp}}{\partial x_{ki} \partial x_{kj}},$$

$$(M_k)_{ij} \equiv \frac{\partial^2 J_k}{\partial x_{ki} \partial x_{kj}} + \sum_{p=1}^n \lambda_{kp} \frac{\partial^2 g_{kp}}{\partial x_{ki} \partial u_{kj}}, \quad c_{ki} \equiv \frac{\partial J_k}{\partial x_{ki}}, \quad d_{ki} \equiv \frac{\partial J_k}{\partial u_{ki}}.$$

where the λ_{kp} , $(k=1, \dots, N+1)$, $(p=1, \dots, n)$ are Lagrange multipliers associated with the constraints (3.3)-(3.4). A method of updating these multipliers for the next sub problem, consistent with standard sequential quadratic programming, will be given at the end of this section. Also, let

$$z_k \equiv \sum_{j=k}^{N+1} J_j, \quad (k=1, \dots, N+1) \quad (6.7)$$

Let us also define a second cost function $\bar{\Gamma}$:

$$\bar{\Gamma} = \sum_{k=1}^{N+1} \Gamma_k \quad (6.8)$$

$$\text{where } \Gamma_k = \frac{1}{2} (E_k \Delta x_k + F_k \Delta u_k - w_k)^T (E_k \Delta x_k + F_k \Delta u_k - w_k), \quad (k=1, \dots, N)$$

$$(6.9)$$

$$\text{and } \Gamma_{N+1} = \frac{1}{2} (E_{N+1} \Delta x_{N+1} - w_{N+1})^T (E_{N+1} \Delta x_{N+1} - w_{N+1}) \quad (6.10)$$

Also, let

$$y_k \equiv \sum_{j=k}^{N+1} \Gamma_j, \quad (k=1, \dots, N+1) \quad (6.11)$$

In words, $\bar{\Gamma}$ is one half the total constraint violation of all the imposed *linear* constraints. Analogous to section 2, we have two cost functions: $\bar{J}$ and $\bar{\Gamma}$. The goal is to find Δu_k (and the associated Δx_k calculated by $\Delta x_{k+1} = A_k \Delta x_k + B_k \Delta u_k$, $k=1, \dots, N$) that minimize the quadratic/linear part of $\bar{J}$ subject to the requirement that our Δu_k be a minimizer of $\bar{\Gamma}$. This goal is not achievable in general, depending on the Hessians in $\bar{J}$ and the second derivatives in (6.1). When it is not achievable, the Hessians will be modified so that the modified problem does have a Δu_k that achieves the (modified) goal.

With the following definitions of $\hat{Q}_k$, $\hat{R}_k$, $\hat{M}_k$, $\hat{c}_k$, and $\hat{d}_k$,

$$\hat{Q}_k \equiv F_k^T F_k, \hat{R}_k \equiv E_k^T E_k, \hat{M}_k \equiv E_k^T F_k, c_k \equiv -E_k^T w_k, \text{ and } \hat{d}_k \equiv -F_k^T w_k \quad (6.12)$$

the Γ_k can be written as (ignoring constant terms):

$$\Gamma_k = \frac{1}{2} \Delta u_k^T \hat{Q}_k \Delta u_k + \frac{1}{2} \Delta x_k^T \hat{R}_k \Delta x_k + \Delta x_k^T \hat{M}_k \Delta u_k + \hat{c}_k^T \Delta x_k + \hat{d}_k^T \Delta u_k \quad (6.13)$$

$$\Gamma_{N+1} = \frac{1}{2} \Delta x_{N+1}^T \hat{R}_{N+1} \Delta x_{N+1} + \hat{c}_{N+1}^T \Delta x_{N+1} \quad (6.14)$$

For ease in establishing recursive relationships, it will prove convenient to define H_k , L_k , P_k , q_k , $\hat{H}_k$, $\hat{L}_k$, $\hat{P}_k$, $\hat{q}_k$, and $\hat{s}_k$ as follows

$$z_k = \frac{1}{2} \Delta x_k^T H_k \Delta x_k + \Delta x_k^T L_k \Delta u_k + \frac{1}{2} \Delta u_k^T P_k \Delta u_k + q_k^T \Delta x_k + s_k^T \Delta u_k \quad (6.15)$$

$$y_k = \frac{1}{2} \Delta x_k^T \hat{H}_k \Delta x_k + \Delta x_k^T \hat{L}_k \Delta u_k + \frac{1}{2} \Delta u_k^T \hat{P}_k \Delta u_k + \hat{q}_k^T \Delta x_k + \hat{s}_k^T \Delta u_k \quad (6.16)$$

Suppose we have z_{k+1} expressed in terms of Δx_{k+1} as

$$z_{k+1} = \frac{1}{2} \Delta x_{k+1}^T T_{k+1} \Delta x_{k+1} + b_{k+1}^T \Delta x_{k+1} \quad (6.17)$$

