Linköping Studies in Science and Technology Dissertations
No. 104

A NEWTON METHOD FOR SOLVING NON-LINEAR OPTIMAL CONTROL PROBLEMS WITH GENERAL CONSTRAINTS

Henrik Jonson

Institutionen för Systemteknik
Linköpings Tekniska Högskola

Linköping 1983

ISBN 91-7372-718-0
ISSN 0345-7524
Printed in Sweden by VTT-Grafiska, Vimmerby 1983

ABSTRACT

Optimal control of general dynamic systems under realistic constraints on input signals and state variables is an important problem area in control theory. Many practical control problems can be formulated as optimization tasks, and this leads to a significant demand for efficient numerical solution algorithms.

Several such algorithms have been developed, and they are typically derived from a dynamic programming view point. In this thesis a different approach is taken. The discrete-time dynamic optimization problem is formulated as a static one, with the inputs as free variables. Newton's approach to solving such a problem with constraints, also known as Wilson's method, is then consistently pursued, and a algorithm is developed that is a true Newton algorithm for the problem, at the same time as the inherent structure is utilized for efficient calculations. An advantage with such an approach is that global and local convergence properties can be studied in a familiar framework.

The algorithm is tested on several examples and comparisons to other algorithms are carried out. These show that the Newton algorithm performs well and is competitive with other methods. It handles state variable constraints in a direct and efficient manner, and its practical convergence properties are robust.

A general algorithm for large scale static problems is also developed in the thesis, and it is tested on a problem with load distribution in an electrical power network.

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to Professor Lennart Ljung and my supervisor Torkel Glad for their guidance and encouragement.

I also thank all my other friends and colleagues at the Automatic Control group in Linköping for a pleasant atmosphere.

Specially

Bengt Bengtsson for proofreading

Ulla Salaneck for her patience when typing this manuscript,

Marianne Anse-Lundberg for drawing the figures.

My parents, relatives and friends for understanding and support.

This research has been supported by the Swedish Institute of Applied Mathemethics (ITM), which is hereby acknowledged.

CONTENTS

1. INTRODUCTION

Optimal control problems arise in many fields in industry, engineering and science. The theory of optimal control has seen a considerable development over the past thirty years, highlighted by Bellman's Dynamic Programming Bellman, (1957) and Pontryagin's Maximum Principle, Pontryagin et al, (1962). This theory has mostly been formulated for continuous-time problems, but paralleling results for discrete-time systems have also been given (for a survey of these results, see Bryson and Ho (1969) and Jacobson and Mayne (1970).

In view of the practical significance of optimal control, an equally important development of algorithms for solving optimal control problems has taken place. Important such algorithms have been presented, e.g. in Jacobson and Mayne (1970), Polak (1973) Bryson and Ho (1969), Ohno (1978) among others.

These algorithms are mostly developed for continuous time problems and have their roots in the Bellman equation. Differential Dynamic Programming (DDP) is the perhaps best known algorithm of this kind.

The "classical" algorithms are typically capable of handling terminal constraints, and mixed input and state constraints, while pure state constraints have proved more difficult to cope with. Approximate techniques, such as penalty functions and Mårtensson's (1973) constraining hyperplane techniques have, however, been developed. Similar algorithms for discrete-time systems have been described in Jacobson and Mayne (1970) and Canon, Callum and Polak (1970).

In this thesis we shall consider algorithms for optimal control of discrete-time systems and for optimal sampled-data control of continuous-time systems. The guiding idea of the thesis is to regard the optimal control problem as a pure non-linear control problem rather than as a solution to the Bellman partial difference equation. This point of view gives a number of advantages:

- The whole, well-developed field of non-linear programming becomes available for direct applications. We set out to develop a true Newton algorithm for the problem in such a framework.

- Convergence, both local and global, can be investigated within a familiar and well-understood framework.

- Pure state constraints no longer lead to special problems. Constraints of all kinds can be handled with common techniques.

A drawback of the method is however that the number of these constraints must not be too large. Therefore an alternative method for many simple constraints is also discussed.

Our method is actually a Newton method applied to the Kuhn-Tucker necessary conditions for the problem; solved iteratively. At each iteration a quadratic sub-problem with linear constraints is constructed. How this construction is done is shown in chapter 4. In chapter 5 we give sufficient conditions for this subproblem to have a unique solution and also an algorithm for solving this sub-problem. In chapter 6 we discuss methods for modifying the problem if the sufficient conditions are not satisfied. In chapter 7 the convergence of our algorithm is investigated and in section 8 we give a method to handle the case of many simple constraints on the controls. A summary of the algorithm and a brief description of the computer programs based on this algorithm is given in chaper 9. In chapter 11, we demonstrate the algorithm with some examples. In the appendix we give the necessary equations when we use discrete time control on a continuous time system.

In chapter 2 we derive how Newton's method is applied to non-linear constrained programming problems. This results in a constrained quadratic programming problem and this problem is sparse if it for instance originates from an optimal control problem. Here, sparsity means that the Hessian of the Lagrangian and the Jacobian of the constraints contain a high proportion of zero ele

ments. A method for solving such sparse linearly constrained quadratic programming problems is given in chapter 3. A computer program designed for solving nonlinear constrained programming problems is described in section 9.5, and in chapter 10, we use this program for solving an optimal electrical power distribution problem.

1.1 Notations

In this thesis we assume that all vectors are column vectors, except derivatives which are row vectors. Derivatives are indicated by subscripts. We also use the following notations or conventions when nothing else is indicated

i,j	indices of vector elements or matrix elements
k	iteration counter. This index will usually be suppressed for the subproblems.
f(x(t),u(t),t)	the transition function for the discrete time system
f(t)	short notation for f(x(t),u(t),t). When we use this notation we assume that u(p), p=0,...,N-1 is a certain control sequence and x(p),p=0,..,N are the corresponding states
g(z)	equality constraints for a nonlinear programming problem
h(z)	inequality constraints for a nonlinear programming problem
$h^i(t_i)$	inequality constraint for the optimal control problem evaluated at $(x(t_i),u(t_i))$
$\ell(z)$	objective function for a nonlinear programming problem
$\ell(t)$	cost function in the optimal control problem evaluated at (x(t),u(t)) or (x(N))

$J(u)$	The total cost when the control sequence u is used.
n_x	the number of states
n_u	the number of controls
p	integer, usually being the number of inequality constraints sometimes also used as time index
s	time variable both for discrete time and continuous time
t	integer, time index
τ	time variable
$u(t)$	control varaible
u	the sequence $u(t);t=0,\ldots,N-1$ and sometimes also the corresponding states (e.g. $J(u)$)
$V(t)$	the cost generated when starting at state $x(t)$ and using a given control sequence $u(p);p=t,\ldots,N-1$
$W(t)$	the second order Taylor expansion of $V(t)$ around the point $(x(t),z(t))$
$x(t)$	the state variable
z	vector of variables in a nonlinear programming problem
$z(t)$	the controls $(u^T(N-1),\ldots,u^T(t))^T$
$z(0)$	same as u

λ	Lagrange multiplier for equality constraints
$\lambda(t)$	Lagrange multipliers for the optimal control problem corresponding to the transition function
μ	Lagrange multiplier for inequality constraints
$(\mu_k)_i$	the i:th component of the Lagrange multiplier μ at iteration k.
$\gamma(t)$	$\sum_{i\in I(t)} (\mu_k)_i h^i(t)$
$I(t)$	$\{i:t_i=t\}$. Indices for the inequality constraints at time t
$v_x(t), \ell_x(t), f_x(t)$	when x,u,z appear as subscripts it means the derivative of the function with respect to the varaible
$\lvert\cdot\rvert$	the usual Euclidian norm
$\lvert\Delta u\rvert$	$\left(\sum_{t=0}^{N-1} \sum_{i=1}^{n_u} \Delta u_i^2(t)\right)^{1/2}$
$V_x(t+1) f_{xx}(t)$	$\sum_{i=1}^{n_x} \cdot \frac{\partial V(t+1)}{\partial x_i(t+1)} \cdot f_{xx}^i((t),u(t),t)$

2. OPTIMIZATION PROBLEMS AND OPTIMIZATION ALGORITHMS

A large number of questions in science and decision-making lead to optimization problems. To get a flavour of typical situations where an optimization formulation is natural, let us consider two simple examples.

Example 2.1. A simple electric power-network problem.

Consider the problem of supplying two consumers with power from two different generators. The generators and the consumers are connected in a network as shown in figure 2.1

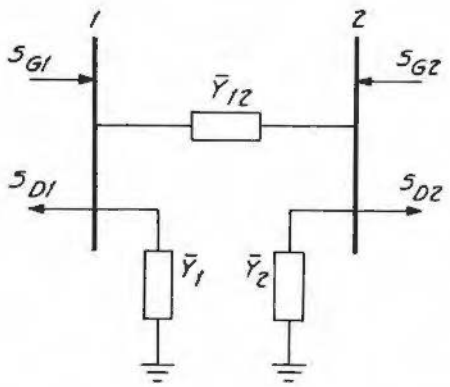


Fig. 2.1. An electrical network with two nodes and one line.

Let us introduce the following notation

S_{Gi}: Power generated at node i, i=1,2
S_{Di}: Power demand at node i, i = 1,2
$\bar{Y}_i$: Conductance between node i and earth, i =1,2
$\bar{Y}_{12}$: Conductance between nodes 1 and 2
$C_i(S_{Gi})$: The cost of generating the power S_{Gi} at node i, i=1,2
$\bar{V}_i$ Voltage at node i, i=1,2.

It is customary to describe the electrical quantities using

complex numbers; the power is thus split into a real and an reactive part in the following way:

$$S_{G1}=P_{G1}+iQ_{G1}$$

Here P_{G1} is the active power and Q_{G1} is the reactive power. Similar expressions also hold for S_{G2}, S_{D1} and S_{D2}.

The complex voltages and conductances can in an analogous way be represented using amplitude (V_1) and phase angle (φ_1), i.e. $\bar{V}_1=V_1 \cdot e^{i\varphi 1}$. For $\bar{V}_2$ a similar representation can be introduced; the conductances will be represented analogously, using δ_i for the corresponding phase angles.

The problem is to satisfy the power demands at the two nodes, and at the same time generate the power as cheaply as possible. The latter condition is expressed as to minimize a criterion

$$C_1(S_{G1})+C_2(S_{G2}). \tag{2.1}$$

There are also a number of physical constraints associated with this problem. At each node there must be a power balance. This means that the following constraints must be met at node 1:

$$0=S_{G1}-S_{D1}-\bar{V}_1\ (\bar{V}_1\cdot\bar{Y}_1)^*-\bar{V}_1((\bar{V}_1-\bar{V}_2)\ \bar{Y}_{12})^*. \tag{2.2}$$

where * means complex conjugate.

Similarly for node 2 we have

$$0=S_{G2}-S_{D2}-\bar{V}_2(\bar{V}_2\cdot\bar{Y}_2)^*-\bar{V}_2\cdot((\bar{V}_2-\bar{V}_1)\cdot\bar{Y}_{12})^*. \tag{2.3}$$

Also, the capacities of the generators are limited, which means that constraints of the following type

$$\begin{matrix} P_{\ell i} \leq P_{Gi} \leq P_{ui} \\ Q_{\ell i} \leq Q_{Gi} \leq Q_{ui} \end{matrix} \qquad i=1,2 \tag{2.4}$$

must be included. Similarly, the voltage amplitudes must lie within certain bounds:

$$V_{\ell} \leq V_i \leq V_u \qquad i=1,2 \quad . \tag{2.5}$$

The requirement that the network must be stable corresponds to a condition on the difference between the phase angles:

$$|\varphi_1 - \varphi_2| < \frac{\pi}{2} \quad . \tag{2.6}$$

Finally, we must introduce a constraint on the heat developed in the line, assuring that it does not melt down. Such a condition corresponds to

$$P_{max} \geq Re[(\bar{V}_1 - \bar{V}_2)(\bar{V}_1 - \bar{V}_2)\ \bar{Y}_{12})^*] = |\bar{V}_1 - \bar{V}_2|^2\ Y_{12} \cos \delta_{12}. \tag{2.7}$$

The problem of generating power in the simple network in a sensible way has now been formalized as an optimization problem, namely to minimize (2.1) subject to the constraints (2.2)-(2.7).

□

Example 2.2. Housing maintenance.

(This example is adapted from Luenberger (1979), pp 433). Consider a rental house, and suppose that its state can be characterized by a single quality measure x(k) at time k. A simple model for how this quality measure develops can then be given as

$$x(k+1) = \alpha x(k) + u(k) - \frac{u^2(k)}{\bar{x} - x(k)} \qquad k=0, \ \ldots, \ N-1 \tag{2.8}$$

here the number α is subject to $0<\alpha<1$, describing the decay of the house with time. The maintenance expenditure during time period k is u(k) and the value $x=\bar{x}>0$ corresponds to "perfect conditions".

The rent of the house is supposed to be proportional to the quality measure x. The problem we consider, is that of a landlord who wishes to determine a maintenance policy, so as to maximize a discounted net profit up to a certain time instance N, when he plans to sell the house. Formally, he wishes to maximize

$$J=\beta^N C\ x(N)+\sum_{k=0}^{N-1}(px(k)-u(k))\beta^k \tag{2.9}$$

where $p>0$, $0<\beta<1$. The quantity $C\ x(N)$ is the sales price at time N and the quantity $p\ x(k)$ is the rental income. At the beginning of the time period considered, the quality of the house is supposed to be

$$x(0)=x_o. \tag{2.10}$$

The optimization problem is then to find the sequence $u(0),\ldots,u(N-1)$ such that the expression (2.9) is maximized when $x(k)$, $k=0,\ldots,N$ satisfies the constraints (2.8) and (2.10).

□

The problems considered in example 2.1 and 2.2 are quite different. Still, they could both be described by the following formulation

$$\text{minimize} \quad \ell(z) \qquad z\in R^n \tag{2.11a}$$

$$\text{subject to} \quad g(z)=0 \qquad g\in R^{m_1} \tag{2.11b}$$

$$h(z)\leq 0 \qquad h\in R^{m_2} \tag{2.11c}$$

In example 2.1 we could let the vector z consist of the parameters V_1, δ_1, V_2, δ_2, P_{G1}, Q_{G1}, P_{G2} and Q_{G2}. The functional ℓ then corresponds to the functions C_1 and C_2 in (2.1). The equality constraints g correspond to equations (2.2) and (2.3), while the inequality constraints h in (2.11c) have their counterparts in the relations (2.4)-(2.7).

In example 2.2 the vector z consists of the states x(t), t=0,..., as well as the control variables u(t), t=0,.., N-1. The function ℓ (which often will be called the objective function) is defined by (2.9) and the equality constraints correspond to (2.8) and (2.10) In this example there are no inequality constraints.

In example 2.2 it is possible to arrive at a slightly different formulation by eliminating the states x(k) in (2.9) using equations (2.8) and (2.10). These two equations define the sequence x(k), k=0,..., N uniquely from the sequence u(k), k=0,..., N-1. Then J in (2.9) will be a function of the control signals u(k), k=0,..., N-1 only. In this example (which is a simple case of an optimal control problem) we thus have the option of considering both u and x as free variables, subject to a constraint (corresponding to (2.8)) or to eliminate the intermediate variables x, and let the vector z in (2.11) consist of the control signals only. In this thesis we shall work with both these variants, choosing one or the other depending on the situation.

The problem (2.11) is the standard formulation of the general nonlinear programming problem. Many algorithms have been proposed for solving this problem, and considerable efforts have been made to find efficient algorithms for a variety of situations. What constitutes an efficient algorithm for a particular case, will be highly dependent on the structure and the complexity of the functions ℓ, g and h. Important properties are, for instance, whether these functions are linear or nonlinear and if they are differentiable or not. In this thesis we shall assume throughout that the functions involved are at least twice continuously differentiable and not necessarily linear. We also assume that all second derivatives are Lipschitz continuous.

For unconstrained problems, i.e. problems where (2.11bc) do not apply, the most common methods are

Steepest Descent Method
Newtons's Method
Conjugate Direction Method
Quasi-Newton Methods.

(See, for example, Luenberger (1973) and Fletcher (1980)). All these methods generate a sequence of points, z_k, which under suitable conditions converges to a local minimum of the criterion (2.11a). This sequence of points is then generated as

$$z_{k+1} = z_k + \alpha_k \cdot d_k \qquad (2.12)$$

where $0 < \alpha_k \leq 1$ is chosen so that a suitable decrease is obtained in the objective function (2.11a). For the Steepest Descent Method, the vector d_k in (2.12) is the negative gradient, i.e.

$$d_k = - \ell_z^T(z_k).$$

For Conjugate Direction Methods, d_k is chosen as a certain linear combination of the gradient $\ell_z^T(z_k)$ and the previous direction d_{k-1}. In Newton's Method, d_k is chosen as the solution of

$$H(z_k)d_k = -\ell_z^T$$

where $H(z_k)$ is the Hessian of the objective function, i.e.

$$H(z_k) = \ell_{zz}(z_k).$$

The calculation of $H(z_k)$ and d_k may be computationally costly. Therefore some methods use

$$H_k d_k = -\ell_z^T(z_k)$$

where H_k is an approximation of $H(z_k)$. This approximation is changed at each step. Such methods are called Quasi-Newton Methods.

For constrained programming problems (i.e. problems where (2.11bc) apply), well known methods are (see Luenberger (1973) and Fletcher (1981)):
Penalty and Barrier Methods
The Reduced Gradient Method

Augmented Lagrangian Methods
Feasible Direction Methods
Wilson's Method (Wilson, 1963)
Quasi-Newton versions of Wilsons Method (Powell (1978), Han (1975, 1977), and Chamberlain et al. (1979)).

In this thesis we shall concentrate on algorithms developed from Wilson's Method. (Described for instance in Bertsekas (1982) pp. 252-256.) This method was proposed in 1963. Since second order derivatives are required in this algorithm, it is rather time consuming. Therefore it is not very useful. For sparse problems (such as those we are to encounter in the later chapters) the computational work may be reasonable. We shall here give a short motivation and description of Wilson's Mehtod. First, we assume that there are no inequality constraints, i.e. the problem is defined by (2.11ab). The Lagrangian of the problem is then given by

$$L(z,\lambda)=\ell(z)+\lambda^T g(z). \tag{2.13}$$

The Kuhn-Tucker necessary conditions (see p. 242 in Luenberger (1973)) then state that if z^* is the solution of (2.11) then there exists a multiplier λ^* such that

$$L_z^T(z^*,\lambda^*)=\ell_z^T(z^*)+g_z^T(z^*)\lambda^*=0 \tag{2.14a}$$

$$g(z^*)=0 \tag{2.14b}$$

The relations (2.14) form a system of nonlinear equations in the variables z and λ. Assume that we have a good estimate (z_k,λ_k) of the solution to (2.14). This estimate could then be improved using Newton-Raphson's method (see Dahlquist and Björck (1974), pp. 249). Then a new estimate (z_{k+1}, λ_{k+1}) to the solution of (2.14) is constructed as follows:

$$L_z^T(z_k,\lambda_k)+L_{zz}(z_k,\lambda_k)(z_{k+1}-z_k)+L_{z\lambda}(z_k,\lambda_k)(\lambda_{k+1}-\lambda_k)=0$$

$$g(z_k)+g_z(z_k)(z_{k+1}-z_k)=0 \ .$$

Noting that

$$L_z^T(z_k,\lambda_k)+L_{z\lambda}(z_k,\lambda_k)(\lambda_{k+1}-\lambda_k)=\ell_z^T(z_k)+g_z^T(z_k)\lambda_{k+1}$$

we can write the equations as

$$\ell_z^T(z_k)+L_{zz}(z_k,\lambda_k)(z_{k+1}-z_k)+g_z^T(z_k)\lambda_{k+1}=0$$

$$g(z_k)+g_z(z_k)(z_{k+1}-z_k)=0.$$

The above equations can however be interpreted as Kuhn-Tucker necessary conditions for the following optimization problem:

$$\underset{z_{k+1}}{\text{minimize}} \quad \ell_z(z_k)(z_{k+1}-z_k)+\frac{1}{2}(z_{k+1}-z_k)^T L_{zz}(z_k,\lambda_k)(z_{k+1}-z_k)$$

$$\text{subject to } g(z_k)+g_z(z_k)(z_{k+1}-z_k)=0$$

The above equation is a quadratic minimization problem with linear constraints for determining z_{k+1} from z_k, λ_k. If we had included inequality constraints (2.11c) we would similarly have been led to the problem

$$\underset{d_k}{\text{minimize}} \quad \ell_z(z_k)d_k+\frac{1}{2}d_k^T L_{zz}(z_k,\lambda_k,\mu_k)d_k \tag{2.15a}$$

$$\text{subject to } g(z_k)+g_z(z_k)d_k=0 \tag{2.15b}$$

$$h(z_k)+h_z(z_k)d_k\leq 0. \tag{2.15c}$$

Here μ_k are the Lagrangian multipliers to the inequality constraints. With d_k determined from (2.15) we then calculate z_{k+1} using (2.12). Since the described calculations consitute Newton-Raphson steps for solving (2.14), the sequence z_k will converge quadratically to z^* locally, provided $\alpha_k=1$ in (2.12). To assure global convergence, it is sometimes necessary to let the step length parameter α_k be less than unity. Cham-

berlain et al. (1979) give a useful way to calculate α_k, which guarantees both global convergence and fast local convergence under suitable conditions.