If we substitute (6.1) into (6.17) and we retain only quadratic and linear terms in the cost function z_{k+1} , just as in standard sequential quadratic programming, we obtain

$$H_k = A_k^T T_{k+1} A_k + R_{k+1}^s + R_k \quad (6.18)$$

$$P_k = B_k^T T_{k+1} B_k + Q_{k+1}^s + Q_k \quad (6.19)$$

$$L_k = A_k^T T_{k+1} B_k + M_{k+1}^s + M_k \quad (6.20)$$

$$q_k = A_k^T b_{k+1} + c_k \quad (6.21)$$

$$s_k = B_k^T b_{k+1} + d_k \quad (6.22)$$

where

$$(R_{N+1}^s)_{jm} \equiv \sum_{\alpha=1}^n \frac{\partial^2 f_{N\alpha}}{\partial x_{Nj} \partial x_{Nm}} b_{N+1,\alpha} \quad (6.23)$$

$$(M_{N+1}^s)_{jm} \equiv \sum_{\alpha=1}^n \frac{\partial^2 f_{N\alpha}}{\partial x_{Nj} \partial u_{Nm}} b_{N+1,\alpha} \quad (6.24)$$

$$(Q_{N+1}^s)_{jm} \equiv \sum_{\alpha=1}^n \frac{\partial^2 f_{N\alpha}}{\partial u_{Nj} \partial u_{Nm}} b_{N+1,\alpha} \quad (6.25)$$

thus giving us z_k in terms of Δx_k and Δu_k . Similarly, suppose we have y_{k+1} expressed in terms of Δx_{k+1} as

$$y_{k+1} = \frac{1}{2} \Delta x_{k+1}^T \hat{T}_{k+1} \Delta x_{k+1} + \hat{b}_{k+1}^T \Delta x_{k+1} \quad (6.26)$$

Then, by substituting (6.1) into (6.26) while using only the linear parts of all functions for these constraints, just as in standard sequential quadratic programming, we likewise obtain y_k in terms of Δx_k and Δu_k by

$$\hat{H}_k = A_k^T \hat{T}_{k+1} A_k + \hat{R}_k \quad (6.27)$$

$$\hat{P}_k = B_k^T \hat{T}_{k+1} B_k + \hat{Q}_k \quad (6.28)$$

$$\hat{L}_k = A_k^T \hat{T}_{k+1} B_k + \hat{M}_k \quad (6.29)$$

$$\hat{q}_k = A_k^T \hat{b}_{k+1} + \hat{c}_k \quad (6.30)$$

$$\hat{s}_k = B_k^T \hat{b}_{k+1} + \hat{d}_k \quad (6.31)$$

By the method of section 2, minimizing z_k subject to minimizing y_k with respect to Δu_k for any Δx_k , we can now calculate Δu_k in terms of Δx_k as

$$\Delta u_k = G_k \Delta x_k + e_k \quad (6.32)$$

at which point we can now express z_k and y_k in terms of Δx_k as

$$T_k = H_k + L_k G_k + G_k^T L_k^T + G_k^T P_k G_k \quad (6.33)$$

$$b_k = L_k e_k + G_k^T P_k e_k + G_k^T s_k + q_k \quad (6.34)$$

$$\hat{T}_k = \hat{H}_k + \hat{L}_k G_k + G_k^T \hat{L}_k^T + G_k^T \hat{P}_k G_k \quad (6.35)$$

$$\hat{b}_k = \hat{L}_k e_k + G_k^T \hat{P}_k e_k + G_k^T \hat{s}_k + \hat{q}_k \quad (6.36)$$

where T_k , b_k , $\hat{T}_k$, and $\hat{b}_k$ are defined just as in (6.17) and (6.26), leaving us in the same situation we were in with (6.17) and (6.26) but with $k+1 \rightarrow k$. The recursive process for obtaining all the G_k is initiated by

$$T_{N+1} = R_{N+1} \quad (6.37)$$

$$b_{N+1} = c_{N+1} \quad (6.38)$$

$$\hat{T}_{N+1} = E_{N+1}^T E_{N+1} \quad (6.39)$$

$$\hat{b}_{N+1} = -E_{N+1}^T w_{N+1} \quad (6.40)$$

Below we provide a concise summary of the recursive solution procedure for solving the quadratic/linear sub problem:

1. Let $k=N$.
2. Calculate $T_{k+1} = R_{k+1}$, $b_{k+1} = c_{k+1}$.
Calculate $\hat{T}_{k+1} = E_{k+1}^T E_{k+1}$, $\hat{b}_{k+1} = -E_{k+1}^T w_{k+1}$
3. Calculate H_k , P_k , L_k , q_k , and s_k via equations (6.18) through (6.25).
4. Calculate $\hat{H}_k$, $\hat{P}_k$, $\hat{L}_k$, $\hat{q}_k$, and $\hat{s}_k$ via equations (6.27) through (6.31).
5. Calculate G_k and e_k using the results of steps 3. and 4. and the method described in section 2. (Store G_k and e_k .)
6. Calculate T_k , b_k , $\hat{T}_k$, and $\hat{b}_k$ via equations (6.33) through (6.36).
7. Let $k = k - 1$ and go to step 3.