In quasi-Newton versions of Wilson's method, the Hessian of the Lagrangian in (2.15a), i.e. $L_{zz}(z_k,\lambda_k,\mu_k)$, is replaced by an approximation H_k.

When the non-linear optimization problem (2.11) is generated by a discrete-time optimal control problem, such as the one in example 2.2, special methods have been developed to utilize the particular structure in question. Methods particularly designed for solving discrete-time optimal control problems can be found in Bryson and Ho (1969), Dyer and McReynolds (1970), Jacobson and Mayne (1970), Bellman and Dreyfus (1962) and Ohno (1978).

We shall in this thesis demonstrate how Wilson's method can be adapted to take the special structure of discrete time-optimal control problems into account.

For large problems, that is when the number of elements in z is more than, say, 100, the choice of algorithm for solving the problem (2.11) is very important indeed. The algorithm must converge fast and it must be numerically stable. For such problems it is necessary to take the particular structure of the problem into account so that suitable decomposition techniques can be applied (Lasdon 1970).

A common example of such large problems is real-life power network problems (such as large network variants of example 2.1). These networks usually have more than 100 nodes, which may lead to more than 500 unknown parameters. Optimal control problems similarly lead to large optimization problems if the number of time points is large and there are several state variables and control signals.

3. SOLVING SPARSE QUADRATIC SUBPROBLEMS

3.1 Introduction

In chapter 2 we discussed the algorithm known as Wilson's Method. In that algorithm one must solve quadratic problems of the type

$$\min_{z} \quad b^T z+\frac{1}{2} z^T B z , \tag{3.1a}$$

$$\text{subject to} \quad g+Gz=0 , \tag{3.1b}$$

$$h+Hz\leq 0 , \tag{3.1c}$$

where z,b,g and h are column vectors of dimensions n,n,m_1 and m_2 respectivley, while B,G and H are matrices of appropriate dimensions, with B symmetric. In order to have a unique solution of (3.1) and to ensure that the algorithm will find this solution, we make the following assumptions about the matrices B, G and H.

A1: The matrix B is positive definite on the null space of G, i.e. $\exists\ \alpha>0$:
$Gz=0 \Rightarrow z^T Bz\geq\alpha||z||^2$.

A2: The rows of G and H are linearly independent, i.e. the only solution (λ,μ) of

$$G^T\lambda+H^T\mu=0$$

is $\lambda=0$ and $\mu=0$.

Several methods for solving problem (3.1) have been given (see e.g. Gill and Murray (1978)) but few of them can utilize a sparse structure of the matrices B, G and H.

3.2 An algorithm for solving quadratic problems

In this section we will give the basic steps of an algorithm proposed by Goldfarb (1975), which can utilize sparsity in the matrices B, G and H. The Lagrangian of the problem (3.1) is

$$L(z,\lambda,\mu)=b^Tz+\frac{1}{2}\,z^TBz+\lambda^T(g+Gz)+\mu^T(h+Hz)\ , \tag{3.2}$$

and the Kuhn-Tucker necessary conditions are

$$b+Bz+G^T\lambda+H^T\mu=0\ , \tag{3.3a}$$

$$g+Gz=0\ , \tag{3.3b}$$

$$h+Hz\leq 0\ , \tag{3.3c}$$

$$\mu\geq 0 \text{ and} \tag{3.3d}$$

$$\mu^T(h+Hz)=0\ . \tag{3.3e}$$

(cf pp. 233 in Luenberger (1973)).

Lemma 3.1: If the matrices B, G and H satisfy assumptions A1 and A2, then there exists a unique point (z,λ,μ) that satisfies the conditions (3.3a-e).

Proof: Existence: Because of A2 we know that there exists at least one point that satisfies the constraints (3.1b) and (3.1c). Because of A1 there is then a solution of (3.1). For this solution the theorem on page 233 in Luenberger (1973) guarantees the existence of multipliers λ and μ such that the conditions (3.3) are satisfied.

Uniqueness: Assume that we have two different points, (z_1,λ_1,μ_1) and (z_2,λ_2,μ_2), that satisfy the conditions (3.3). If $z_1=z_2$, then assumption A2 and the condition (3.3a) imply that $\lambda_1=\lambda_2$ and $\mu_1=\mu_2$ so we can assume that $z_1\neq z_2$. From (3.3b) we get $G(z_1-z_2)=0$. Hence z_1-z_2 is in the null space of G. Similarly (3.3a) gives

$$B(z_1-z_2)+G^T(\lambda_1-\lambda_2)+H^T(\mu_1-\mu_2)=0.$$

Using this together with A1 gives

$$0<(z_1-z_2)^TB(z_1-z_2)=-(\lambda_1-\lambda_2)^TG(z_1-z_2)-(\mu_1-\mu_2)^TH(z_1-z_2)=$$

$$=-(\mu_1-\mu_2)^T((h+Hz_1)-(h+Hz_2))=\mu_1^T(-h+Hz_2)+\mu_2^T(h+Hz_1).$$

In the last equality we used the conditions (3.3e) for z_1 and z_2. The above expression contradicts the conditions (3.3c) and (3.3d), which proves that we have at most one point that satisfies the conditions (3.3).

□

We will now consider ways of solving the problem defined by (3.1a-c). Let the i:th element of h be denoted h_i and let the i:th row of H be denoted H_i. Let J be a given set of distinct positive integers j (J is the set of supposed active constraints), such that $j\leq m_2$ and define $\tilde{h}$ and $\tilde{H}$ as

$$\left.\begin{aligned}\tilde{h}&=(h_{j1},\ldots,h_{jp})^T\\ \tilde{H}&=(H_{j1}^T,\ldots,H_{jp}^T)^T\end{aligned}\right\}\quad j_i\in J,\ i=1,\ldots,p'$$

A way of finding the solution of (3.3) is given by the following algorithm.

I. Let z_0 be an initial approximation of the solution. Let J consist of the integers j such that $h_j+H_jz_0>0$. Let k=0 and go to step II.

II. Solve the system

$$\begin{pmatrix} B & G^T & \tilde{H}^T \\ G & 0 & 0 \\ \tilde{H} & 0 & 0 \end{pmatrix} \begin{pmatrix} z \\ \lambda \\ \mu \end{pmatrix} = \begin{pmatrix} -b \\ -g \\ -\tilde{h} \end{pmatrix} \qquad (3.4)$$

Denote the solution by z', λ', $\tilde{\mu}'$

III. If $h+Hz' \leq 0$ then go to step IV, otherwise let α be the largest number, such that $h_j+H_j(z_k+\alpha(z'-z_k)) \leq 0$ for all integers j that are not yet in J. Add one integer j, for which $h_j+H_j(z_k+\alpha(z'-z_k))=0$, to the index set J. Put $z_{k+1}=z_k+\alpha(z'-z_k)$. Put $k=k+1$ and go back to step II.

IV. Put $z_{k+1}=z'$, $k=k+1$.
If $\tilde{\mu}_j' \geq 0$ for all j then go to V, else delete the index j from J for which $\tilde{\mu}_j'$ is most negative, and to back to step I.

V. z_k is the solution.

This way to choose active constraints is basically the same as that given in Gill and Murray (1978) and Powell (1981). Note that we do not start to delete any constraints from the active set before we have found a point z_k that satisfies $h+Hz_k \leq 0$. After that, every generated point will satisfy this constraint and the algorithm will find the solution to (3.3) after a finite number of steps according to section 7 of Gill and Murray (1978). If the algorithm fails to find a solution, that is if the algorithm starts cycling or the matrix in (3.4) becomes singular, then the constraints are either linearly dependent or the matrix B is not positive definite on the null space of 6. Examples of methods for solving the system (3.4) can be found in section 5.3 in Bartels et al. (1970), in Bunch and Kaufman (1980) and in Paige and Sanders (1975).

3.3 How to utilize the sparsity

If we have an initial value z_0, that generates a correct or almost correct set of active constraints, then we usually need

to solve the system (3.4) only a few times. This is, however, not always the case. If n is large, it is very expensive to solve (3.4) in a straightforward way every time. Only one column and one row of the system matrix in (3.4) is included or removed from one iteration to another. Therefore, we ought to use the previous solution when calculating the next. We cannot use the technique proposed by Powell (1981), if we want to utilize the sparsity of the problem. Assume, however, that the solution $(z', \lambda', \tilde{\mu}')$ of (3.4) can be written in the following way

$$z'=\bar{z}+Z\,\tilde{\mu}'$$
$$\lambda'=\bar{\lambda}+\Lambda\,\tilde{\mu}', \tag{3.5}$$

where Z and Λ are matrices of proper dimensions, and $\bar{z}$ and $\bar{\lambda}$ are solutions of

$$\begin{pmatrix} B & G^T \\ G & 0 \end{pmatrix} \begin{pmatrix} \bar{z} \\ \bar{\lambda} \end{pmatrix} = \begin{pmatrix} -b \\ -g \end{pmatrix} \tag{3.6}$$

Then Z and Λ are given as the solution of

$$\begin{pmatrix} B & G^T \\ G & 0 \end{pmatrix} \begin{pmatrix} Z \\ \Lambda \end{pmatrix} = \begin{pmatrix} -\tilde{H}^T \\ 0 \end{pmatrix}, \tag{3.7}$$

and $\tilde{\mu}'$ is given by the solution of

$$A\tilde{\mu}'=d\ , \tag{3.8}$$

where

$$A=-\tilde{H}Z\ , \tag{3.9}$$

and

$$d=\tilde{h}+\tilde{H}\bar{z}\ . \tag{3.10}$$

The matrix A is symmetric and positive definite if the assumptions A1 and A2 hold, as will be shown in section 3.5. However, A is usually not sparse.

The systems (3.6) and (3.7) could be solved with the methods listed at the end of section 2, but the choice here depends very much on the type of sparsity in the matrices B and G. Since we solve systems with the same left hand side system matrix several times, the method should be of a factorization type. We thus assume that we have an algorithm that can factorize the left hand side matrix of (3.6) efficiently by utilizing the sparsity of B and G.

The idea is now to utilize the fact that the system matrices are the same in (3.6) and (3.7). One may therefore use the same factorization when calculating Z and Λ as when calculating $\bar{z}$ and $\bar{\lambda}$. We then get the following modification of the algorithm given in section 3.2:

I. Find a factorization of the left hand side matrix of (3.6), with a routine for sparse factorization. Find the vectors $\bar{z}$ and $\bar{\lambda}$ defined from (3.6). The rest of this step is the same as step I in the algorithm given in section 3.2.

II. Find Z and A defined by (3.7), using the factorization obtained in step I. Calculate A and d defined by (3.9) and (3.10). Find $\tilde{\mu}'$ by solving (3.8), and calculate z' and λ' using (3.5).

III. Same as step III in section 3.2.
IV. Same as step IV in section 3.2.
V. Same as step V in section 3.2.

If we add a constraint to the active set J in step III, we only need to calculate one new column in Z and A. The other columns are known from earlier steps. In step IV, if we delete one constraint from the active set, we only need to delete the corresponding column in the current matrices Z and A. This also applies to the matrix A and the vector d. How to utilize this last fact is described in the next section.

Remark 3.1. $\bar{z}$ in (3.6) is the solution of problem (3.1) when the inequality constraints are ignored and the i:th column of A is generated by the solution of the problem

$$\underset{z}{\text{minimize}} \quad -H_i z + \frac{1}{2} z^T B z$$

$$Gz = 0$$

This will be utilized in Section 5.5.

Remark 3.2. When we have found the solution $\tilde{\mu}'$ to (3.8) we do not necessarily have to use (3.5) to obtain z' and λ'. From (3.4) we see that z' and λ' can be obtained as the solution of

$$\begin{pmatrix} B & G^T \\ G & 0 \end{pmatrix} \begin{pmatrix} z \\ \lambda \end{pmatrix} = - \begin{pmatrix} b+\tilde{H}^T\tilde{\mu} \\ g \end{pmatrix} . \tag{3.11}$$

Thus if we want to save computer memory, we can find the solution of (3.4) without storing the matrices Z and A. Instead we have to solve the system (3.11) to get the values of z' and λ'.

Remark 3.3. Since the matrix A in (3.8) is positive definite, we can solve equation (3.5) using a conjugate gradient method. Using this method it is not necessary to store the matrix A explicitly.

3.4 Updating algorithms for the A matrix.

In step II of the algorithm given in section 3.3 we solve equation (3.8). Between iterations, only one row and one column are inserted into or deleted from the matrix A. All the other elements are unchanged. We now show how the LDL^T-factorization of A is updated from one iteration to the next.

A constraint is added to the active set.

Let A be the previous matrix and $\bar{A}$ be the new matrix. Assume

also that the new elements are stored in the last row and last column of $\bar{A}$, i.e.

$$\bar{A} = \begin{pmatrix} A & a \\ a^T & \alpha \end{pmatrix} ,$$

where a and α are the new elements with a a vector and α a scalar.
Assume also that we have the LDL^T-factorization of A, i.e.

$$LDLT=A \quad .$$

Then

$$\bar{L}\bar{D}\bar{L}^T = \bar{A} ,$$

where

$$\bar{L}=\begin{pmatrix} L & 0 \\ \ell^T & 1 \end{pmatrix} \; ; \quad \bar{D} = \begin{pmatrix} D & 0 \\ 0 & d \end{pmatrix} \quad ,$$

and

$$\ell = (LD)^{-1} a$$

$$d = \alpha - \ell^T D \ell \quad .$$

A constraint is deleted from the active set.

Assume that we delete row k and column k from A and get $\bar{A}$, i.e.

$$A = \begin{pmatrix} A_{11} & a_1 & A_{12} \\ a_1^T & \alpha & a_2^T \\ A_{21} & a_2 & A_{22} \end{pmatrix} \; ; \quad \bar{A} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \quad .$$

We also have

$$LDL^T = A,$$

where

$$L = \begin{pmatrix} L_{11} & 0 & 0 \\ \ell_1^T & 1 & 0 \\ L_{21} & \ell_2 & L_{22} \end{pmatrix} \quad ; \quad D = \begin{pmatrix} D_1 & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & D_2 \end{pmatrix} ,$$

We want to find matrices $\bar{L}$, $\bar{D}$ such that $\bar{L}\bar{D}\bar{L}^T = \bar{A}$. Then $\bar{L}$ and $\bar{D}$ are given by

$$\bar{L} = \begin{pmatrix} L_{11} & 0 \\ L_{21} & \bar{L}_{22} \end{pmatrix} \quad ; \quad \bar{D} = \begin{pmatrix} D_1 & 0 \\ 0 & \bar{D}_2 \end{pmatrix} ,$$

where $\bar{L}_{22}$ and $\bar{D}_2$ are given by

$$\bar{L}_{22}\,\bar{D}_2\,\bar{L}_{22}^T = L_{22}\,D_2\,L_{22}^T + d\,\ell_2\,\ell_2^T \quad . \tag{3.12}$$

How to update the factorization (3.12) is shown in Gill et al. (1974)

From p. 516 in Gill et al. (1974) we have, that given L_{22}, D_2, d, and L_2, then $\bar{L}_{22}$ and $\bar{D}_2$ in (3.12) could be calculated by the following algorithm:

1. Define $\alpha_1 = d$, $w^{(1)} = l_2$.

2. For j=1, 2,..., n compute

$$p_j = w_j^{(j)} ,$$

$$\bar{d}_j = d_j + \alpha_j p_j^2 ,$$

$$\beta_j = p_j \alpha_j / \bar{d}_j ,$$

$$\alpha_{j+1} = d_j \alpha_j / \bar{d}_j ,$$

$$w_r^{(j+1)} = w_r^{(j)} - p_j / L_{rj},$$

$$\bar{L}_{rj} = L_{rj} + \beta_j w_r^{(j+1)} \quad , \; r=j+1,\ldots,n$$

where d_j are the diagonal elements in D_2 and $\bar{d}_j$ are the diagonal elements in $\bar{D}_2$. L_{rj} are the elements in L_{22} and $\bar{L}_{rj}$ are the elements in $\bar{L}_{22}$.

3.5 Positive definiteness of the A matrix

Theorem 3.1. If z is linearly independent of the columns in G^T and if assumptions A1 and A2 hold, then

$$(z^T, 0) \begin{pmatrix} B & G^T \\ G & 0 \end{pmatrix}^{-1} \begin{pmatrix} z \\ 0 \end{pmatrix} > 0$$

□

In order to prove this theorem we need some lemmas. First we introduce the following notation

$V(A)=\{x: \; x=Az \text{ for some } z\}$ (the range space of A)

$N(A)=\{y: Ay=0\}$ (the null space of A)

Lemma 3.2. Let the vectors $\{u_i\}_{i=1}^{n-m}$ be a basis in N(G). Let U be the matrix that has u_i as its i:th column, i=1,..., n-m, i.e.

$$U=\{u_1,\ldots, u_{n-m}\}$$

Then, if A1 and A2 hold, the matrix U^T B U is positive definite.

Proof. Let $\sigma = (\sigma_1,\ldots, \sigma_{n-m})^T$ be an arbitrary vector. Then $U\sigma \in N(G)$. We then get

$$\sigma^T U^T B U \sigma \geq \alpha ||U\sigma||^2 \geq \alpha' ||\sigma||^2 ,$$

where $\alpha'>0$. The first inequality follows from A1 and the last inequality from the fact that the columns of U are linearly independent. Because σ was arbitrary, it follows that $U^T B U$ is positive definite.

□

Lemma 3.3 If A1 and A2 hold, then for every z there exist vectors u and v such that

$$Bu + G^T v = z$$

$$Gu = 0 .$$

Also, the vectors u and v are unique.

Proof. Let u be given by $u=U\sigma$, where σ is the unique solution to $U^T BU\sigma = U^T z$. Then $v^T(z-Bu)=0$ if $v \in N(G)$, because N(G) is spanned by the columns of U and $U^T(z-Bu)=0$.

Because dim $N(G)=n-m$ and dim $V(G^T)=m$ (G has full rank by A2) and because $V(G^T)$ and N(G) are orthogonal we then have

$$z-Bu \in V(G^T)$$

Since the columns of G^T form a basis in $V(G^T)$ there exists a unique v such that $z-Bu=G^T v$.

□

Proof of theorem 3.1.

From Lemma 3.3 we have that the matrix $\begin{pmatrix} B & G^T \\ G & 0 \end{pmatrix}$ is nonsinqular and that

$$(z^T,\ 0)\begin{pmatrix} B & G^T \\ G & O \end{pmatrix}^{-1}\begin{pmatrix} z \\ 0 \end{pmatrix} = (z^T,\ 0)\begin{pmatrix} u \\ v \end{pmatrix} = z^T u$$

if

$$Bu+G^T v=z$$

$$Gu = 0.$$

Now $z \notin V(G^T) \Rightarrow u\neq 0$. But $u\in N(G)\Rightarrow z^T u=u^T(Bu+G^T v)=u^T Bu\geq \alpha\|u\|^2$.

□

Theorem 3.3. If A1 and A2 hold, then the matrix A in (3.8) is positive definite.

Proof. Let z be a linear combination of the columns in $\tilde{H}^T$, that is $z=\tilde{H}^T\mu$ for some $\mu\neq 0$. Then from Theorem 3.2 we have

$$0< (\mu^T\tilde{H},0)\begin{pmatrix} B & G^T \\ G & 0 \end{pmatrix}^{-1}\begin{pmatrix} \tilde{H}^T\mu \\ 0 \end{pmatrix} =$$

$$= \mu^T(\tilde{H},0)\begin{pmatrix} B & G^T \\ G & 0 \end{pmatrix}^{-1}\begin{pmatrix} \tilde{H}^T \\ 0 \end{pmatrix}\mu=\mu^T A\mu$$

Hence A is positive definite.

□

4. NONLINEAR OPTIMAL CONTROL PROBLEMS

4.1 Problem definition

In this chapter we will consider optimal control of discrete time dynamical systems, described in state space form. At time instant t, the system is described by a state vector $x(t)$ of dimension n_x and a control vector $u(t)$, of dimension n_u. Over the time horizon $t=0,1,\ldots,N$ the system is assumed to be described by the difference equation

$$x(t+1)=f(x(t),u(t),t),\quad t=0,\ldots,N-1\ . \tag{4.1}$$

Here f is a vector function of dimension n_x. The initial state of the system is given by

$$x(0)=x_0 \tag{4.2}$$

We want to choose the sequence of input vectors $u(t)$, $t=0,\ldots,N-1$ so that the system behaves in some desired manner. Suppose that we can measure how well the system satisfies our objectives by the performance index

$$J(u)=\ell(x(N),N)+\sum_{t=0}^{N-1}\ell(x(t),u(t),t) \tag{4.3}$$

where the functions ℓ are scalars. From equations (4.1) and (4.2) we see that the sequence of state vectors, $x(t)$, is uniquely determined, once we have chosen the control variables $u(t)$. Hence the performance index (4.3) is a function of the controls $u(t)$. This fact is stressed by the argument u in $J(u)$.

For safe operation of the system, it might be required that its states remain within certain limits, that may depend on time. Also, the input variable may be restricted in some way. To incorporate this situation in the formal description, we add the following constraints

$$h^i(x(t_i),u(t_i)) \leq 0, i=1,\ldots,p, \qquad (4.4)$$

to the system description. Here t_i are integers satisfying $0 \leq t_i \leq N$ and h^i are scalar functions. The number of constraints at a given time instant may vary from instant to instant, that is, the t_i:s in (4.4) need not be different. In (4.4) we may also easily incorporate terminal constraints, involving the state x(N) only.

The functions f, ℓ and h^i in (4.1), (4.2) and (4.4) are all supposed to be twice continuously differentiable with respect to x and u. Furthermore, the second order derivatives are supposed to be Lipschitz continuous in x and u. The optimal control problem is now to select the control variables u(t), t=0,1...,N-1 so that the performance index (4.3) is minimized, while the constraints (4.4) are satisfied. We recognize the housing maintenance problem, example 2.2, as a simple example of this general problem formulation.

4.2 Optimal control methods

Many computational methods have been designed to solve the optimal control problem defined in the previous section. Most of these methods use dynamic programming techniques. In dynamic programming, the optimal value function $V^o(x(t),t)$ is defined as

$$V^o(x(N),N)=\ell(x(N),N) \qquad (4.5a)$$

$$V^o(x(t),t)=\min_{u(t)} \{\ell(x(t),u(t),t)+V^o(f(x(t),u(t),t),t+1)\}. \qquad (4.5b)$$

The basic idea behind the methods proposed in Mayne (1966) and Dyer and McReynolds (1970) is to use (4.5b) to find the increment u(t) that minimizes the second order Taylor expansion of the expression within curly brackets in (4.5b). In order to accomplish that, the first and second order derivatives of V^o with respect to x are assumed to exist.

The perhaps best known method for solving optimal control problems is the differential dynamic programming (DDP-method) proposed by Jacobson and Mayne (1970). This method employs a global minimization with respect to u(t), after which the right hand side of (4.5b) is expanded to second order around this control. Another, very similar, algorithm is described in Mitter (1966), and McReynolds and Bryson (1965). This latter method requires the solution of an extra set of vector difference equations. The method is very closely related to the one that we will derive in section 4.3.1.

None of these methods is very good at handling constraints of the type (4.4). However, if these constraints are explicit functions of u(t), then they can be handled by a method proposed by Ohno (1978). This method is based on the fact that the Kuhn-Tucker conditions must be satisfied (pointwise in time) for the minimizing values in (4.5b), subject to the constraints (4.4). A comprehensive survey of the methods for optimal control is given by Polak (1973), and the reader is referred to this reference for further details.

4.3 Newton's method for optimal control problems

As we remarked earlier, the performance index (4.3) is a function of the control variables u(t), since the sequence of state vectors is uniquely determined as functions of u(t). The same is of course true also for the constraints (4.4). This means that the optimal control program (4.1)-(4.4) is a problem of the type (2.11), where the inputs u(t),t=0,...,N-1 are the unknown parameters. When we applied Wilson's (1963) method to the problem (2.11), we ended up with the quadratic subproblem (2.15). Here, we shall derive the corresponding quadratic subproblem, when Wilson's method is applied to the optimal control problem. The case with no constraints is handled in section 4.3.1. Section 4.3.2 deals with a case when there is only one constraint of the type (4.4), while the general case is deferred to section 4.3.3.

4.3.1 Derivation of the quadratic subproblem: the case of no constraints.

Let the vector z(t) contain all the control variables from time t up to time N-1:

$$z(t)=(u^T(N-1),u^T(N-2),\ldots,u^T(t))^T \tag{4.6}$$

Note that z(t) satisfies the recursion

$$z(t)=(z^T(t+1),u^T(t))^T \tag{4.7}$$

Let us also introduce the function V(x(t),z(t),t) which is the cost from time t up to time N when starting in state x(t) and using the control z(t). These functions can formally be written as

$$V(x(t),z(t),t)=\ell(x(t),u(t),t)+V(f(x(t),u(t),t),z(t+1),t+1) \tag{4.8}$$

Notice the difference between V^o in (4.5) and V in (4.8). V is a function of z(t) whereas V^o is the infimum of this function with respect to z(t) for the same x(t). Clearly, minimizing the performance index (4.3), subject to (4.1)-(4.2) is the same problem as that of minimizing $V(_0,z(0),0)$ with respect to z(0). Consequently, by the introduction of the functions V in (4.8) we have rewritten the optimal control problem (4.1)-(4.3) as an unconstrained minimization problem for the functions $V(x_0,z(0),0)$ in the variable z(0). Let us solve this problem using Wilson's (1963) method. Wilson's method reduces to Newton's method, when applied to an unconstrained problem (see Luenberger (1973), p. 155). The method is thus as follows: Let $z_k(0)$ be an approximation to the solution. Let $W(\Delta z(0),0)$ be the second order Taylor expansion of V(0) around this point, i.e.

$$V(x_0,z_k(0)+\Delta z(0),0,)=W(\Delta z(0),0)+O(|\Delta z(0)|^3)$$

or

$$W(\Delta z(0),0)=V(x_0,z_k(0),0)+V_z(0)\Delta z(0)+\tfrac{1}{2}\,\Delta z^T(0)V_{zz}(0)\Delta z(0) \quad (4.9)$$

A better approximation to the solution is now found by minimizing (4.9) with respect to $\Delta z(0)$ and adding the minimizing argument to $z_k(0)$, i.e.

$$z_{k+1}(0)=z_k(0)+\Delta z(0)$$

where

$$\Delta z(0)=-[V_{zz}(0)]^{-1}V_z^T(0) \quad (4.10)$$

To improve the convergence properties, eq. (4.10) is usually modified to

$$z_{k+1}(0)=z_k(0)+\alpha_k\Delta z(0) \quad (4.11)$$

where α_k is a scalar in the interval $(0,1]$. In chapter 8, we shall discuss how this scalar should be chosen. For convenience, we have suppressed the arguments x_0 and $z_k(0)$ in the above expressions. We shall do so also in the sequel, when there is no risk of confusion.

The expression (4.10) contains the first and second order derivatives of $V(0)$ with respect to $z(0)$. The following lemma guarantees that these derivatives exist.

Lemma 4.1. Suppose that that the functions f and ℓ in (4.1) and (4.3) are twice continuously differentiable with respect to x and u, and that the second derivatives are Lipschitz continuous in x and u. Then the functions $V(x(t),z(t),t)$ in (4.8) are twice continuously differentiable with respect to x and z, and the second order derivatives are Lipschitz continuous in x and z.

Proof. The lemma follows trivially by induction on (4.8) and from Theorem 9.12 in Rudin (1964).

□

From the lemma it also follows that the following derivatives exist:

$$V_x(N)=\ell_x(x(N)) \tag{4.12a}$$

$$V_{xx}(N)=\ell_{xx}(x(N)) \tag{4.12b}$$

which follows from (4.8a). From (4.8b) and (4.7) we further get

$$V_x(t)=\ell_x(t)+V_x(t+1)f_x(t) \tag{4.12c}$$

$$V_z(t)=(V_z(t+1),\ \ell_u(t)+V_x(t+1)f_u(t)) \tag{4.12d}$$

$$V_{xx}(t)=\ell_{xx}(t)+V_x(t+1)f_{xx}(t)+f_x^T(t)V_{xx}(t+1)f_x(t) \tag{4.12e}$$

$$V_{xz}(t)=(f_x^T(t)V_{xz}(t+1),\ell_{xu}(t)+V_x(t+1)f_{xu}(t)+f_x^T(t)V_{xx}(t+1)f_u(t)) \tag{4.12f}$$

$$V_{zz}(t)=\begin{pmatrix} V_{zz}(t+1) & V_{zx}(t+1)f_u(t) \\ f_u^T(t)V_{xz}(t+1) & \ell_{uu}(t)+V_x(t+1)f_{uu}(t)+f_u^T(t)V_{xx}(t+1)f_u(t) \end{pmatrix} \tag{4.12g}$$

These derivates are all evaluated at the point (x(t), z(t)), where x(t) satisfies (4.1) and (4.2) for all t, and u(t) is given by z(0).

Let W(t)=W(Δx(t),Δz(t),t) be the second order Taylor expansion of the function V(t) around the point (x(t),z(t)), i.e.

$$V(x(t)+\Delta x(t),z(t)+\Delta z(t),t)=W(t)+O\left(\left|\begin{matrix}\Delta x(t)\\ \Delta z(t)\end{matrix}\right|^3\right) \qquad (4.13)$$

or

$$W(t)=V(t)+V_x(t)\Delta x(t)+V_z(t)\Delta z(t)+\frac{1}{2}\,(\Delta x^T(t)V_{xx}(t)\Delta x(t)+$$

$$+2\Delta x^T(t)V_{xz}(t)\Delta z(t)+\Delta z^T(t)_{zz}+\Delta z^T(t)V_{zz}(t)\Delta z(t) \qquad (4.14)$$

If we now substitute $V(t)$, $V_x(t)$, $V_z(t)$, $V_{xx}(t)$, $V_{xz}(t)$, $V_{zz}(t)$ and $z(t)$ in (4.14) using the expressions (4.8b), (4.12c)-(4.12g) and (4.7) we get

$$W(t)=\ell(t)+V(t+1)+\ell_x(t)\Delta x(t)+V_x(t+1)f_x(t)\Delta x(t)+$$

$$+V_z(t+1)\Delta z(t+1)+\ell_u(t)\Delta u(t)+V_x(t+1)f_u(t)\Delta u(t)+$$

$$+\frac{1}{2}\,(\Delta x^T(t)(\ell_{xx}(t)+V_x(t+1)f_{xx}(t)+f_x^T(t)V_{xx}(t+1)f_x(t))\Delta x(t)+$$

$$+2\Delta x^T(t)f_x^T(t)V_{xz}(t+1)\Delta z(t+1)+$$

$$+2\Delta x^T(t)(\ell_{xu}(t)+V_x(t+1)f_{xu}(t)+f_x^T(t)V_{xx}(t+1)f_u(t))\Delta u(t)+$$

$$+\Delta z^T(t+1)V_{zz}(t+1)\Delta z(t+1)+2\Delta u^T(t)f_u^T(t)V_{xz}(t+1)\Delta z(t+1)+$$

$$+\Delta u^T(t)(\ell_{uu}(t)+V_x(t+1)f_{uu}(t)+f_u^T(t)V_{xx}(t+1)f_u(t))\Delta u(t) \qquad (4.15)$$

By rearranging the terms in (4.15) and introducing the auxiliary variable $D(t+1)$ as

$$D(t+1)=f_x(t)\Delta x(t)+f_u(t)\Delta u(t), \qquad (4.16)$$

we can write (4.15) as

$$W(t)=V(t+1)+V_x(t+1)D(t+1)+V_z(t+1)\Delta z(t+1)+$$

$$+\frac{1}{2}(D^T(t+1)V_{xx}(t+1)D(t+1)+2D^T(t+1)V_{xz}(t+1)\Delta z(t+1)+$$

$$+\Delta z^T(t+1)V_{zz}(t+1)\Delta z(t+1))+\ell(t)+\ell_x(t)\Delta x(t)+$$

$$+\ell_u(t)\Delta u(t)+\frac{1}{2}(\Delta x^T(t)(\ell_{xx}(t)+V_x(t+1)f_{xx}(t))\Delta x(t)+$$

$$+2\Delta x^T(t)(\ell_{xu}(t)+V_x(t+1)f_{xu}(t))\Delta u(t)+$$

$$+\Delta u^T(t)(\ell_{uu}(t)+V_x(t+1)f_{uu}(t))\Delta u(t)) \qquad (4.17)$$

Here we can identifiy the first part (cf (4.14)) as

$W(D(t+1),\Delta z(t+1),t+1)$.

Hence

$$W(\Delta x(t),\Delta z(t),t)=W(D(t+1),\Delta z(t+1),t+1)+\ell(t)+\ell_x(t)\Delta x(t)+$$

$$+\ell_u(t)\Delta u(t)+\frac{1}{2}(\Delta x^T(t)(\ell_{xx}(t)+V_x(t+1)f_{xx}(t))\Delta x(t)+$$

$$+2\Delta x^T(t)(\ell_{xu}(t)+V_x(t+1)f_{xu}(t))\Delta u(t)+$$

$$+\Delta u^T(t)(\ell_{uu}(t)+V_x(t+1)f_{uu}(t))\Delta u(t)) \qquad (4.18)$$

where $D(t+1)$ is given by (4.16). Notice that this auxiliary variable satisfies the dynamics of (4.1), when linearized around $(x(t),u(t))$. Therefore we shall henceforth use the natural notation $\Delta x(t+1)$ instead of $D(t+1)$. (See eq. (4.20b) below.) Let us also introduce the notation

$$Q_{xx}(N)=\ell_{xx}(N) \qquad (4.19a)$$

$$Q_{xx}(t)=\ell_{xx}(t)+V_x(t+1)f_{xx}(t) \qquad (4.19b)$$

$$Q_{xu}(t)=\ell_{xu}(t)+V_x(t+1)f_{xu}(t) \qquad (4.19c)$$

$$Q_{uu}(t)=\ell_{uu}(t)+V_x(t+1)f_{uu}(t), \tag{4.19d}$$

where $V_x(t)$ is defined by (4.12a) and (4.12c).

Eliminating W(t) for t=0,..,N-1 in (4.9) using (4.18) we obtain

$$W(\Delta z(0),0)=V(0)+\ell_x(N)\Delta x(N)+\frac{1}{2}\Delta x^T(N)Q_{xx}(N)\Delta x(N)+$$

$$+\sum_{t=0}^{N-1}\{\ell_x(t)\Delta x(t)+\ell_u(t)\Delta u(t)+\frac{1}{2}(\Delta x^T(t)Q_{xx}(t)\Delta x(t)+$$

$$+2\Delta x^T(t)Q_{xu}(t)\Delta u(t)+\Delta u^T(t)Q_{uu}(t)\Delta u(t))\}. \tag{4.20a}$$

The auxiliary variables $\Delta x(t), t=0,\ldots,N$ must satisfy

$$\Delta x(t+1)=f_x(t)\Delta x(t)+f_u(t)\Delta u(t) \tag{4.20b}$$

$$\Delta x(0)=0\ . \tag{4.20c}$$

With

$$\Delta z(0)=(\Delta u^T(N-1),\ldots,\Delta u^T(0))^T.$$

We notice that the expression (4.20) is the same as (4.9). Consequently, minimization of (4.20a) under the constraints (4.20bc) will give the same sequence $\Delta u(t), t=0,..,N-1$ as (4.10).

In the literature, the problem (4.20) is usually called the linear-quadratic control problem, since the dynamics is linear in $\Delta x(t)$ and $\Delta u(t)$, and the performance index is quadratic in these variables. The standard linear-quadratic control problem, however, contains no linear terms in the performance index (4.20a). For further details see Kwakernaak and Sivan (1972), p. 490, Dyer and McReynolds (1970), p. 42 or Bryson and Ho (1969) p. 46.

We shall discuss two different approaches to solving (4.20). In

chapter 5, we shall solve the equations that correspond to the Kuhn-Tucker necessary conditions for (4.20). In this section, we shall solve the problem by introducing a sequence of subproblems. These subproblems are:

$$\underset{\Delta z(t)}{\text{minimize}}\ J(\Delta x(t),\Delta z(t))=\ell_x(N)\Delta x(N)+\frac{1}{2}\Delta x^T(N)Q_{xx}(N)\Delta x(N)+$$

$$+\sum_{s=t}^{N-1}\{\ell_x(s)\Delta x(s)+\ell_u(s)\Delta u(s)+\frac{1}{2}(\Delta x^T(s)Q_{xx}(s)\Delta x(s)+$$

$$+2\Delta x^T(s)Q_{xu}(s)\Delta u(s)+\Delta u^T(s)Q_{uu}(s)\Delta u(s))\}. \qquad (4.21a)$$

Subject to

$$\Delta x(s+1)=f_x(s)\Delta x(s)+f_u(s)\Delta u(s),\ s=t,\ldots,N-1 \qquad (4.21b)$$

Let $J^*(\Delta x(t),t)$ be the value of the objective function in (4.21a) corresponding to the optimal control sequence $\Delta u(s),s=t,..,N-1$. We shall now proceeed to show that this function is quadratic in $\Delta x(t)$, i.e.

$$J^*(\Delta x(t),t)=a(t)+W_x(t)\Delta x(t)+\frac{1}{2}\Delta x^T(t)W_{xx}(t)\Delta x(t) \qquad (4.22)$$

for some $a(t),W_x(t)$ and $W_{xx}(t)$. Clearly, this holds for t=N, since

$$J^*(\Delta x(N),N)=\ell_x(N)\cdot\Delta x(N)+\frac{1}{2}\Delta x^T(N)Q_{xx}(N)\Delta x(N) \qquad (4.23)$$

Hence

$$a(N)=0 \qquad (4.24a)$$

$$W_x(N)=\ell_x(N) \qquad (4.24b)$$

$$W_{xx}(N)=Q_{xx}(N) \qquad (4.24c)$$

Suppose now that (4.22) holds for t=N,...,p+1. Then

$$J^*(\Delta x(p),p)=\min_{\Delta u(p)}\{\ell_x(p)\Delta x(p)+\ell_u(p)\Delta u(p)+\frac{1}{2}(\Delta x^T(p)Q_{xx}(p)\Delta x(p)+$$

$$+2\Delta x^T(p)Q_{xu}(p)\Delta u(p)+\Delta u^T(p)Q_{uu}(p)\Delta u(p))+$$

$$+J^*(f_x(p)\Delta x(p)+f_u(p)\Delta u(p),\ p+1)\} \qquad (4.25)$$

The $\Delta u(p)$ that minimizes the right hand side of (4.25) is given by

$$\Delta u(p)=-(Q_{uu}(p)+f_u^T(p)W_{xx}(p+1)f_u(p))^{-1}\{(\ell_u(p)+W_x(p+1)f_u(p))^T+$$

$$+(Q_{xu}(p)+f_x^T(p)W_{xx}(p+1)f_u(p))^T\Delta x(p)\} \qquad (4.26)$$

Substituting $\Delta u(t)$ in (4.25) by the expression (4.26) we find that $J^*(\Delta x(p),p)$ is also quadratic in $\Delta x(p)$ and that the coefficients are given by

$$a(p)=a(p+1)-\frac{1}{2}(\ell_u(p)+W_x(p+1)f_u(p))(Q_{uu}(p)+$$

$$+f_u^T(p)W_{xx}(p+1)f_u(p))^{-1}\cdot(\ell_u(p)+W_x(p+1)f_u(p))^T \qquad (4.27a)$$

$$W_x(p)=\ell_x(p)+W_x(p+1)f_x(p)-(\ell_u(p)+W_x(p+1)f_u(p))\cdot$$

$$\cdot(Q_{uu}(p)+f_u^T(p)W_{xx}(p+1)f_u(p))^{-1}(Q_{xu}(p)+f_x^T(p)W_x(p+1)f_u(p))^T$$

(4.27b)

$$W_{xx}(p)=Q_{xx}(p)+f_x^T(p)W_{xx}(p+1)f_x(p)-(Q_{xu}(p)+f_x^T(p)W_{xx}(p+1)f_u(p))\cdot$$

$$\cdot(Q_{uu}(p)+f_u^T(p)W_{xx}(p+1)f_u(p))^{-1}(Q_{xu}(p)+f_x^T(p)W_{xx}(p+1)f_u(p))^T$$

(4.27c)

By induction, we have consequently proved that (4.22) holds and

that the coefficients in these functions are given by (4.24) and (4.27). Notice that (4.27c) is the discrete-time matrix Riccati equation.