The above procedure is repeated until $k=1$ in step 5, at which point we obtain $\Delta u_1 = e_1$. Then, the values of the Δu_k and Δx_k are obtained via $\Delta x_{k+1} = A_k \Delta x_k + B_k \Delta u_k$, and $\Delta u_k = G_k \Delta x_k + e_k$, ($k=1, \dots, N$). However, if the initial condition is variable and is therefore part of the search variables, we proceed to step 6 with $k=1$ and calculate Δx_1 via the method of section 2, where this is a special case in which there is only one, not two, variables. Then, calculate $\Delta u_1 = G_1 \Delta x_1 + e_1$, and then the rest of the Δx_k and Δu_k are calculated just as indicated above.

Before proceeding to the next quadratic approximation of the OCP, we must update the Lagrange multipliers associated with the imposed constraints (3.3)-(3.4). The values of these multipliers affects the values of the Q_k , M_k , and R_k in the next sub problem. Let L_x^k be the gradient of the sub problem's cost function with respect to Δx_k , and let L_u^k be the gradient of the sub problem's cost function with respect to Δu_k . We have

$$L_x^k = c_k + R_k \Delta x_k + M_k \Delta u_k \quad (6.41)$$

$$L_u^k = d_k + Q_k \Delta u_k + M_k^T \Delta x_k \quad (6.42)$$

The Lagrange multipliers λ_{sp} can be recovered by writing the necessary conditions for optimality of the sub problem. These necessary conditions express that the gradient be a linear combination of the constraint gradients. Letting λ_k^s denote the Lagrange multiplier associated with the k th state equation, these conditions are

$$-L_x^1 + A_1^T \lambda_1^s + E_1^T \lambda_1 = 0 \quad (6.43)$$

$$-L_x^k + A_k^T \lambda_1^s + E_k^T \lambda_k - \lambda_{k-1}^s = 0, (k=2, \dots, N) \quad (6.44)$$

$$-L_x^{N+1} + E_{N+1}^T \lambda_{N+1} - \lambda_N^s = 0 \quad (6.45)$$

$$-L_u^k + B_k^T \lambda_k^s + F_k^T \lambda_k = 0, (k=1, \dots, N) \quad (6.46)$$

A least squares solution to equations (6.43) to (6.46) can be obtained by a variety of methods, one of which is to form the associated banded-matrix normal equations, which can be solved with a standard banded matrix numerical linear algebra routine.

7. Numerical Example

This section will present a numerical example, which will show the expected quadratic convergence of the algorithm presented herein. The dynamic system considered is a three-mass/three-spring system, as might be used to approximately model a fixed-free rod in longitudinal vibration. The springs are nonlinear. The spring forces are given by

$$F_s = -k\delta - a\delta^3 \quad (7.1)$$

where δ is the deflection of the spring, F_s the spring force, and k and a constants. If we let q_1 , q_2 , and q_3 denote the positions of the masses, measured from equilibrium positions, and relative to an inertial frame and let each mass value be m and let all three springs be equivalent and let there be control forces u_1 and u_2 on only the first two masses (closest to the fixed end), then the equations of motion are

$$m\ddot{q}_1 = k(q_2 - q_1) + a(q_2 - q_1)^3 - kq_1 - aq_1^3 + u_1 \quad (7.2)$$

$$m\ddot{q}_2 = k(q_3 - q_2) + a(q_3 - q_2)^3 - k(q_2 - q_1) - a(q_2 - q_1)^3 + u_2 \quad (7.3)$$

$$m\ddot{q}_3 = -k(q_3 - q_2) - a(q_3 - q_2)^3 \quad (7.4)$$

Equations (7.2)-(7.4) can be easily brought to a state space form by defining $x_1 \equiv q_1$, $x_2 \equiv q_2$, $x_3 \equiv q_3$, $x_4 \equiv \dot{q}_1$, $x_5 \equiv \dot{q}_2$ and $x_6 \equiv \dot{q}_3$. An approximate state transition function f (from equation (3.2)) was obtained by Euler integration with a fixed step size.

The system is initially at rest with $q_i(0) = 0.1$ and $\dot{q}_i(0) = 0 \forall i$. The optimal control problem is to minimize $\sum_{k=1}^N u_k^T u_k$ subject to $q_i(t_f) = 0$ and $\dot{q}_i(t_f) = 0 \forall i$, where t_f is the final time.