For convenience we introduce the following notation:

$$\delta u(t)=-(Q_{uu}(t)+f_u^T(t)W_{xx}(t+1)f_u(t))^{-1}(\ell_u(t)+W_x(t+1)f_u(t))^T \quad (4.28)$$

and

$$\beta_t=(Q_{uu}(t)+f_u^T(t)W_{xx}(t+1)f_u(t))^{-1}(Q_{xu}(t)+f_x^T(t)W_{xx}(t+1)f_u(t))^T \quad (4.29)$$

With this notation, equation (4.26) can be rewritten as

$$\Delta u(t)=\delta u(t)-\beta_t\Delta x(t). \quad (4.30)$$

The above shows that the problem (4.20) can be solved iteratively by solving the subproblems (4.21). The value of the performance index at the solution to these subproblems is given by (4.22) where the coefficients are given by (4.24) and (4.27). Finally, the solution to (4.20) is given by (4.30) where $\Delta x(t)$ is given by (4.20bc).

We are now ready to summarize the minimization of the performance index (4.3) using Newton's method, when no constraints (4.4) are present.

Algorithm 4.1.

I. Let an initial control sequence $z_0(0)$ be given (where $z(0)$ is defined in (4.6)), and put $k=0$.

II. Calculate and store the states $x(t)$, $t=0,\ldots,N$ for $z_k(0)$ according to (4.1) and (4.2). Calculate and store the value of the objective function (4.3).

III. For t=N,...,0 solve equations (4.12c) and (4.27) with the initial values (4.12a) and (4.24) respectively. During these calculations compute and store $\delta u(t)$ and β_t given by (4.28) and (4.29).

IV. If $\left(\sum_{t=0}^{N-1} \left| \delta u(t) \right|^2 \right)^{1/2} < \varepsilon$, where ε is a small positive number, go to step VII.

V. Let the elements $\Delta u(t)$ in $\Delta z_k(0)$ be given by (4.30), in which $\Delta x(t)$ satisfies (4.20b) and (4.20c).

VI. Put $z_{k+1}(0)=z_k(0)+\alpha_k \Delta z_k(0)$, where α_k is chosen in the interval $(0,1]$ such that a sufficient reduction is obtained in the performance index (4.3) when using the control $z_{k+1}(0)$ instead of $z_k(0)$. Let k=k+1 and go back to step II.

VII. Stop. $z_k(0)$ is close to a local minimum point.

Remark 4.1. In the above algorithm we assumed that the matrices $Q_{uu}(t)+f_u^T(t)W_{xx}(t+1)f_u(t)$, t=0,...,N-1 are positive definite for all k. In chapter 6 we shall discuss how to modify the algorithm if this is not the case.

□

Remark 4.2. It is not trivial to chose α_k in step VI. Different choices will be discussed in chapter 7.

□

We are now able to compare the method presented here with some of those mentioned in section 4.2. Let us start with some comments on the results of Mitter (1966) (continuous time) and McReynolds (1966) (as described in Bryson and Ho (1969)). Both

these authors have derived the same results as those given here, but they propose that the control should be given by

$$\Delta u(t)=\delta u(t)-\beta_t(x_{k+1}(t)-x_k(t)) \tag{4.31}$$

instead of (4.30). In (4.31) the state $x_{k+1}(t)$ is the result of the control $u_k(t)+\Delta u(t)$. (For simplicity we assume that $\alpha_k=1$.) This modification saves computer time, since it is no longer necessary to solve (4.20bc). However, with (4.31) we no longer take a Newton step towards the solution (cf equations (4.9) and (4.10)).

The methods proposed by Mayne (1966) and Dyer and McReynolds (1970) differ from the method discussed here in the following way. First they use (4.31) instead of (4.30). Second, they use the vectors $W_x(t)$ given by (4.24b) and (4.27b) instead of the vectors $V_x(t)$ given by (4.11ad), when calculating the matrices $Q_{xx}(t)$, $Q_{xu}(t)$ and $Q_{uu}(t)$ in (4.19). With this change it is not necessary to solve the difference equation (4.11d), and hence less computer work is required. But again, the method is no longer the true Newton method (as claimed in Dyer and McReynolds (1970), p. 69). Close to the solution $z^*(0)$ these differences are marginal since

$$x_{k+1}(t)-x_k(t) \rightarrow \Delta x(t) \tag{4.32}$$

and

$$W_x(t) \rightarrow V_x(t) \tag{4.33}$$

as $z_k(0) \rightarrow z^*(0)$.

Clearly when the dynamics (4.1) is linear in the variables x and u, all the described algorithms are identical.

4.3.2. Derivation of the quadratic subproblem: the case of one constraint.

Now assume that we have one constraint of the type (4.4) and that this constraint is active at time t=s:

$$h(x(s),u(s))\leq 0, \tag{4.34}$$

where h is a scalar function. As remarked earlier we can view x(s) as a function of x_0 and z(0), where z(0) is defined by (4.6). To find these functions and derivatives we proceed as with V(t) in (4.8). Consequently, introduce the functions H(x(t),z(t),t) through

$$H(x(t),z(t),t)=0, \qquad t>s \tag{4.35a}$$

$$H(x(s),z(s),s)=h(x(s),u(s)) \tag{4.35b}$$

$$H(x(t),z(t),t)=H(f(x(t),u(t),t),z(t+1),t+1),t<s. \tag{4.35c}$$

As in Lemma 4.1 it follows that these functions are twice differentiable and that they are given, for t=s, by

$$H_x(s)=h_x(s) \tag{4.36a}$$

$$H_z(s)=(0,h_u(s)) \tag{4.36b}$$

$$H_{xx}(s)=h_{xx}(s) \tag{4.36c}$$

$$H_{xu}(s)=(0,h_{xu}(s)) \tag{4.36d}$$

$$H_{zz}(s)=\begin{pmatrix} 0 & 0 \\ 0 & h_{uu}(s) \end{pmatrix} \tag{4.36e}$$

and for t<s by

$$H_x(t)=H_x(t+1)f_x(t) \tag{4.36f}$$

$$H_z(t)=(H_z(t+1),H_x(t+1)f_u(t)) \tag{4.36g}$$

$$H_{xx}(t)=H_x(t+1)f_{xx}(t)+f_x^T(t)H_{xx}(t+1)f_x(t) \tag{4.36h}$$

$$H_{xz}(t)=(f_x^T(t)H_{xz}(t+1),H_x(t+1)f_{xu}(t)+f_x^T(t)H_{xx}(t+1)f_u(t)) \tag{4.36i}$$

$$H_{zz}(t)=\begin{pmatrix} H_{zz}(t+1) & H_{zx}(t+1)f_u(t) \\ f_u^T(t)H_{xz}(t+1) & H_x(t+1)f_{uu}(t)+f_u^T(t)H_{xx}(t+1)f_u(t) \end{pmatrix} \tag{4.36j}$$

Our problem now is to find the value of $z(0)$ that minimizes $V(x_0,z(0),0)$ subject to the constraint

$$H(x_0,z(0),0)\leq 0$$

where $V(0)$ is given by (4.8) and $H(0)$ by (4.35). (Recall our convention of supressing the arguments x_0 and $z(0)$.) This is a problem of the type (2.11). For our problem, Wilson's method (corresponding to equations (2.15)) takes the form

$$\underset{\Delta z(0)}{\text{minimize}}\ V_z(0)\Delta z(0)+\frac{1}{2}\Delta z^T(0)\tilde{V}_{zz}(0)\Delta z(0) \tag{4.37a}$$

$$\text{subject to } H(0)+H_z(0)\Delta z(0)\leq 0 \tag{4.37b}$$

where the matrix $\tilde{V}_{zz}(0)$ is given by

$$\tilde{V}_{zz}(0)=V_{zz}(0)+\mu H_{zz}(0), \tag{4.38}$$

and μ is the Lagrange multiplier corresponding to the constraint (4.34). We shall now reformulate the problem (4.37) in the same way as we did in the previous subsection where (4.9) was re-written as (4.20). We start by introducing the functions $\tilde{W}(t)$ which are analagous to $W(t)$ in (4.14). They are defined by

$$\tilde{W}(t)=V(t)+V_x(t)\Delta x(t)+V_z(t)\Delta z(t)+\frac{1}{2}(\Delta x^T(t)\tilde{V}_{xx}(t)\Delta x(t)+$$

$$+2\Delta x^T(t)\tilde{V}_{xz}(t)\Delta z(t)+\Delta z^T(t)\tilde{V}_{zz}(t)\Delta z(t)) \qquad (4.39)$$

where

$$\tilde{V}_{xx}(t)=V_{xx}(t)+\mu H_{xx}(t) \qquad (4.40a)$$

$$\tilde{V}_{xz}(t)=V_{xz}(t)+\mu H_{xz}(t) \qquad (4.40b)$$

$$\tilde{V}_{zz}(t)=V_{zz}(t)+\mu H_{zz}(t) \qquad (4.40c)$$

and V(t) is given by (4.8). The matrices on the right hand side of (4.40) are either given by (4.12) or by (4.36). For $t>s$ the functions W(t) and $\tilde{W}(t)$ are identical. For $t<s$ we shall now rewrite $\tilde{W}(t)$ in the same way as W(t) was rewritten in (4.18). From (4.40) and (4.39) we have

$$\tilde{W}(t)=V(t)+V_x(t)\Delta x(t)+V_z(t)\Delta z(t)+\frac{1}{2}(\Delta x^T(t)(V_{xx}(t)+\mu H_{xx}(t))\Delta x(t)+$$

$$+2\Delta x^T(t)(V_{xz}(t)+\mu H_z(t))\Delta z(t)+\Delta z^T(t)(V_{zz}(t)+\mu H_{zz}(t))\Delta z(t))$$

Using (4.7), (4.8), (4.12) and (4.36) we get

$$\tilde{W}(t)=\ell(t)+V(t+1)+\ell_x(t)\Delta x(t)+V_x(t+1)f_x(t)\Delta x(t)+$$

$$+V_z(t+1)\Delta z(t+1)+\ell_u(t)\Delta u(t)+V_x(t+1)f_u(t)\Delta u(t)+$$

$$\frac{1}{2}\{\Delta x^T(t)(\ell_{xx}(t)+V_x(t+1)f_{xx}(t)+f_x^T(t)V_{xx}(t+1)f_x(t)+$$

$$+\mu H_x(t+1)f_{xx}(t)+\mu f_x^T(t)H_{xx}(t+1)f_x(t))\Delta x(t)+$$

$$+2\Delta x^T(t)f_x^T(t)(V_{xz}(t+1)+\mu H_{xz}(t+1))\Delta z(t+1)+2\Delta x^T(t)(\ell_{xu}(t)+$$

$$+V_x(t+1)f_{uu}(t)+f_x^T(t)V_{xx}(t+1)f_u(t)+$$

$$+\mu H_x(t+1)f_{xu}(t)+\mu f_x^T(t)H_{xx}(t+1)f_u(t))\Delta u(t)+$$

$$+\Delta z^T(t+1)(V_{zz}(t+1)+\mu H_{zz}(t+1))\Delta z(t+1)+2\Delta z^T(t+1)(V_{xz}(t+1)+$$

$$+\mu H_{zx}(t+1))f_u(t)\Delta u(t)+\Delta u^T(t)(\ell_{uu}(t)+$$

$$+V_x(t+1)f_{uu}(t)+f_u^T(t)V_{xx}(t+1)f_u(t)+$$

$$+\mu H_x(t+1)f_{uu}(t)+\mu f_u^T(t)H_{xx}(t+1)f_u(t))\Delta u(t)\}$$

Again we introduce the auxiliary variable D(t+1) as (4.16) and then let $\Delta x(t+1)=D(t+1)$. The above equation could then be written

$$\tilde{W}(t)=\ell(t)+\tilde{W}(t+1)+\ell_x(t)\Delta x(t)+\ell_u(t)\Delta u(t)+$$

$$+\frac{1}{2}\Delta x^T(t)(\ell_{xx}(t)+(V_x(t+1)+\mu H_x(t+1))f_{xx}(t))\Delta x(t)+$$

$$+2\Delta x^T(t)(\ell_{xu}(t)+(V_x(t+1)+\mu H_x(t+1))f_{xu}(t))\Delta u(t)+$$

$$+\Delta u^T(t)(\ell_{uu}(t)+(V_x(t+1)+\mu H_x(t+1))f_{uu}(t))\Delta u(t)) \qquad (4.41)$$

Performing the same calculations for t=s gives

$$\tilde{W}(s)=\ell(s)+W(s+1)+\ell_x(s)\Delta x(s)+\ell_u(s)\Delta u(s)+$$

$$+\frac{1}{2}(\Delta x^T(s)(\ell_{xx}(s)+V_x(s+1)f_{xx}(s)+\mu h_{xx}(s))\Delta x(s)+$$

$$+2\Delta x^T(s)(\ell_{xu}(s)+V_x(s+1)f_{xu}(s)+\mu h_{xu}(s))\Delta u(s)+$$

$$+\Delta u^T(s)(\ell_{uu}(s)+V_x(s+1)f_{uu}(s)+\mu h_{uu}(s))\Delta u(s)) \qquad (4.42)$$

From (4.12c) and (4.36f) we have

$$V_x(t)+\mu H_x(t)=\ell_x(t)+V_x(t+1)f_x(t)+\mu H_x(t+1)f_x(t)=$$

$$=\ell_x(t)+(V_x(t+1)+\mu H_x(t+1))f_x(t) \qquad (4.43)$$

for $t<s$ and for $t=s$ we have

$$V_x(s)+\mu H_x(s)=\ell_x(s)+V_x(s+1)f_x(s)+\mu h_x(s) \qquad (4.44)$$

Now introduce the row vectors $\tilde{V}_x(t)$ and the matrices $\tilde{Q}_{xx}(t)$, $\tilde{Q}_{xu}(t)$ and $\tilde{Q}_{uu}(t)$ as

$$\tilde{V}_x(N)=V_x(N)=\ell_x(N) \qquad (4.45a)$$

$$\tilde{V}_x(t)=\ell_x(t)+\tilde{V}_x(t+1)f_x(t), \qquad t\neq s \qquad (4.45b)$$

$$\tilde{V}_x(s)=\ell_x(s)+\tilde{V}_x(s+1)f_x(s)+\mu h_x(s) \qquad (4.45c)$$

and

$$\tilde{Q}_{xx}(N)=\ell_{xx}(N) \qquad (4.46a)$$

$$\tilde{Q}_{xx}(t)=\ell_{xx}(s)+\tilde{V}_x(t+1)f_{xx}(t), \qquad t\neq s \qquad (4.46b)$$

$$\tilde{Q}_{xx}(s)=\ell_{xx}(s)+\tilde{V}_x(s+1)f_{xx}(s)+\mu h_{xx}(s) \qquad (4.46c)$$

$$\tilde{Q}_{xu}(t)=\ell_{xu}(t)+\tilde{V}_x(t+1)f_{xu}(t), \qquad t\neq s \qquad (4.46d)$$

$$\tilde{Q}_{xu}(s)=\ell_{xu}(s)+\tilde{V}_x(s+1)f_{xu}(s)+\mu h_{xu}(s) \qquad (4.46e)$$

$$\tilde{Q}_{uu}(t)=\ell_{uu}(t)+\tilde{V}_x(t+1)f_{uu}(t), \qquad t\neq s \qquad (4.46f)$$

$$\tilde{Q}_{uu}(s)=\ell_{uu}(s)+\tilde{V}_x(s+1)f_{uu}(s)+\mu h_{uu}(s) \qquad (4.46g)$$

From equations (4.41)-(4.46) we see that $\tilde{W}(0)$ can be written

$$\tilde{W}(0)=V(0)+\ell_x(N)\Delta x(N)+\frac{1}{2}\Delta x^T(N)Q_{xx}(N)\Delta x(N)+$$

$$+\sum_{t=0}^{N-1}\left(\ell_x(t)\Delta x(t)+\ell_u(t)\Delta u(t)+\frac{1}{2}\left(\Delta x^T(t)\tilde{Q}_{xx}(t)\Delta x(t)+\right.\right.$$

$$\left.\left.+2\Delta x^T(t)\tilde{Q}_{xu}(t)\Delta u(t)+\Delta u^T(t)\tilde{Q}_{uu}(t)\Delta u(t)\right)\right) \qquad (4.47)$$

where the auxiliary variable $\Delta x(t)$ satisfies (4.20bc).

Hence solving the problem (4.37) will give the same $\Delta u(t)$ as the solution to:

$$\underset{\Delta u(t)}{\text{minimize}}\ \ell_x(N)\Delta x(N)+\frac{1}{2}\Delta x^T(N)\tilde{Q}_{xx}(N)\Delta x(N)+$$

$$+\sum_{t=0}^{N-1}\left\{\ell_x(t)\Delta x(t)+\ell_u(t)\Delta u(t)+\frac{1}{2}\left(\Delta x^T(t)\tilde{Q}_{xx}(t)\Delta x(t)+\right.\right.$$

$$\left.\left.+2\Delta x^T(t)\tilde{Q}_{xu}(t)\Delta u(t)+\Delta u^T(t)\tilde{Q}_{uu}(t)\Delta u(t)\right)\right\} \qquad (4.48a)$$

$$\text{subject to } \Delta x(t+1)=f_x(t)\Delta x(t)+f_u(t)\Delta u(t) \qquad (4.48b)$$

$$\Delta x(0)=0 \qquad (4.48c)$$

$$h(s)+h_x(s)\Delta x(s)+h_u(s)\Delta u(s)\leq 0 \qquad (4.48d)$$

4.3.3 Derivation of the quadratic subproblem: the case of several constraints.

We shall now generalize the results of the previous subsection to the case of several constraints of the type (4.4). To do this we need some instruments to indicate which constraints that are associated with a particular time instant. Therefore we introduce the sets I(t) as

$$I(t)=\{i:t_i=t\}. \qquad (4.49)$$

This definition of I(t) is illustrated by the following example.

Example 4.1. Let the constraints be given by

$$h^1(x(2),u(2))\leq 0$$

$$h^2(x(2),u(2))\leq 0$$

$$h^3(x(3))\leq 0$$

and let N=3. Then $I(0)=I(1)=\emptyset$, $I(2)=\{1,2\}$ and $I(3)=\{3\}$. □

We also introduce the functions

$$\gamma(t)=\gamma(x(t),u(t),\mu,t)=\sum_{i\in I(t)}\mu_i h^i(x(t_i),u(t_i)). \tag{4.50}$$

where μ now is a vector with the components μ_i.

In section 4.3.2 we found that the objective function in (4.20a) was changed to (4.48a) in case of one constraint. This constraint changed the calculation of $V_x(t)$ in (4.12c) to the calculation of $\tilde{V}_x(t)$ in (4.45). Also the calculation of $Q_{xx}(t)$, $Q_{xu}(t)$ and $Q_{uu}(t)$ changed from (4.19) to (4.46). If we examine these changes we see that they are linear. Hence adding more constraints will only result in linear changes in $\tilde{V}(t)$,$\tilde{Q}_{xx}(t)$, $\tilde{Q}_{xu}(t)$ and $\tilde{Q}_{uu}(t)$. Hence in case of several constraints we then let $\tilde{V}_x(t)$ be given by

$$\tilde{V}_x(N)=\ell_x(N)+\gamma_x(N) \tag{4.51a}$$

$$\tilde{V}_x(t)=\ell_x(t)+\tilde{V}_x(t+1)f_x(t)+\gamma_x(t) \tag{4.51b}$$

and the matrices $\tilde{Q}_{xx}(t)$, $\tilde{Q}_{xu}(t)$ and $\tilde{Q}_{uu}(t)$ by

$$\tilde{Q}_{xx}(N)=\ell_{xx}(N)+\gamma_{xx}(N) \tag{4.52a}$$

$$\tilde{Q}_{xx}(t)=\ell_{xx}(t)+\tilde{V}_x(t+1)f_{xx}(t)+\gamma_{xx}(t) \tag{4.52b}$$

$$\tilde{Q}_{xu}(t)=\ell_{xu}(t)+\tilde{V}_x(t+1)f_{xu}(t)+\gamma_{xu}(t) \tag{4.52c}$$

$$\tilde{Q}_{uu}(t)=\ell_{uu}(t)+\tilde{V}_x(t+1)f_{uu}(t)+\gamma_{uu}(t) \tag{4.52d}$$

If we use Wilson's method to find the $z(0)$ that minimizes the performance index (4.3) while satisfying the constraints (4.4) the increments in $z_k(0)$ are given by $\Delta z(0)$ in which $\Delta u(t)$ is given by the solution to

$$\underset{\Delta u(t)}{\text{minimize}}\ \ell_x(N)\Delta x(N)+\frac{1}{2}\Delta x^T(N)\tilde{Q}_{xx}(N)\Delta x(N)+$$

$$+\sum_{t=0}^{N-1}\{\ell_x(t)\Delta x(t)+\ell_u(t)\Delta u(t)+\frac{1}{2}(\Delta x^T(t)\tilde{Q}_{xx}(t)\Delta x(t)+$$

$$+2\Delta x^T(t)\tilde{Q}_{xu}(t)\Delta u(t)+\Delta u^T(t)\tilde{Q}_{uu}(t)\Delta u(t))\} \tag{4.53a}$$

$$\text{subject to}\quad \Delta x(t+1)=f_x(t)\Delta x(t)+f_u(t)\Delta u(t) \tag{4.53b}$$

$$\Delta x(0)=0 \tag{4.53c}$$

$$h^i(t_i)+h^i_x(t_i)\Delta x(t_i)+h^i_u(t_i)\Delta u(t_i)\leq 0 \tag{4.53d}$$

where $\tilde{Q}_{xx}(t)$, $\tilde{Q}_{xu}(t)$ and $\tilde{Q}_{uu}(t)$ are given by (4.51) and (4.52).

In chapter 5 we shall dicuss one method for finding the solution to (4.53) as well as the values of the corresponding multipliers μ_i. If this method to calculate the increments in $z(0)$, converges, then the convergence will be of second order. This follows since Newton's method is of second order.

4.4 Summary

The long and technical calculations in this chapter can be summarized in the following theorem:

Theorem 4.1. Suppose that the solution to (4.53) exists and is unique. Let this solution be denoted by $\Delta u(t)$. Then $\Delta u(t)$ is the update that is assigned to the control variable $u(t)$ when Wilson's method is applied to the optimal control problem (4.1)-(4.4).

□

The existence of a unique solution to (4.53) will be investigated in the next chapter.

5. ON THE CONSTRAINED LINEAR QUADRATIC PROBLEM

5.1 Introduction.

At the end of chapter 4 we discussed how to compute a Newton direction in the control space, for minimizing the performance index (4.3) subject to the constraints (4.1), (4.2) and (4.4). We showed that it could be found by solving (4.53), which is a constrained linear quadratic problem. In this chapter, we will investigate under which conditions a solution to (4.53) exists. We will also apply the method proposed in chapter 3 for finding this solution. Hence we will consider the following problem:

$$\text{minimize} \sum_{t=0}^{N-1} \{q_1^T(t)x(t)+q_2^T(t)u(t)+$$

$$+ \frac{1}{2} \{x^T(t)Q_1(t)x(t)+2x^T(t)Q_{12}(t)u(t)$$

$$+u^T(t)Q_2(t)u(t)\}\}+q_1^T(N)x(N)+ \frac{1}{2} x^T(N)Q_1(N)x(N) \qquad (5.1a)$$

$$\text{subject to} \quad x(t+1)=F(t)x(t)+G(t)u(t) \qquad (5.1b)$$

$$x(0)=x_0 \qquad (5.1c)$$

$$h^i+h_1^i x(t_i)+h_2^i u(t_i)\leq 0, \quad i=1,\ldots,p \qquad (5.1d)$$

Here $Q_1(t)$, $Q_{12}(t)$, $Q_2(t)$, $F(t)$ and $G(t)$ are given matrices of proper dimensions. The matrices $Q_1(t)$ and $Q_2(t)$ are symmetric. The column vectors $q_1(t)$, $q_2(t)$ and x_0, and the row vectors h_1^i and h_2^i are also assumed to be given and h^i are given scalars. The problem (5.1) is the same as (4.53) but we have rewritten it in order to simplify the notation. From now on we will refer to (5.1) as the CLQ-problem (the constrained linear quadratic problem). We will sometimes address problem (5.1) without the constraints (5.1d). This problem will be called the LQ-problem. The CLQ-problem is a problem of the type (3.1). To see this we let z be formed by the state vectors x(t) and the control

vectors u(t) such that

$$z=(x^T(N),u^T(N-1),x^T(N-1),..,\ u^T(0),\ x^T(0))^T. \tag{5.2}$$

Note that in chapter 4 we considered only the controls u(t) to be free variables. Doing this we could (at least implicitly) eliminate the state variables.

In this chapter however, we consider both the controls and the states as being free, because then it is easier to see the relationship between the problems (3.1) and (5.1). Using (5.2) and comparing (5.1a) with (3.1a) gives the relationships

$$B=\begin{pmatrix} Q_1(N) & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & Q_2(N-1) & Q_{12}^T(N-1) & & 0 & 0 & 0 & 0 \\ 0 & Q_{12}(N-1) & Q_1(N-1) & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & Q_2(1) & Q_{12}^T(1) & 0 & 0 \\ 0 & 0 & 0 & \dots & Q_{12}(1) & Q_1(1) & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & Q_2(0) & Q_{12}^T(0) \\ 0 & 0 & 0 & \dots & 0 & 0 & Q_{12}(0) & Q_1(0) \end{pmatrix} \tag{5.3}$$

and

$$b=(q_1^T(N),q_2^T(N-1),q_1^T(N-1),\dots,q_2^T(1),q_1^T(1),q_2^T(0),q_1^T(0))^T \tag{5.4}$$

Identifying (3.1b) with (5.1b) and (5.1c) gives

$$G=\begin{pmatrix} -I & G(N-1) & F(N-1) & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & -I & G(N-2) & F(N-2) & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & -I & G(0) & F(0) \\ 0 & 0 & 0 & 0 & 0 & & 0 & 0 & -I \end{pmatrix} \tag{5.5}$$

and

$$g=(0,0,\ldots,0,-x_0^T)^T \tag{5.6}$$

where in (5.5), I is the $n_x \times n_x$ unit matrix. The constraints (5.1d) determine the vector h and the matrix H in (3.1c) for the CLQ-problem, but we can only write H explicitly when the sequence $t_1, t_2,..,t_p$ in (5.1d) is known.

In chapter 3 we found that the problem (3.1) had a unique solution if the assumptions A1 and A2 were satisfied. Here we are going to investigate what these assumptions mean in the CLQ case. In section 5.2 we take a closer look at assumption A1 and in section 5.3 we look at assumption A2. In chapter 3 a method for solving the problem (3.1) was proposed. In section 5.4 we will present a factorization algorithm, based upon the Riccati equation, that is needed when using this method on the CLQ-problem. In section 5.5 we show how the rest of the method applies to the CLQ-problem.

5.2 Assumption A1 in the CLQ case.

Assumption A1 states that the matrix B is positive definite on the null space of G. For the CLQ case we then have the following result.

Theorem 5.1. If B is given by (5.3) and G by (5.5) then B is positive definite on the null space of G if the matrices

$$Q_2(t)+G^T(t)P(t+1)G(t),\ t=0,\ldots,\ N-1 \tag{5.7}$$

are positive definite, where P(t) is the solution of the Riccati equation

$$P(N)=Q_1(N) \tag{5.8a}$$

$$P(t)=Q_1(t)+F^T(t)P(t+1)F(t)-(Q_{12}(t)+F^T(t)P(t+1)G(t))\cdot$$

$$\cdot(Q_2(t)+G^T(t)P(t+1)G(t))^{-1}(Q_{12}^T(t)+G^T(t)P(t+1)F(t)),\ t=0,\ldots,N-1$$

(5.8b)

□

Proof: We start by constructing a basis for the null space of G in (5.5). Note that the basis consists of column vectors of the type (5.2) in which $x(0)=0$ and the other elements satisfy (5.1b). It is easily seen that we can satisfy $Gz=0$ for every choice of $u(t)$, $t=0,\ldots,N-1$ by just choosing the values of $x(t)$, $t=1,\ldots,N$ so that they satisfy (5.1b). Hence $Gz=0$ leaves $u(t)$, $t=0,\ldots,N-1$, as free variables and therefore the dimension of the null-space of G is at least $N\cdot n_u$.

Define the matrices $Z(t)$

$$Z(t)=(X_t^T(N),U_t^T(N-1),X_t^T(N-1),\ldots,U_t^T(0),X_t^T(0))^T \tag{5.9}$$

where the matrices $X_t(t)$ and $U_t(t)$ all have n_u columns and are given by

$$U_t(s)=0,\ s=0,\ldots,t-1 \tag{5.10a}$$

$$X_t(s)=0,\ s=0,\ldots,t \tag{5.10b}$$

$$U_t(t)=I \tag{5.10c}$$

$$X_t(t+1)=G(t) \tag{5.10d}$$

$$U_t(s)=-\beta_s X_t(s),\ s=t+1,\ldots,N-1 \tag{5.10e}$$

$$X_t(s+1)=\bar{F}(s)X_t(s),\ s=t+1,\ldots,N-1 \tag{5.10f}$$

where I is the $n_n \times n_u$ unit matrix, β_t is given by

$$\beta_t=(Q_2(t)+G^T(t)P(t+1)G(t))^{-1}(Q_{12}^T(t)+G^T(t)P(t+1)F(t)) \qquad (5.11)$$

and $\bar{F}(t)$ is defined as

$$\bar{F}(t)=F(t)-G(t)\beta_t \qquad (5.12)$$

In words, the i:th column of Z(t) (cf (5.2)) is generated by starting at state zero using zero control up to time t-1. At time t we let the i:th control signal be 1. After that we use the feedback law $u(s)=-\beta_s x(s)$. From this way of constructing Z(t) it is obvious that $GZ(t)=0$, $t=0,\ldots,N-1$ when G is given by (5.5). Let $\alpha(t)=(\alpha_1(t),\ldots,\alpha_{n_u}(t))^T$ be a vector. Then any vector of the type

$$z=\sum_{t=0}^{N-1} Z(t)\alpha(t) \qquad (5.13)$$

is a vector in the null space of G. On the other hand let z be a given vector in the null space of G. Then we can choose $\alpha(t)$, $t=0,\ldots,N-1$ so that (5.13) hols. Also all the columns in the matrices Z(t) are linearly independent so these columns form a basis in the null space of G.

An alternative way of writing (5.10) is

$$U_t(s)=0,\ s=0,\ldots,t-1 \qquad (5.14a)$$

$$X_t(s)=0,\ s=0,\ldots,t \qquad (5.14b)$$

$$U_t(t)=I, \qquad (5.14c)$$

$$X_t(t+1)=G(t), \qquad (5.14d)$$

$$U_t(s)=-\beta_s\bar{F}(s-1)\ldots\bar{F}(t+1)G(t),\ s>t \qquad (5.14e)$$

$$X_t(s+1)=\bar{F}(s)\bar{F}(s-1)\ldots\bar{F}(t+1)G(t),\ s>t \qquad (5.14f)$$

If z is given by (5.13) we have

$$z^T Bz = \sum_{t=0}^{N-1} \sum_{s=0}^{N-1} \alpha^T(s) Z^T(s) BZ(t) \alpha(t). \tag{5.15}$$

The term $Z^T(s)BZ(t)$ has the following property. From (5.3) and (5.14) we have

$$BZ(t) = (\bar{X}_t^T(N), \bar{U}_t^T(N-1), \ldots, \bar{U}_t^T(0), \bar{X}_t^T(0))^T \tag{5.16}$$

where $\bar{X}_t(t)$ and $\bar{U}_t(t)$ are given by

$$\bar{X}_t(N) = Q_1(N)\bar{F}(N-1)\ldots\bar{F}(N-1)\ldots\bar{F}(t+1)G(t) \tag{5.17a}$$

$$\bar{U}_t(p) = (Q_{12}^T(p) - Q_2(p)\beta_p)\bar{F}(p-1)\ldots\bar{F}(t+1)G(t) \quad p>t \tag{5.17b}$$

$$\bar{X}_t(p) = (Q_1(p) - Q_{12}(p)\beta_p)\bar{F}(p-1)\ldots\bar{F}(p-1)\ldots\bar{F}(t+1)G(t) \quad p>t \tag{5.17c}$$

$$\bar{U}_t(t) = Q_2(t) \tag{5.17d}$$

$$\bar{X}_t(t) = Q_{12}(t) \tag{5.17e}$$

$$\bar{U}_t(p) = 0 \qquad p<t \tag{5.17f}$$

$$\bar{X}_t(p) = 0 \qquad p \leq t \tag{5.17g}$$

For s>t we then have

$$Z^T(s)BZ(t) = (Q_{12}^T(s) + G^T(s)P(s+1)F(s) - (Q_2(s) + G^T(s)P(s+1)G(s))\beta_s) \cdot$$

$$\cdot \bar{F}(s-2)\ldots\bar{F}(t+1)G(t) = 0. \tag{5.18}$$

For s=t we have

$$Z^T(t)BZ(t) = Q_2(t) + G^T(t)P(t+1)G(t) \tag{5.19}$$

and for s<t we have

$$z^T(s)\beta Z(t)=G^T(s)\bar{F}^T(s+1)\ldots\bar{F}^T(t-2)(Q_{12}(t)+F^T(t)P(t+1)G(t)-$$

$$-\beta_t^T(Q_2(t)+G^T(t)P(t+1)G(t)))=0. \qquad (5.20)$$

To obtain (5.18)-(5-20) we have used (5.8) and (5.11).

Hence (5.15) reduces to

$$z^TBz=\sum_{t=0}^{N-1}\alpha^T(t)(Q_2(t)+G^T(t)P(t+1)G(t))\alpha(t) \qquad (5.21)$$

and we finally conclude that $z^TBz>0$ if $z\neq 0$ and $Gz=0$ and if the matrices $Q_2(t)+G^T(t)+G^T(t)P(t+1)G(t)$ are positive definite.

□

If the LQ problem (5.1a-c) was constructed to obtain a Newton direction to the constrained optimal control problem (4.1)-(4.3), we have the following result.

Corollary 5.1. The matrix $V_{zz}(0)$ given by (4.13) is positive definite if the matrices

$$Q_{uu}(t)+f_u^T(t)W_{xx}(t+1)f_u(t),\quad t=0,\ldots,N-1$$

are positive definite.

□

Proof: The statement follows from theorem 5.1 and from theorem 4.1, which says that $\Delta z(0)$ given by (4.10) is algebraically identical to $\Delta u(t)$, $t=0,\ldots,N-1$, which minimizes (4.20a) under the constraints (4.20c).

□

5.3 Assumption A2 in the CLQ case.

Assumption A2 is a constraint qualification for the quadratic programming problem. In the CLQ case it is also closely related to the concept of controllability as will be shown later in this section. For the CLQ problem, assumption A2 has the following equivalent formulation.

Assumption A2: $\mu_i=0$, $i=1,\ldots,p$ and $\lambda(t)=0$, $t=0,\ldots,N$, is the only solution of

$$\lambda(N)=\sum_{i\in I(N)}\mu_i(h_1^i(N))^T \tag{5.22a}$$

$$0=G^T(t)\lambda(t+1)+\sum_{i\in I(t)}\mu_i(h_2^i(t))^T;\ t=0,\ldots,N-1 \tag{5.22b}$$

$$\lambda(t)=F^T(t)\lambda(t+1)+\sum_{i\in I(t)}\mu_i(h_1^i(t))^T,\ t=0,\ldots,N-1 \tag{5.22c}$$

where $I(t)$ are defined by

$$I(t)=\{i:\ t_i=t\} \tag{5.23}$$

(cf (4.49)).

For the general case it is very difficult to decide whether or not (5.22a-c) have non-zero solutions in λ and μ. For some special cases however it is possible to reformulate equations (5.22) into a more practical condition. When there are no constraints at all we have the following result.

Proposition 5.1. If there are no constraints of the type (5.1d) defined by problem (5.1a-d) then the equations (5.22a-c) have the unique solution $\lambda(t)=0$, $t=0,..,N$.

□

Proof. If we have no constraints, then the sums in (5.22) are all zero. Hence (5.22a) gives $\lambda(N)=0$ and (5.22c) shows that

$\lambda(t+1)=0$ implies $\lambda(t)=0$. The proof then follows by induction.

□

Note that another way to formulate proposition 5.1 is to say that the matrix G given by (5.5) has full rank.

If assumption A2 holds, then there exists at least one point that satisfies the constraints. For the CLQ-problem it is obvious that we have a feasible point of the constraints (5.1b)-(5.1d) if the constraints (5.1d) are explicit functions of u(t) and that we for all t could satisfy (5.1d) $i\varepsilon I(t)$ by just choosing a proper u(t). This fact suggests the following theorem.

Theorem 5.1. If $I(N)=\emptyset$ and if for each t the vectors h_2^i, $i\varepsilon I(t)$ are linearly independent, then A2 holds.

□

Proof. From (5.22a) and $I(N)=\emptyset$ we have $\lambda(N)=0$. Assume that we have $\lambda(t)=0$ and $\mu_i=0$, $i\varepsilon I(t)$ for t=N,..., p+1. By using (5.22b) for t=p we then get

$$0=\sum_{i\varepsilon I(p)} \mu_i (h_2^i)^T. \tag{5.24}$$

But because the vectors $h_2^i, i\varepsilon I(p)$ are linearly independent, (5.24) implies $\mu_i=0$, $i\varepsilon I(p)$. The equation (5.22c) then reduces to

$$\lambda(p)=F_x^T(p)\lambda(p+1) \tag{5.25}$$

But $\lambda(p+1)=0$ and hence $\lambda(p)=0$.

The result then follows from induction.

□

The most interesting case, however, is when we have some pure state constraints, i.e. $h_2^i=0$. In order to handle this type of problem we consider the concept of controllability. We define the transition matrix $\phi(t,s)$, $t\geq s$, and the controllability matrix $W(t,s)$, $t>s$, as

$$\phi(t+1,s)=F(t)\phi(t,s),\ t\geq s,\ \phi(s,s)=I \tag{5.26}$$

and

$$W(t,s)=\sum_{p=s}^{t-1}\phi(t,p+1)G(p)G^T(p)\phi^T(t,p+1) \tag{5.27}$$

If we have only one constraint which is a state constraint relating to the time t_1, then we have the following result.

Theorem 5.2. Suppose that we have only one constraint and that $h_2^1=0$ for this constraint. Suppose also that $W(t_1,t_0)$, defined by (5.27) is nonsingular for some t_0 such that $0\leq t_0<t_1\leq N$. Then A2 holds if $h_1^1\neq 0$.

□

Proof: From (5.22) and (5.26) we have

$$\lambda(t)=0,\ t=t_1+1,\ldots,N \tag{5.28a}$$

$$\lambda(t)=\mu_1\phi^T(t_1,t)(h_1^1)^T,\ t=t_0,\ldots,t_1 \tag{5.28b}$$

and from (5.22b) we then get

$$\mu_1G^T(t)\phi^T(t_1,t+1)(h_1^i)^T=0;\ t=t_0,\ldots,t_1-1. \tag{5.29}$$

This could also be written

$$\mu_1C(h_1^1)^T=0 \tag{5.30}$$

where C is the matrix

$$C=(\phi(t_1,t_1)G(t_1-1),\phi(t_1,t_1-1)G(t_1-2),\ldots,\phi(t_1,t_0+1)G(t_0))^T \tag{5.31}$$

But $C^TC=W(t_1,t_0)$. Hence, multiplying (5.30) by C^T we get

$$\mu_1 C^TC(h_1^1)^T=\mu_1 W(t_1,t_0)(h_1^1)^T=0. \tag{5.32}$$

Because $W(t_1,t_0)$ is nonsingular and $h_1^1\neq 0$, (5.32) implies $\mu_1=0$ and from (5.28b) it follows that $\lambda(t)=0$, $t=0,\ldots,N$.

□

If we have several state constraints but they all relate to different times, the following corollary follows from theorem 5.2.

Corollary 5.1 Let the times t_i, $i=0,\ldots,p$ satisfy $0\leq t_0<t_1<\ldots<t_p\leq N$. If the matrices $W(t_i,t_{i-1})$ are nonsingular and $h_1^i\neq 0$ for $i=1,..,p$ then A2 holds.