Numerical results for the case of $q_i(0) = 0.1 \forall i$, $m=1$, $t_f = 2$ seconds, $a = 0.001$, $k = 0.1$, and $N=100$ time steps were obtained by the algorithm presented herein. Table 1 shows how the 2-norm distance to the solution varies with iteration number. This table clearly shows the quadratic convergence of the sequential quadratic programming implementation.

Table 1: Demonstration of the Algorithm's Quadratic Convergence

Iteration Number	2-Norm Distance to Solution
2	31.18
3	2.003×10^{-2}
4	2.072×10^{-5}
5	2.630×10^{-8}
6	1.110×10^{-14}

Figure 1 below shows the input history for u_1 and Figure 2 that for u_2 .

Figure 1. Input 1 (Newtons) versus time (sec.)

Figure 2. Input 2 (Newtons) versus time (sec.)

Figures 3 through 5 illustrate the time histories for the three position coordinates

Figure 3. Position 1 (m) versus time (sec.)

Figure 4. Position 2 (m) versus time (sec.)

Figure 5. Position 3 (m) versus time (sec.)

8. Conclusion

This work presented a multi-objective differential dynamic programming approach to constrained discrete-time optimal control. In the backward sweep of the dynamic programming in the quadratic sub problem, the sub problem input at a stage or time step is solved for in terms of the sub problem state entering that stage so as to minimize the summed immediate and future cost subject to minimizing the summed immediate and future constraint violations, for all such entering states. The method differs from previous dynamic programming methods, which used penalty methods, in that the constraints of the sub problem, which may include terminal constraints and path constraints, are solved exactly if they are solvable; otherwise, their total violation is minimized. Again, the resulting solution of the sub problem is an input history that minimizes the quadratic cost function subject to being a minimizer of the total constraint violation. The expected quadratic convergence of the proposed algorithm was demonstrated on a numerical example.

9. Acknowledgment

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

References

- [1] Wright, S., "Interior Point Methods for Optimal Control of Discrete Time Systems." *Journal of Optimization Theory and Applications*, April 1993, vol. 77, No. 1, pp. 161-187.
- [2] Wright, S., "Structured Interior Point Methods for Optimal Control," Proceedings of the 30th Conference on Decision and Control, Brighton England, Dec., 1991, pp. 1711-1716.
- [3] Ruxton, D., "Differential Dynamic Programming Applied to Continuous Optimal Control Problems with State Variable Inequality Constraints," *Dynamics and Control*, 1993, vol. 3, pp. 175-185.
- [4] Murray, D. and Yakowitz, S., "The Application of Optimal Control Methodology To Nonlinear Programming Problems," *Mathematical Programming*, 1981, pp. 331-447.
- [5] Gill, P, Murray, W., and Wright, M., *Practical Optimization*, New York: Academic Press, 1981.
- [6] Bertsekas, D.P., "Projected Newton Methods for Optimization Problems with Simple Constraints," *Siam Journal of Control and Optimization*, Vol. 20, No. 2, March 1982.
- [7] Gawande, M. and Dunn, J., "A Projected Newton Method in a Cartesian Product of Balls," *Journal of Optimization Theory and Applications*, Vol. 59, No. 1, 1988.
- [8] Bazaraa, S., et al, *Nonlinear Programming, Theory and Algorithms, Second Edition*, New York: John Wiley and Sons, Inc., 1979.
- [9] Dohrmann, C. and Robinett, R., "Efficient Sequential Quadratic Programming Implementations for Equality-Constrained Optimal Control," to appear in *Journal of Optimization Theory and Applications*.
- [10] Ho., Y and Bryson, A., *Applied Optimal Control*, New York: Hemisphere Publishing Corporation, 1975.
- [11] Lewis, F., *Optimal Control*, New York: John Wiley and Sons, Inc., 1986.
- [12] Wie, B. and Sunkel, J., "Minimum-Time Pointing Control of a Two-Link Manipulator," *Journal of Guidance Control and Dynamics*, Vol. 13, No. 5, Sept.-Oct., 1990, pp. 867-873.
- [13] Li, F. and Bainum, P., "Analytic Time-Optimal Control Synthesis of Fourth-Order System and Maneuvers of Flexible Structures," *Journal of Guidance, Control and Dynamics*, Nov.-Dec., 1994, Vol. 17, No. 6, pp. 1171-1178.

- [14] Geering, H. et al, "Time-Optimal Motions of Robots in Assembly Tasks", *IEEE Transactions on Automatic Control*, Vol. AC-31, No. 6, June 1986.
- [15] Gill, P. et al, *Practical Optimization*, New York: Academic Press, 1981.

9 780000 000000

UC Category (19) UC-905, DOE/ER

DOE