□

Now consider the time invariant case with one time invariant state constraint related to each time.

Theorem 5.3. Let the dynamics in (5.1b) be time invariant such that $F(t)=\bar{F}$ and $G(t)=\bar{G}$, $t=0,\ldots,N-1$ and let the constraints in (5.1d) be time invariant state constraints such that

$$h_1x(t_i)+h^i\leq 0 \tag{5.33}$$

($h_1\neq 0$) where $t_i\geq n_x$, $\forall i$ and $t_i\neq t_j$ if $i\neq j$. Suppose that the

matrix C given by

$$C=(\bar{G},\bar{F}\bar{G},\ldots,\bar{F}^{n_x-1}\bar{G})^T \tag{5.34}$$

has full rank. Then A2 holds.

□

Proof: Without loss of generality we assume that $t_i=n_x+i-1$.

Let k be the smallest nonnegative integer such that

$$h_1(\bar{F})^k\bar{G}\neq 0. \tag{5.35}$$

That this k exists follows from (5.34) and the fact that C has full rank and $h_1\neq 0$.

From (5.22) we get

$$\lambda(t)=\sum_{s=t}^{N}\mu_{s-n_x+1}(\bar{F}_x^T)^{s-t}(h_1)^T,\quad n_x\leq t\leq N \tag{5.36a}$$

$$\lambda(t)=\sum_{s=n_x}^{N}\mu_{s-n_x+1}(\bar{F}^T)^{s-t}(h_1)^T,\quad 0\leq t\leq n_x-1 \tag{5.36b}$$

Now using (5.22b) for t=N-k gives

$$0=\bar{G}^T\lambda(N-k)=\mu_{N-n_x+1}\bar{G}^T(\bar{F}^T)(h_1)^T \tag{5.37}$$

The second equality follows from (5.36) and the fact that $h_1(\bar{F})^p\bar{G}=0$ if $p<k$. Hence from (5.37) we have $\mu_{N-n_x+1}=0$. Assume that $\mu_{N-n_x+1}=\ldots=\mu_{p-n_x+2}=0$. Then using this fact and (5.22b) for t=p-k gives

$$0=\bar{G}^T\lambda(p-k)=\mu_{p-n_x+1}\bar{G}^T(\bar{F}^T)^k(h_1)^T \tag{5.38}$$

which says that $\mu_{p-n_x+1}=0$. Hence by induction we have $\mu_i=0$,

$i=1,\ldots,N-n_x+1$, and from (5.36) it follows that

$\lambda(t)=0,\ t=0,..,N.$

□

5.4 Factorization and the Riccati equation

As we have already mentioned, the problem (5.1) is a problem of the type defined by (3.1). Therefore we can use the method proposed in chapter 3 to find the solution to (5.1). When using this method we need an efficient method for factorizing the matrix of (3.6) when B and G are given by (5.3) and (5.5). We will show here that such a factorization method is obtained when the Riccati equation is solved. Therefore we first solve the problem neglecting the constraints (3.1c), which in our case corresponds to the constraints (5.1d). We have already presented a method for finding the solution to the LQ-problem in section 4.3.1. In this chapter we will again present the same method but derive it in a different way. We do this for two reasons. The first reason is that we want to stress that solving the Riccati equation is a factorization method and the second reason is that it is easier to see what matrices define this factorization. Hence consider the LQ-problem (5.1a)-(5.1c). The Lagrangian for this problem is

$$L(x,u,\lambda)=q_1^T(N)x(N)+\frac{1}{2}x^T(N)Q_1(N)x(N)+\lambda^T(0)(x_0-x(0))+$$

$$+\sum_{t=0}^{N-1}\{q_1^T(t)x(t)+q_2^T(t)u(t)+\frac{1}{2}(x^T(t)Q_1(t)x(t)$$

$$+2x^T(t)Q_{12}(t)u(t)+u^T(t)Q_2(t)u(t))+$$

$$+\lambda^T(t+1)(F(t)x(t)+G(t)u(t)-x(t+1)\} \qquad (5.39)$$

If $\bar{x}(t)$, $t=0,\ldots,N$ and $\bar{u}(t)$, $t=0,\ldots,N-1$ is the solution to the LQ-problem, then the Kuhn-Tucker necessary conditions state that there exist multipliers $\bar{\lambda}(t)$, $t=0,\ldots,N$ such that $\frac{\partial L}{\partial x(t)}=0$,

$\frac{\partial L}{\partial \lambda(t)}=0$, t=0,...,N and $\frac{\partial L}{\partial u(t)}=0$, t=0,...,N where the derivatives of L are evaluated for the arguments $\bar{x}(t)$, $\bar{\lambda}(t)$ and $\bar{u}(t)$. We then get the following equation for $\bar{x}(t)$, $\bar{u}(t)$ and $\bar{\lambda}(t)$

$$\frac{\partial L}{\partial x(N)} = -\bar{\lambda}(N)+Q_1\bar{x}(N)+q_1(N)=0 \tag{5.40a}$$

$$\frac{\partial L}{\partial \lambda(t+1)} = -\bar{x}(t+1)+G(t)\bar{u}(t)+F(t)\bar{x}(t)=0 \tag{5.40b}$$

$$\frac{\partial L}{\partial u(t)} = G^T(t)\bar{\lambda}(t+1)+Q_2(t)\bar{u}(t)+Q_{12}^T(t)\bar{x}(t)+q_2(t)=0 \tag{5.40c}$$

$$\frac{\partial L}{\partial x(t)} = F^T(t)\bar{\lambda}(t+1)+Q_{12}(t)\bar{u}(t)-\lambda(t)+Q_1(t)\bar{x}(t)+q_1(t)=0 \tag{5.40d}$$

$$\frac{\partial L}{\partial \lambda(0)}=-\bar{x}(0)+x_0=0 \tag{5.40e}$$

where the equalities (5.40bcd) hold for t=0,...,N-1.

By merging the vectors $\lambda(t)$, $x(t)$ and $u(t)$ into $\bar{\zeta}$ such that

$$\bar{\zeta}=(\bar{\lambda}^T(N),\bar{x}^T(N),\bar{u}^T(N-1),\bar{\lambda}^T(N-1),\bar{x}^T(N-1),\ldots,\bar{u}^T(0),\bar{\lambda}^T(0)\bar{x}^T(0))^T, \tag{5.41}$$

we could write the conditions (5.40) as

$$\mathcal{A}\bar{\zeta}=b \tag{5.42a}$$

where

$$b=(-q_1^T(N),0,-q_2^T(N-1),\ldots,0,-q_2^T(0),-q_1^T(0),-x_0^T)^T \tag{5.43b}$$

and

$$\mathcal{A} = \begin{pmatrix} \mathcal{A}_N & 0 & 0 & 0 & 0 \\ & \mathcal{A}_{N-1} & 0 & 0 & 0 \\ 0 & & \mathcal{A}_{N-2} & 0 & 0 \\ 0 & 0 & 0 & \mathcal{A}_0 & \\ 0 & 0 & 0 & 0 & -I \end{pmatrix} \tag{5.43}$$

where I is the $n_x \times n_x$ unit matrix, $\mathcal{A}_t$, t=0,..., N-1 are $(2n_x+n_u)\times(4n_x+n_u)$ matrices given by

$$\mathcal{A}_t = \begin{pmatrix} 0 & -I & G(t) & 0 & F(t) \\ G^T(t) & 0 & Q_2(t) & 0 & Q_{12}^T(t) \\ F^T(t) & 0 & Q_{12}(t) & -I & Q_1(t) \end{pmatrix} \tag{5.44}$$

and $\mathcal{A}_N$ is a $n_x \times (2n_x)$ matrix given by

$$\mathcal{A}_N = (-I, Q_1(N)). \tag{5.45}$$

The matrix $\mathcal{A}$ in (5.43) is hence a block diagonal matrix with rectangular blocks where the width of the diagonal is 5 blocks. The matrix $\mathcal{A}$ is the same matrix as

$$\begin{pmatrix} B & G \\ G & 0 \end{pmatrix}$$

where B and G are given by (5.3) and (5.5) but where we have reordered the columns and the rows in a certain way. This reordering has been done so that if assumtion A1 holds (cf section 5.2) then it is possible to obtain a LU-factorization of the matrix without performing any pivoting. We have the following result.

Theorem 5.4. If assumption A1 holds for the LQ-problem and $\mathcal{A}$ is given by (5.43), then there exists a lower block triangular matrix $\mathcal{L}$ and an upper triangular matrix $\mathcal{U}$ with ones on the diagonal such that

$$\mathcal{A} = \mathcal{L}\mathcal{U} \tag{5.46}$$

The matrix $\mathcal{L}$ is given by

$$\mathcal{L} = \begin{pmatrix} -I & 0 & 0 & 0 & 0 & 0 \\ \mathcal{L}_{N-1} & & 0 & 0 & 0 & 0 \\ 0 & \mathcal{L}_{N-2} & & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathcal{L}_0 & & 0 \\ 0 & 0 & 0 & 0 & 0 & -I \end{pmatrix} \tag{5.47}$$

where $\mathcal{L}_t$ are $(2n_x+n_u)\times(3n_x+n_u)$ matrices given by

$$\mathcal{L}_t = \begin{pmatrix} 0 & -I & 0 & 0 \\ G^T(t) & G^T(t)P(t+1) & Q_2(t)+G^T(t)P(t+1)G(t) & 0 \\ F^T(t) & F^T(t)P(t+1) & Q_{12}(t)+F^T(t)P(t+1)G(t) & -I \end{pmatrix} \tag{5.48}$$

The matrix $\mathcal{U}$ is given by

$$\mathcal{U} = \begin{pmatrix} I & -P(N) & 0 & 0 & 0 & 0 \\ 0 & \mathcal{U}_{N-1} & & 0 & 0 & 0 \\ 0 & 0 & \mathcal{U}_{N-2} & & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathcal{U}_0 & \\ 0 & 0 & 0 & 0 & 0 & I \end{pmatrix} \tag{5.49}$$

where $\mathcal{U}_t$ is $(2n_x+n_u)\times(3n_x+n_u)$ matrices given by

$$\mathcal{U}_t = \begin{pmatrix} I & -G(t) & 0 & -F(t) \\ 0 & I & 0 & \beta_t \\ 0 & 0 & I & -P(t) \end{pmatrix} \tag{5.50}$$

The matrices P(t) and β_t in (5.48)-(5.50) are given by (5.8) and (5.11).

□

Proof. From multiplication of $\mathcal{L}$ and $\mathcal{U}$ and from (5.8) and (5.11) the theorem follows trivially.

□

The solution of (5.42a) is obtained by first solving

$$\mathcal{L}\zeta' = b \tag{5.51}$$

and then solving

$$\mathcal{U}\bar{\zeta}=\zeta' \tag{5.52}$$

Solving (5.51) gives

$$\zeta'=(W^T(N),0,u_0^T(N-1),W^T(N-1),0,\ldots,u_0^T(0),W^T(0),x_0^T)^T \tag{5.53}$$

where $W(t)$ and $\delta u(t)$ are given by

$$W(N)=q_1(N) \tag{5.54a}$$

$$W(t)=q_1(t)-\beta_t^T q_2(t)+(F(t)-G(t)\beta_t)^T W(t+1),t=0,\ldots,N-1 \tag{5.54b}$$

and

$$u_0(t)=-(Q_2(t)+G^T(t)P(t+1)G(t))^{-1}(q_2(t)+G^T(t)W(t+1)),t=0,\ldots,N-1 \tag{5.54c}$$

Compare (4.24b), (4.27b), (4.28)and (4.29). In these equations the row vector $W_x(t)$ corresponds to $W(t)$ in (5.54).

From (5.52) we finally get the solution (5.41) to (5.42a) as

$$\bar{x}(0)=x_0 \tag{5.55a}$$

$$\bar{u}(t)=u_0(t)-\beta_t\bar{x}(t) \tag{5.55b}$$

$$t=0,\ldots,N-1$$

$$\bar{x}(t+1)=F(t)\bar{x}(t)+G(t)\bar{u}(t) \tag{5.55c}$$

$$\bar{\lambda}(t)=W(t)+P(t)\bar{x}(t),t=0,\ldots,N \tag{5.55d}$$

Usually, however, we have no need for the multipliers $\bar{\lambda}(t)$. Therefore we do not necessarily have to compute $\bar{\lambda}(t)$ and therefore we do not have to store $W(t)$. But the vectors $u_0(t)$ must be stored.

When we examine the equations (5.54) and (5.55) we find that the factorization of $\mathcal{A}$ in (5.42) is defined by the matrices $G(t), F(t), Q_2(t)+G^T(t)P(t+1)G(t)$ and β_t if we do not need $\bar{\lambda}(t)$.

Hence we now have an effective method to factorize the left hand side matrix in (3.6) in the LQ-case. Also, we know how to find the solution $\bar{z}$ to (3.6) in this case.

5.5 Inequality constraints.

We are now ready to handle the inequality constraints (5.1d). In the method proposed in chapter 3 we need to calculate the matrix Z in (3.5) which is the solution to (3.7). We also need to calculate $\tilde{\mu}'$ which is the solution to (3.8). Hence we also need to calculate the matrix A in (3.9) and the vector d in (3.10).

Without any loss of generality we assume that the first p' of the constraints (5.1d) are in the active set. Now we use Remark 3.1 to calculate the i:th column in Z (3.5).

Hence in the vector b given by (5.42b) we replace $q_2(t_i)$ and $q_1(t_i)$ by h_2^i and h_1^i. All other elements in b are set to zero and the resulting vector is called b_i. Let the i:th columns of Z and A be stored in ζ_i with the same structure as indicated by (5.41). Then we get this ζ_i as the solution to

$$\mathcal{L}\zeta_i' = b_i \tag{5.56a}$$

$$\mathcal{U}\zeta_i = \zeta_i' \tag{5.56b}$$

where ζ_i' has the same structure as (5.53). Comparing with (5.54) we see that the solution to (5.56a) is

$$W_i(t)=0, \quad t=t_i+1,\ldots,N \tag{5.57a}$$

$$u_i(t)=0,\ t=t_i+1,\ldots,N \tag{5.57b}$$

$$W_i(t_i)=(h_1^i-\beta_t^T h_2^i)^T \tag{5.57c}$$

$$u_i(t_i)=-(Q_2(t)+G^T(t)P(t+1)G(t))^{-1}(h_2^i)^T \tag{5.57d}$$

$$W_i(t)=(F(t)-G(t)\beta_t)^T W_i(t+1),t=0,\ldots,t_i-1, \tag{5.57e}$$

$$u_i(t)=-(Q_2(t)+G^T(t)P(t+1)G(t))^{-1}G^T(t)W_i(t+1),t=0,\ldots,t_i-1. \tag{5.57f}$$

and if we compare with (5.55) we see that the solution to (5.56b) is

$$x_i(0)=0 \tag{5.58a}$$

$$\bar{u}_i(t)=u_i(t)-\beta_t x_i(t), \tag{5.58b}$$

$$t=0,\ldots,N-1$$

$$x_i(t+1)=F(t)x_i(t)+G(t)\bar{u}_i(t), \tag{5.58c}$$

where in (5.57) and (5.58) the subscript i indicates the i:th column of Z.

The element a_{ij} in the matrix A in (3.9) is given by

$$a_{ij}=-H_1^i x_j(t_i)-h_2^i\bar{u}_j(t_i). \tag{5.59}$$

The element d_i in the vector d in (3.10) is given by

$$d_i=h^i+h_1^i\bar{x}(t_i)+h_2^i\bar{u}(t_i). \tag{5.60}$$

where $\bar{x}(t)$ and $\bar{u}(t)$ is given by (5.55).

We then know all quantities that are needed to solve (3.8) to obtain $\tilde{\mu}'$. The complete solution to the CLQ-problem is

$$x(t)=\bar{x}(t)+\sum_{i=1}^{p'}\tilde{\mu}_i'x_i(t),t=0,\ldots,N \tag{5.61a}$$

$$u(t)=\bar{u}(t)+\sum_{i=1}^{p'}\tilde{\mu}_i'\bar{u}_i(t),t=0,\ldots,N-1. \tag{5.61b}$$

This is of course true only if we have chosen the correct set of active constraints. Otherwise we have to find another set of active constraints according to the rules given in section 3. By substituting $\bar{u}(t)$ and $u_i(t)$ in (5.61b) and using (5.55b) and (5.58b) we get

$$u(t)=u_0(t)-\beta_t\bar{x}(t)+\sum_{i=1}^{p'}\tilde{\mu}_i'(u_i(t)-\beta_t x_i(t))=u_0(t)+\sum_{i=1}^{p'}\tilde{\mu}_i'u_i(t)-$$

$$-\beta_t(\bar{x}(t)+\sum_{i=1}^{p'}\tilde{\mu}_i'x_i(t))=u_0(t)+\sum_{i=1}^{p'}\tilde{\mu}_i'u_i(t)-\beta_t x(t) \tag{5.62}$$

where the last equality follows from (5.61a). Therefore we can choose to store $u_i(t)$ instead of $\bar{u}_i(t)$ and $x_i(t)$. Instead of (5.61) we then use

$$x(0)=x_0 \tag{5.63a}$$

$$u(t)=u_0(t)+\sum_{i=1}^{p'}\tilde{\mu}'_i u_i(t)-\beta_t x(t) \tag{5.63b}$$

$$t=0,\ldots,N-1$$

$$x(t+1)=F(t)x(t)+G(t)u(t) \tag{5.63c}$$

However, in order to be able to calculate the elements a_{ij} given by (5.59) we have to calculate $x_i(t)$ and $u_i(t)$ but we need not necessarily store them. In the first iteration of the method proposed in chapter 3 we know which constraints belong to the active set. We can therefore use (5.59) to calculate all elements in A. However, in later iterations when a new constraint is added to the active set, we cannot use (5.59) for all new elements if we do not want to recalculate $x_i(t)$ and $\bar{u}_i(t)$ for already active constraints. Only the elements in the new column in A could be obtained from (5.59). But then we utilize the fact

that the matrix A is symmetric, and therefore also the new row could be calculated without storing the values $x_i(t)$ and $\bar{u}_i(t)$ for already active constraints.

6 REGULARIZATION ISSUES

6.1 Introduction

In chapter 4 we discussed how to determine a Newton direction in the space of control actions for the optimal control problem (4.1)-(4.4). This was achieved by solving the quadratic sub-problem (4.53). For this sub-problem to have a unique solution we introduced the sufficient assumptions A1 and A2 dealing with positive definiteness and linear independence respectively. In the previous chapter we discussed how to test whether these assumptions are satisfied.

However, even if these assumptions are satisfied at the optimal solution to the posed problem, it may happen that they are violated for some sub-problem that we encounter during the iterative search for the solution. In this chapter we shall briefly discuss how the sub-problems can be modified so that the sufficient conditions A1 and A2 are assured to be satisfied for all subproblems that arise during the iterative procedure. In section 6.2, assumption A1 is treated, while assumption A2 is discussed in section 6.3. When nothing else is explicitly stated, we shall use the same notation as in chapters 2 and 4.

6.2 Regularization of matrices (Assumption A1)

The assumption A1 requires that the Hessian of the Lagrangian is positive definite on the orthogonal complement to the space spanned by the gradients of the equality constraints. When applied to an unconstrained optimization problem this simply means that the Hessian of the objective function should be positive definite. If this is not the case then Newton's method is unsuitable for the numerical search, since it does not generate descent directions. A very natural and common way to handle that situation is to add a matrix to the Hessian so that the resulting matrix is positive definite (see e.g. Luenberger (1973), p. 157). This idea was proposed by Levenberg (1944) and Marquardt (1963). Here we shall describe two basically different choices of such matrices, and how to adapt the Levenberg-Marquardt

method to our particular problem. The two choices in question are:

C1: μI where $\mu \geq 0$ and I is the unit matrix
C2: A non-negative diagonal matrix D.

These methods are also described in Fletcher (1980). The choice C1 may be motivated in the following way:
Assume that we want to solve the problem

$$\text{minimize } \ell(z) \qquad z \in R^n \tag{6.1}$$

(cf (2.11)).

Newton's method for this problem then generates search directions which are the solutions to

$$\underset{d_k}{\text{minimize}}\ \ell_z(z_k)d_k + \frac{1}{2}\, d_k^T \ell_{zz}(z_k)d_k. \tag{6.2}$$

If $\ell_{zz}(z_k)$ is positive definite and the Taylor series is adequate in a sufficently large region around z_k, then this should give a reasonable decrease in the objective function. However, if $\ell_{zz}(z_k)$ is indefinite then the expression (6.2) has no lower bound and the problem has no bounded solution. A possible way to modify this problem is to restrict the search for a new value to a small enough region around the previous iterate:

$$\underset{d_k}{\text{minimize}}\ \ell_z(z_k)d_k + \frac{1}{2}\, d_k^T \ell_{zz}(z_k)d_k \tag{6.3a}$$

$$\text{subject to } |d_k|^2 \leq \varepsilon_k \tag{6.3b}$$

here ε_k is a properly chosen positive scalar. The Lagrangian for the problem (6.3) is

$$L(d_k,\mu) = \ell_z(z_k)d_k + \frac{1}{2}\, d_k^T \ell_{zz}(z_k)d_k + \frac{\mu}{2}\,(d_k^T d_k - \varepsilon_k^2) \tag{6.4}$$

and the Kuhn-Tucker conditions for this problem are

$$\ell_z^T(z_k)+(\ell_{zz}(z_k)+\mu I)d_k=0 \tag{6.5a}$$

$$d_k^T d_k-\varepsilon_k^2\leq 0 \tag{6.5b}$$

$$\mu\geq 0 \tag{6.5c}$$

$$\mu(d_k^T d_k-\varepsilon_k^2)=0 \tag{6.5d}$$

If the constraint (6.3b) is active, the solution to the problem (6.3) is

$$d_k= -(\ell_{zz}(z_k)+\mu I)^{-1}\ell_z^T(z_k) \tag{6.6}$$

where μ is chosen so that $|d_k|=\varepsilon_k$. However, any choice of μ such that the matrix

$$\ell_{zz}(z_k)+\mu I \tag{6.7}$$

is positive definite will make d_k given by (6.6) a descent direction. Therefore, in practice the variable ε_k is never explicitly given. Rather, μ is chosen so that the matrix (6.7) is positive definite. When μ tends to infinity, this direction tends to the steepest descent direction. This means that too big a value of μ may a give slow convergence rate.

The choice C2 is motivated by the following idea from Gill, Murray and Picken (1972). The idea is to make a U^TU-factorization of the modified matrix:

$$U^TU=\ell_{zz}(z_w)+D_k. \tag{6.8}$$

The elements of the diagonal matrix D_k are chosen in the course of this factorization so that the right hand side of (6.8) is

made positive definite. This way of choosing D_k gives $D_k=0$ if $\ell_{zz}(z_k)$ is positive definite. An advantage of this method over (6.7) is that the factorization (6.8) has to be completed only once, while several tests for positive definiteness in (6.7) may be necessary. On the other hand, the choice (6.8) may give a direction d_k which is is almost orthogonal to the steepest descent direction. Examples of some search directions are given in figure 6.1

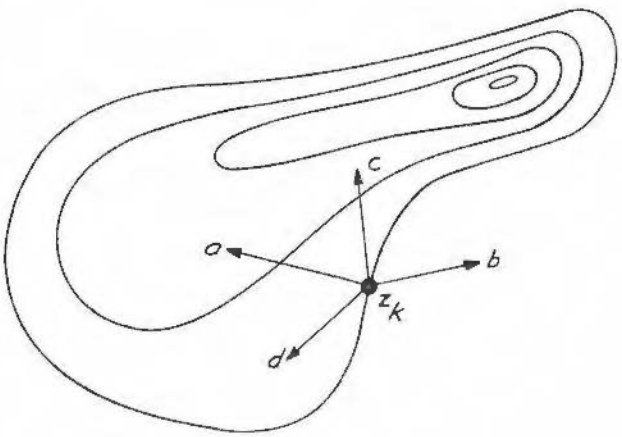


Figure 6.1 Examples of search direction d_k when d_k is given by $d_k=-M^{-1}\ell_z^T(z_k)$ and M is given by a) M=I (steepest descent), b) $M=\ell_{zz}(z_k)$ (Newton, indefinite case), c) $M=\ell_{zz}(z_k)+\mu I$, d) $\mu=\ell_{zz}(z_k)+D_k$. (worst case).

□

As the figure is drawn, the choice C1 appears to be superior to the others. Some numerical evidence that this may indeed be the case are given in Fletcher (1980), p. 88.

Let us now consider the optimal control problem (4.1)-(4.4). We found in section 4.3.1 that the Newton step for the unconstrained optimal control problem is given by (4.10):

$$\Delta z(0) = -\left[V_{zz}(0)\right]^{-1} V_z^T(0),$$

where V(t) is given by (4.8) and $\Delta z(0)$ is the increment in u(t), t=0,...,N-1. (see equation (4.6)). Now, if $V_{zz}(0)$ is not positive definite, then we modify it using either of the choices C1 or C2. Comparing with equation (4.14g) we find that the choice C1 corresponds to adding μI to the matrices $\ell_{uu}(t)$, t=0,...,N-1 and the choice C2 to adding D(t) to $\ell_{uu}(t)$ where D(t) are non-negative diagonal matrices. From corollary 5.1 we know that we can check if the resulting matrix is positive definite just by investigating the matrices

$$\ell_{uu}(t)+D(t)+V_x(t+1)f_{uu}(t)+f_u^T(t)W_{xx}(t+1)f_u(t),\ t=0,\ldots,N-1 \tag{6.9}$$

where $W_{xx}(t+1)$ is the solution to the Riccati equation

$$V_x(N)=\ell_x(N) \tag{6.10a}$$

$$W_{xx}(N)=\ell_{xx}(N) \tag{6.10b}$$

$$V_x(t)=\ell_x(t)+V_x(t+1)f_x(t) \tag{6.10c}$$

$$\begin{aligned} W_{xx}(t)=&\ell_{xx}(t)+V_x(t+1)f_{xx}(t)+f_x^T(t)W_{xx}(t+1)f_x(t)-(\ell_{xu}(t)+\\ &+V_x(t+1)f_{xu}(t)+f_x^T(t)W_{xx}(t+1)f_u(t))\cdot(\ell_{uu}(t)+D(t)+V_x(t+1)f_{uu}(t)+\\ &+f_u^T(t)W_{xx}(t+1)f_u(t))^{-1}\cdot(\ell_{ux}+V_x(t+1)f_{ux}(t)+f_u^T(t)W_{xx}(t+1)f_x(t)) \end{aligned} \tag{6.10d}$$

We have written these equations here for the choice C2. In the case of C1 we just replace D(t) with μI. Notice that the choice C1 requires μ to be equal for all the N time instants. Therefore, using the choice C1 we may have to solve the Riccati equation (6.10) several times before a proper value of μ is found. In the case of C2, however, the matrices D(t) can be found on-line, by solving (6.10) only once per iteration. Therefore the choice C2 will be used in our algorithm for the optimal control problem, despite the possible disadvantage indicated in figure 6.1.

It is interesting to note that adding the matrix D(t) to $\ell_{uu}(t)$ may be interpreted as adding the term

$$\frac{1}{2}\,(u(t)-u_k(t))^T D(t)(u(t)-u_k(t)) \tag{6.11}$$

to the cost-functions $\ell(x(t),u(t),t)$ in the performance index (4.3), where $u^k(t)$ is the currently used control signal. Adding such a term is very similar to the convergence control parameter (CCP) technique proposed by Järmark (1977), for continuous time optimal control problems. In the original CCP-method, however, the same matrix is used at every time instant, and the choice of the matrix is determined by the decrease in the cost function between iterations. From this discussion it may consequently be concluded that the CCP-technique may be interpreted as a Levenberg-Marquardt method.

Hence when we use the choice C2, we replace the matrices $\tilde{Q}_{uu}(t)$ by

$$\tilde{\tilde{Q}}_{uu}(t)=\tilde{Q}_{uu}(t)+D_t \tag{6.12}$$

when solving (4.53). In the following when we consider this formulation instead of (4.53) we will call it "the regularized CLQ-problem".

6.3 On assumption A2.

The assumption A2, that the gradients of all the constraints be linearly independent, is actually stronger than necessary. The algorithm given in chapter 3 will work well as long as the rows of G and $\tilde{H}$ are linearly independent throughout the computations, where $\tilde{H}$ contains the rows corresponding to the active constraints (3.1c). If, during the computations, the active constraints become linearly dependent, there are four, conceptually different, possible causes for this:

R1. There exists no point that satisfies all the constraints (2.11bc).

R2. The constraints are linearly dependent at the optimum.

R3. There exists no point that satisfies the constraints in the quadratic sub-problem (2.15) even though (2.11bc) have feasible points.

R4. The constraints (2.15bc) have feasible points, but when solving the quadratic subproblem (2.15), the algorithm uses so many active constraints that they become linearly dependent.

The first two reasons depend on the actual problem only, and there is nothing that the algorithm can do about it. The following example illustrates how R2 may happen:

Example 6.1. Consider the problem

$$\text{minimize} \quad z_1$$

$$\text{subject to} \quad z_2 - z_1^3 \leq 0$$

$$-z_2 \leq 0$$

The solution to this problem is $z=\begin{pmatrix}0\\0\end{pmatrix}$. At this point however, the constraints have the derivatives (0,1) and (0,-1), respectively, which are linearly dependent.

□

The possibilities R3 and R4 of course also depend on the problem but they also depend on the starting point given to the algorithm, and on how the algorithm chooses the active constraints among the inequality constraints. We illustrate R3 with the following example.

Example 6.2. Assume that we have two variables only, z_1 and z_2, and that we have two nonlinear equality constraints $g_1(z)=0$ and $g_2(z)=0$. Assume also that we have simple bounds on the variables such that

$$-1\leq z_i \leq 1,\ i=1,2 \tag{6.12}$$

and that the constraints have a feasible point strictly inside this box. If the point z_k is too far away from this feasible point it may happen that the linearized constraint $g(z)=0$ has no solution inside the box (6.12). See Figure 6.2.

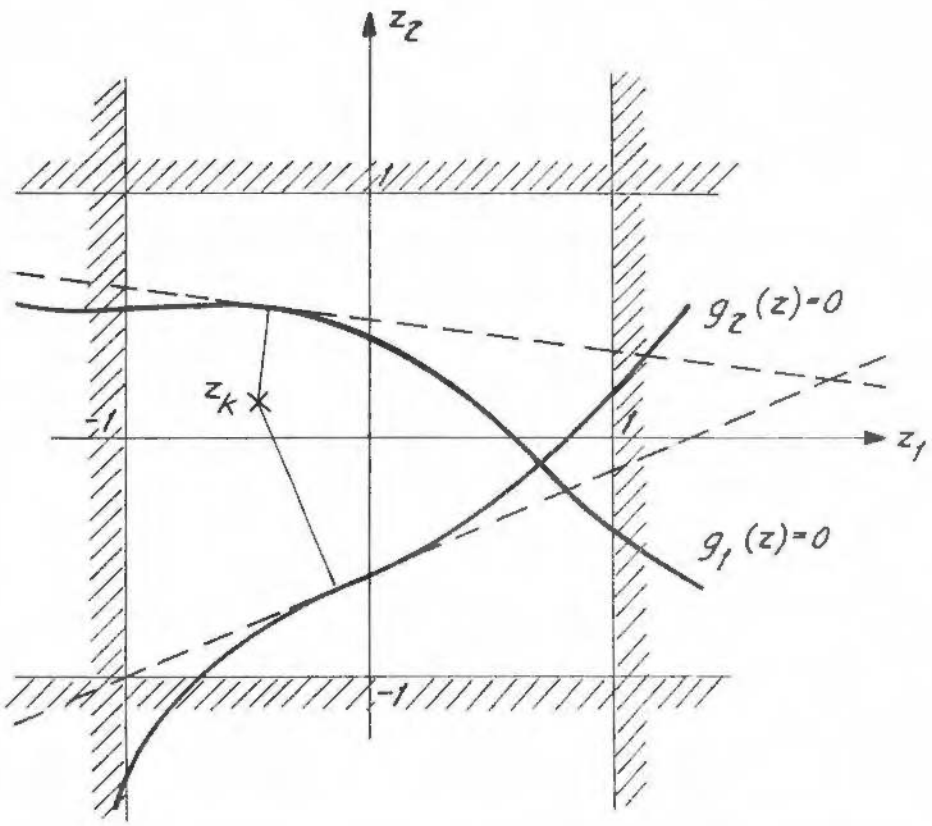


Figure 6.2. Two dimensional example of reason R3. The solid curves are the constraints $g_1(z)=0$ and $g_2(z)=0$. The dashed lines are the constraints $g(z_k)+g_z(z_k)(z-z_k)=0$ which have no solution inside the box (6.12).

□

A remedy for R3 has been proposed by Powell (1977). His idea is to scale all equality constraints and all violated inequality constraints so that they will have a feasible solution close to z_k. Then the nonviolated inequality constraints should not, hopefully, be activated during the iteration in question. Therefore a little step could be taken towards a feasible point of the nonlinear constraints. The choice of the scaling factor, mentioned above, has been further studied by Tone (1983).

The possibility R4 may occur when there are more active constraints than free variables. Therefore, in order to avoid that R4 occurs, the technique for choosing active constraints when solving the quadratic problems (2.15) should be designed so that the number of active constraints is kept small throughout the calculations. A strategy for selecting the active set, that usually gives fewer active constraints than the strategy described in section 3.2, is given in Goldfarb and Idnani (1983). They use the strategy of section 3.2 applied to the dual problem to (2.15).

7. CONVERGENCE OF THE OPTIMIZATION ALGORITHM

7.1 Introduction

So far we have only considered the problem of finding a proper direction when iteratively solving the nonlinear constrained programming problem (2.11). We have said nothing about how to choose the value of α_k in (2.12). This choice is very important for the convergence of the algorithms proposed earlier.

The discussion of convergence is usually divided into two concepts, namely global convergence and local convergence. When we investigate global convergence for an algorithm, we find the conditions under which the algorithm converges to a Kuhn-Tucker-point from an arbitrary starting point. This Kuhn-Tucker-point might not even be a local minimum point, but to prove convergence to a local minimum point from an arbitrary starting point seems to be a very complicated task.

When we talk about local convergence, we estimate the speed of convergence close to a local minimum point. The speed could for instance be linear, superlinear or quadratic.

When we derived our algorithm for constrained optimal control problems, we did that by interpreting it as a nonlinear programming problem. Therefore, all convergence results obtained for Wilson's method or Quasi-Newton versions of Wilson's method can be applied to our algorithm. According to Ohno (1978) no proof of convergence has been presented for the Mayne (1970) algorithm or analogus second order algorithms e.g. (Jacobson (1968), Dyer and McReynolds (1970)). However, Ohno (1978) proved that his algorithm for solving constrained optimal control problems converges to a Kuhn-Tucker point.

In section 7.2 we give some rules for choosing the step-length for unconstrained problems and in section 7.3 we discuss constrained problems. We then use the Watchdog technique, given in Chamberlain et al (1979), for the constrained optimal control problem and give proofs for both global and local convergence

in this case. In Section 7.4 we prove global convergence and in Section 7.5 local convergence.

7.2 Step length rules for unconstrained optimization.

The basic idea of step length rules is the following. We assume that we have the problem (2.11) but without constraints. Then the problem reduces to

$$\text{minimize } \ell(z) \qquad z\in R^n \tag{7.1}$$

We also assume that with some method (for instance the Newton method or the steepest descent method) we have found a descent direction d_k, i.e. $\ell_z(z_k)\cdot d_k<0$. Now we want to determine α_k in the formula

$$z_{k+1}=z_k+\alpha_k d_k \tag{7.2}$$

so that $z_k\to z^*$ (i.e. the solution of (7.1)) when $k\to\infty$. The most obvious way to choose α_k is to minimize $\ell(z_{k+1})$. Hence

$$\alpha_k=\arg\min_{\alpha} \ell(z_k+\alpha d_k) \tag{7.3}$$

This method is, however, usually not feasible to use in practice, since it requires a lot of function evaluations. (Luenberger (1973) p. 147).

Therefore we must accept rules for choosing α_k which do not make $\ell(z_{k+1})$ as small as possible. However, to accept α_k as soon as the criterion

$$\ell(z_{k+1}) <\ell(z_k) \tag{7.4}$$

is satisified, is not sufficient, because then the segence $\{z_k\}$ may converge to a point $\bar{z}$ for which $\ell_z(\bar{z})\neq 0$. Goldstein and Armijo have proposed rules for choosing α_k in (7.2) such that $\{z_k\}$ converges to a point $\bar{z}$ for which $\ell_z(\bar{z})=0$. See pp. 20-21 in Bertsekas (1982) for details of these methods and other rules for choosing the step-size.

7.3 Step length rules for constrained optimization.

When we have constraints in (2.11) we must not only pay attention to the value of $\ell(z_k+\alpha_k d_k)$ when we choose the parameter α_k, but also to the violations of the constraints. We must, therefore, choose α_k to obtain a suitable balance between the decrease (or increase) in the objective function $\ell(z)$ and how well the constraints

$$g(z)=0 \tag{7.5a}$$

$$h(z)\leq 0 \tag{7.5b}$$

are satisfied at the point z_{k+1}.

This balance is usually achieved by constructing a criterion function which is the sum of the objective function $\ell(z)$ and a penalty term to the constraints (2.11bc). For examples of such criterion functions, see Powell (1977) and Han (1977). The method for choosing the step-length α, for which the strongest convergence results holds, is the Watchdog technique given by Chamberlain et al. (1979). In the next section we will apply that method to the optimal control problem.

7.4 Convergence for the constrained optimal control problem.

In section 4.1 we defined the constrained optimal control problem by (4.1)-(4.4).

$$x(t+1)=f(x(t),u(t),t),t=0,\ldots,N-1 \tag{7.6a}$$

$$x(0)=x_0 \tag{7.6b}$$

$$J(u)=\ell(x(N),N)+\sum_{t=0}^{N-1}\ell(x(t),u(t),t) \tag{7.6c}$$

$$h^i(x(t_i),u(t_i))\leq 0,i=1,\ldots,p \tag{7.6d}$$

The problem is to minimize J(u) in (7.6c) when the constraints (7.6ab) and (7.6d) are satisfied. The Kuhn-Tucker necessary conditions for a solution to this problem is that there must exist multipliers λ and μ such that the following conditions are satisfied:

$$\ell_x(N)+\gamma_x(N)-\lambda^T(N)=0 \tag{7.7a}$$

$$\ell_x(t)+\gamma_x(t)+\lambda^T(t+1)f_x(t)-\lambda^T(t)=0 \tag{7.7b}$$

$$\ell_u(t)+\gamma_u(t)+\lambda^T(t+1)f_u(t)=0 \tag{7.7c}$$

$$\mu \geq 0 \tag{7.7d}$$

$$\mu_i \cdot h^i(x(t_i),u(t_i))=0 \tag{7.7e}$$

(for defeninition of $\gamma(t)$ see eq. (4.50)) as well as the conditions (7.6a), (7.6b) and (7.6d). When we talk about a Kuhn-Tucker-point for the problem (7.6) in the following we mean $u(t)$, $\lambda(t)$ and μ such that these conditions are satisfied.

We assume that we use Newton's method in the control space for solving problem (7.6). We then get a sequence $\{z_k(0)\}$ (cf (4.6)) which hopefully converges to a local minimum point (4.1)-(4.4).

The sequence $\{z_k(0)\}$ is defined by

$$z_{k+1}(0)=z_k(0)+\alpha_k \Delta z(0) \tag{7.8}$$

where $\Delta z(0)$ is given by the solution to the constrained linear quadratic optimal control problem (4.53). When solving this problem we assume that the regularization technique proposed in section (6.2) is used.

Here, we will use the Watchdog technique as described in

Chamberlain et al (1979) for choosing the value of α_k in (7.7). We start by defining two criterion functions $J_p(u)$ and $J_L(u)$ as

$$J_p(u)=J(u)+\sum_{i=1}^{p} \bar{\mu}_i \max(0,h^i(x(t_i),u(t_i)) \tag{7.9a}$$

$$J_L(u,\mu)=J(u)+\sum_{i=1}^{p} \mu_i h^i(x(t_i),u(t_i)) \tag{7.9b}$$

where $J(u)$ is given by (7.6c) and μ_i in (7.9b) is the Lagrange multiplier of the constraint i. The function $J_L(u,\mu)$ is actually the Lagrangian to the problem (7.6). This is the reason for the subscript L in $J_L(u)$. The subscript P in $J_p(u)$ stands for penalty. We assume that the constants $\bar{\mu}_i, i=1,\ldots,p$ have been chosen so that

$$\bar{\mu}_i > (\mu_k)_i \quad i=1,\ldots,p, \quad k=0,1,\ldots \tag{7.10}$$

at each iteration. The multiplier μ_k is obtained when solving (4.53) at iteration k and $(\mu_k)_i$ is the i:th component of this multiplier. We now define the following two criteria.

$$\text{C1:} \quad J_p(u_k+\alpha\Delta z(0)) \leq J_p(u_k)+\alpha\cdot\sigma\Delta J_{kz} \tag{7.11}$$

where $\sigma\in(0,\frac{1}{2})$ and ΔJ_{kz} is given by

$$\Delta J_{kz}=\ell_x(N)\Delta x(N)+\sum_{t=0}^{N-1}(\ell_x(t)\Delta x(t)+\ell_u(t)\Delta u(t))+$$

$$+\sum_{i=1}^{p}(h_x^i(t_i)\Delta x(t_i)+h_u^i(t_i)\Delta u(t_i))\cdot\bar{\mu}_i\cdot I(h^i(t_i)>0) \tag{7.12}$$

Here $\Delta u(t)$ is generated by the solution to (4.53) and $\Delta x(t)$ is the solution of the difference equation

$$\Delta x(t+1)=f_x(t)\Delta x(t)+f_u(t)\Delta u(t), t=0,..,N-1 \tag{7.13a}$$

$$\Delta x(0)=0 \tag{7.13b}$$

I(T) is an indicator function which is one if the condition T is true and zero otherwise. Note that ΔJ_{kz} is the directional derivative of $J_p(u_k)$ in the direction $\Delta z(0)$.

The other criterion is:

$$C2: J_L(u_k+\alpha\Delta z(0),\mu_k)<J_L(u_k,\mu_k) \tag{7.14}$$

Let $\beta\in(0,1)$ and let s be a positive integer. The watchdog technique is now given by the following scheme.

Step 0: Let a control sequence u_0 be given. Set k=0 and $w=J_p(u_0)$. Calculate $\Delta z(0)$ given by the solution to (4.53) for this control sequence.

Step 1: Let $\alpha_k=\beta^j$ where j is the smallest nonnegative integer such that at least one of the criteria C1 and C2 is satisfied. Put $u_{k+1}=u_k+\alpha_k\Delta z(0)$ and go to step 3.

Step 2: Let $\alpha_k=\beta^j$ where j is the smallest nonnegative integer such that criterion C1 is satisfied. Put $u_{k+1}=u_k+\alpha_k\Delta z(0)$.

Step 3: Calculate $\Delta z(0)$ for the control sequence u_{k+1}. If $J_p(u_k)\leq w$, set $r=k, w=J_p(u_k)$ and $v=J_p(u_k)+\sigma\alpha_k\Delta J_{kz}$.

Step 4: Put k=k+1. If $J_p(u_k)\leq v$ go to step 1, and if $v<J_p(u_k)\leq w$ go to step 2.

Step 5: If $k-s\leq r$ go to step 2. Otherwise set $u_k=u_r$ and r=k and then go to step 2.

If we have no constraints of the type (7.6d), this method reduces to the Armijo step size rule.

The reason for using two criterion functions is that using only criterion C1 may prevent superlinear convergence of the sequence $\{z_k\}$ and using only the criterion C2, the sequence $\{z_k\}$ may converge to a point that is not a Kuhn-Tucker point.

The global convergence of this algorithm is assured by the following theorem.

Theorem 7.1. (From Chamberlain et al. (1979)). Let $\{u_k\}$ be given by the watchdog technique and assume that the predicted decrease ΔJ_{kz} is bounded away from zero if u_k is bounded away from a Kuhn-Tucker point. Then, if $J_p(u_k)$ is bounded below, the sequence $\{u_k: k=0,1,.\}$ has a limit point which is a Kuhn-Tucker point.

□

Proof: See Chamberlain et al. (1979).

□

That ΔJ_{kz} is bounded away from zero if u_k is bounded away from a Kuhn-Tucker-point in our algorithm follows from the following theorem.

Theorem 7.2. If for all k the subproblems (4.53a) satisfy the assumptions A1, explained in section 5.2, and A2 explained in section 5.3, then ΔJ_{kz} is bounded away from zero if u_k is not a Kuhn-Tucker-point of (4.1)-(4.4). This result holds also if the regularization method proposed in section 6.2 is used.

□

Proof. Let the row vectors $\lambda(t)$ satisfy

$$\lambda(N)=\ell_x(N)+\sum_{i\in I(N)}(\mu_k)_i h_x^i(N) \tag{7.15a}$$

$$\lambda(t)=\ell_x(t)+\lambda(t+1)f_x(t)+\sum_{i\in I(t)}(\mu_k)_i h_x^i(t)-$$

$$-(\ell_u(t)+\lambda(t+1)f_u(t)+\sum_{i\in I(t)}(\mu_k)_i h_u^i(t))\beta_t \tag{7.15b}$$

where μ_k are the Lagrange multipliers obtained when solving the k:th quadratic subproblem (4.53). Using (7.13a) we get the equality

$$0=-\lambda(t+1)\Delta x(t+1)+\lambda(t+1)f_x(t)\Delta x(t)+\lambda(t+1)f_u(t)\Delta u(t) \qquad (7.16)$$

If we for t=0,...,N-1 add (7.16) to (7.12) we get

$$\Delta J_{zk}=(\ell_x(N)-\lambda(N)+\sum_{i\in I(N)}\bar{\mu}_i h_x^i(N)I(h^i(N)>0))\Delta x(N)+$$

$$+\sum_{t=0}^{N-1}\{(\ell_x(t)+\lambda(t+1)f_x(t)-\lambda(t))\Delta x(t)+(\ell_u(t)+\lambda(t+1)f_u(t))\Delta u(t)+$$

$$+\sum_{i\in I(t)}\bar{\mu}_i(h_x^i(t)\Delta x(t)+h_u^i(t)\Delta u(t))I(h^i(t)>0)\}=$$

$$=(\ell_x(N)+\sum_{i\in I(N)}(\mu_k)_i h_x^i(N)-\lambda(N)+$$

$$+\sum_{i\in I(N)}(\bar{\mu}_i\cdot I(h^i(N)>0)-(\mu_k)_i)h_x^i(N))\Delta x(N)+$$

$$+\sum_{t=0}^{N-1}\{(\ell_x(t)+\lambda(t+1)f_x(t)+\sum_{i\in I(t)}(\mu_k)_i h_x^i(t)-\lambda(t))\Delta x(t)+$$

$$+(\ell_u(t)+\lambda(t+1)f_u(t)+\sum_{i\in I(t)}(\mu_k)_i h_u^i(t))\Delta u(t)+$$

$$+\sum_{i\in I(t)}(\bar{\mu}_i\cdot I(h^i(t)>0)-(\mu_k)_i)(h_x^i(t)\Delta x(t)+h_u^i(t)\Delta u(t))\}$$

Now, $\Delta u(t)$ is given by the solution to the quadratic subproblem (4.53). Using (5.62) we can then write $\Delta u(t)$ as

$$\Delta u(t)=\delta u(t)-\beta_t\Delta x(t)$$

and using (7.15) we get

$$\Delta J_{zk}=\sum_{i\in I(N)}(\bar{\mu}_i I(h^i(N)>0)-(\mu_k)_i)h_x^i(N))\Delta x(N)+$$

$$+\sum_{t=0}^{N-1}\{(\ell_u(T)+\lambda(t+1)f_u(t)+\sum_{i\in I(t)}(\mu_k)_i h_u^i(t))\delta u(t)+$$

$$+ \sum_{i\in I(t)} (\bar{\mu}_i I(h^i(t)>0)-(\mu_k)_i)(h_x^i(t)\Delta x(t)+h_u^i(t)\Delta u(t)\} \qquad (7.17)$$

Using the fact that $\Delta x(t)$ and $\Delta u(t)$ satisfy the constraint (4.53d) we have

$$h_x^i(t_i)\Delta x(t_i)+h_u^i(t_i)\Delta u(t_i)\leq -h^i(t_i) \qquad (7.18)$$

If $h^i(t_i)>0$, then the left hand side of (7.18) must be negative, and using (7.10) we obtain

$$(\bar{\mu}_i I(h^i(t)>0)-(\mu_k)_i)(h_x^i(t_i)\Delta x(t_i)+h_u^i(t_i)\Delta u(t_i))\leq 0 \qquad (7.19)$$

If $h^i(t_i)\leq 0$ and $(\mu_k)_i>0$, we have equality in (7.18) and the inequality (7.19) is again obtained. Hence the inequality (7.19) holds for all i, and from (7.17) we get

$$\Delta J_{zk}\leq \sum_{t=0}^{N-1} (\ell_u(t)+\lambda(t+1)f_u(t)+\sum_{i\in I(t)} (\mu_k)_i h_u^i(t))\delta u(t) \qquad (7.20)$$

Comparing (4.53) and (5.1) and then using (5.62) we obtain

$$\delta u(t)=-C_t^{-1}(\ell_u(t)+\lambda(t+1)f_u(t)+\sum_{i\in I(t)} (\mu_k)_i h_u^i(t))^T. \qquad (7.21)$$

Here C_t is the matrix $C_t=\tilde{\Theta}_{uu}(t)+D(t)+f_u^T(t)\tilde{W}_{xx}(t+1)f_u(t)$ where $\tilde{W}_{xx}(t+1)$ is the solution to the Riccati equation for problem (4.53).

So

$$\Delta J_{zk}\leq -\sum_{t=0}^{N-1} \delta u^T(t)C_t\delta u(t) \qquad (7.22)$$

Hence $\Delta J_{zk}<0$ if $\delta u(t)\neq 0$ for any t because C_t is positive definite.

□

7.5 Local convergence rate.

After having established convergence we will now examine the convergence rate. In section 4.3.3 we found that Wilson's method applied to problem (7.6) generates the search direction $\Delta u(t)$, $t=0,\ldots,N-1$ in the control space, where $\Delta u(t)$ is the solution to (4.53). But Wilson's method is obtained by using Newton's method for solving the Kuhn-Tucker necessary conditions (see the end of chapter 2). Newton's method has quadratic local convergence (cf p. 250 in Dahlquist and Björck (1974)) so the method that we have proposed in section 4.3.3 for solving the problem (7.6) should have quadratic local convergence if $\alpha_k=1$ in (7.8). That this really is the case under certain conditions is assured by the following theorem.

Theorem 7.2. Let $z^*(0)=(u^*(0)^T,\ldots,u^*(N-1)^T)^T$ be the solution of problem (7.6) and let μ^* be the multipliers that satisfy equation (7.7). Assume that the following conditions hold.

C7.1: At the point $(z^*(0),\mu^*)$ the CLQ-problem satisfies assumption A2 in section 5.3.

C7.2: The strict complementarity condition holds for the constraints (7.6d), i.e. if $\mu_i^*=0$ then $h^i(x^*(t_i),u^*(t_i))<0$ and if $h^i(x^*(t_i),u^*(t_i))=0$ then $\mu_i^*>0$.

C7.3: At the point $(z^*(0),\mu^*)$ the CLQ-problem (4.53) satisfies the Assumption A1 described in section 5.2, i.e. there exists an $\alpha>0$ such that

$$z^T(\tilde{Q}_{uu}(t)+f_u^T(t)\tilde{W}_{xx}(t+1)f_u(t))z\geq\alpha|z|^2 \qquad (7.23)$$

where $\tilde{W}_{xx}(t+1)$ is defined by

$$\tilde{W}_{xx}(N)=\tilde{Q}_{xx}(N) \qquad (7.24a)$$

$$\tilde{W}_{xx}(t)=\tilde{Q}_{xx}(t)+f_x(t)\tilde{W}_{xx}(t+1)f_x(t)-$$

$$-(\tilde{Q}_{xu}(t)+f_x^T(t)\tilde{W}_{xx}(t+1)f_u(t)\cdot$$

$$\cdot(\tilde{Q}_{uu}(t)+f_u^T(t)\tilde{W}_{xx}(t+1)f_u(t))^{-1}\cdot$$

$$\cdot(\tilde{Q}_{ux}(t)+f_u^T(t)\tilde{W}_{xx}(t+1)f_x(t)) \qquad (7.24b)$$

and the matrices $\tilde{Q}_{xx}(t)$, $\tilde{Q}_{xu}(t)$ and $\tilde{Q}_{uu}(t)$ are given by (4.52). (The condition (7.23) must be valid without using the regularization method proposed in section 6.2.).

Then there exist constants ε_z and ε_μ such that if $|z^*(0)-z_k(0)|<\varepsilon_k$ and $|\mu^*-\mu_{k-1}|<\varepsilon_\mu$ then the sequence $\{z_k(0)\}$ generated by the algorithm defined in section 9.2, with $\alpha_k=1$, converges quadratically to $z^*(0)$, i.e. there exists a K such that

$$|z_{k+1}(0)-z^*(0)|\leq K|z_k(0)-z^*(0)|^2$$

Proof. In section 4.3.3 we showed that the algorithm is equivalent to Wilson's method. The result then follows from pp. 252-256 in Bertsekas (1982) and Theorem 5.1.

□

The theorem shows that close to the solution our method has quadratic convergence if $\alpha_k=1$. In practice, however, α_k is given by the step length rule. That the Watchdog technique does not destroy the rate of convergence is assured by the following lemma from Chamberlain et al. (1979).

Lemma 7.1. Assume that the three conditions of Theorem 7.3 hold and that condition (7.10) holds. Then there exist positive constants ε_1, ε_2, ε_3 such that for any sequence satisfying

$$|z_k(0)-z^*(0)|<\varepsilon_1$$

$$|\lambda_k-\lambda^*|<\varepsilon_2$$

$$\frac{|z_{k+1}(0)-z^*(0)|}{|z_k(0)-z^*(0)|}<\varepsilon_3$$

$h^i(x_k(t_i),u_k(t_i))+h^i(t_i)\Delta x(t) + h^i_u(t_i)\Delta u(t)\leq 0$

for each k, the Watchdog technique accepts the point $z_{k+1}(0)$.

Proof: See the proof of Theorem 2 in Chamberlain et al. 1979.

□

We conclude from the two theorems above that our algorithm using the Watchdog step length technique has global convergence as well as fast convergence near the optimum.

8. SIMPLE CONTROL CONSTRAINTS

8.1 Introduction

The method for solving constrained quadratic problems proposed in chapter 3 is not useful when the number of active constraints is large, because then the matrix A defined by (3.9) is also large. Since A is usually not sparse, the computational burden will be heavy. It is easy to find optimal control problems where many of the constrains (4.4) are active at the solution. Examples are bang-bang optimal control problems or problems where some of the states touch their boundaries at almost every point in time.

In section 1.5 of Bertsekas (1982) a method is presented which efficiently solves problems of the type

$$\text{minimize } \ell(z) \tag{8.1a}$$

$$\text{subject to } \underline{z} \leq z \leq \bar{z} \tag{8.1b}$$

where $\underline{z}$ and $\bar{z}$ contain lower and upper bounds on the variables in z. Constraints of the type (8.1b) are in Bertsekas (1982) called "simple constraints". Another common name is "box constraints". Since our approach is to consider optimal control problems as non-linear programming problems in the control u(t), we can easily adapt Bertsekas' method to optimal control problems when there are simple constraints on the controls only. The details of the resulting method is given in section 8.2, which also contains proof of convergence. In section 8.3 we propose a hybrid version of Bersekas method and the method described in section 4.3. The method seems to be promising as shown by the examples in chapter 11.

8.2 A fast algorithm for optimal control problems with simple control constraints.

In this section we consider the following special case of problem (4.1)-(4.4).

$$\underset{u}{\text{minimize}}\ J(u)=\ell(x(N),N)+\sum_{t=0}^{N-1}\ell(x(t),u(t),t) \tag{8.2a}$$

$$\text{subject to } x(t+1)=f(x(t),u(t),t),\ t=0,\ldots,N-1 \tag{8.2b}$$

$$x(0)=x_0 \tag{8.2c}$$

$$0\leq u(t) \quad t=0,\ldots,N-1 \tag{8.2d}$$

The generalization of the method to handle box constraints on the controls is trivial. In section 4.3.1 it was shown how to use Newton's method on (8.2a-c). If we just take the projection on the constraints (8.2d) of the direction generated by Newton's method, we might converge to a point which is not a Kuhn-Tucker-point of the problem. On the other hand if we project the steepest descent direction onto the constraints (8.2d) we get a method that converges to a Kuhn-Tucker-point, but the convergence might be slow. Bertsekas' idea is to divide the variables into two groups at each iteration. The first group consists of the variables that are on the boundaries and will probably remain there and the variables that will probably hit their boundaries during the next iteration. The second group consists of the variables that are clearly inside the feasible region or those varaibles which are supposed to move away from the boundary during the next iteration. Then, at each iteration we construct a search direction which is a steepest descent direction in the variables in the first group and a Newton direction in the variables in the second group. Then during the line search the new point is projected into the feasible area.

Let us denote the first group at iteration k by I_k. In the following we adopt the notation z(0) defined by (4.6) to denote the sequence u(t),t=0,...,N-1.

For a given $\varepsilon_k>0$ we define the first group I_k at iteration k as

$$I_k=\{i:(z(0))_i<\varepsilon_k \text{ and } (V_z(0))_i>0\} \tag{8.3}$$

Here $(z)_i$ stands for the i:th component of the vector z. $V_z(0)$ is defined from (4.12a) and (4.12c). Now using the steepest descent direction for the variables indexed by I_k gives

$$(\Delta z(0))_i=-(V_z(0)),\ i\in I_k \tag{8.4}$$

For the elements in the second group examine the LQ-problem (4.20). From Theorem 4.1 we know that the solution (in $\Delta u(t)$) to (4.20) is also given (in $\Delta z(0)$) by (4.10) which is the solution to the problem

$$\underset{\Delta z(0)}{\text{minimize}} \quad V_z(0)\Delta z(0)+\frac{1}{2}\Delta z^T(0)V_{zz}(0)\Delta z(0) \tag{8.5}$$

The Newton direction in the free varaibles is found by solving problem (9.5) under the constraint

$$(\Delta z(0))_i=0 \quad I\in I_k \tag{8.6}$$

and after this solution is obtained we use (8.4) to get the final search direction. However, minimizing (8.5) under the constraint (8.6) is the same problem as solving the LQ-problem (4.20) under the constraint

$$(\Delta u(t))_j=0 \quad j\in I_k(t),t=0,\ldots,N-1 \tag{8.7}$$

where $I_k(t)$ denotes the elements of $\Delta u(t)$ corresponding to elements of $\Delta z(0)$ in I_k.

We will now show in detail how to solve the LQ-problem. To begin with, (8.7) will be replaced by the more general constraints

$$h+h_u\Delta u(p)=0. \tag{8.8}$$

Here p is a given time, h is a vector and h_u is a full rank matrix. We derive the more general result using the constraint (8.8) because we will need it in the next section. Later in this section the result will be specialized to the constraints (8.7).

As in section 4.3.1 we solve our new problem by putting up the subproblems (4.21a). The minimum of these subproblems is according to (4.22) given by

$$J^*(\Delta x(t),t)=a(t)+W_x(t)\Delta x(t)+\frac{1}{2}\Delta x^T(t)W_{xx}(t)\Delta x(t) \qquad (8.9)$$

where at least for t>p, a(t), $W_x(t)$ and $W_{xx}(t)$ are given by (4.24) and (4.27). For t=p we define the Lagrangian

$$L(\Delta x(p),\Delta u(p),\mu,p)=J^*\{f_x(p)\Delta x(p)+f_u(p)\Delta u(p),p+1\}+$$

$$+\ell_x(p)\Delta x(p)+\ell_u(p)\Delta u(p)+\frac{1}{2}\{\Delta x^T(p)Q_{xx}(p)\Delta x(p)+$$

$$2\Delta x^T(p)Q_{xu}(p)\Delta u(p)+\Delta u^T(p)Q_{uu}(p)\Delta u(p)\}+\mu^T(h+h_u\Delta u(p))=$$

$$=a(p+1)+\mu^T h+(\ell_x(p)+W_x(p+1)f_x(p))\Delta x(p)+$$

$$+(\ell_u(p)+W_x(p+1)f_u(p)+\mu^T h_u)\Delta u(p)+$$

$$+\frac{1}{2}\{\Delta x^T(p)A_p\Delta x(p)+2\Delta x^T(p)B_p\Delta u(p)+\Delta u^T(p)C_p(\Delta u(p))\} \qquad (8.10)$$

where A_p, B_p and C_p are defined by

$$A_p=Q_{xx}(p)+f_x^T(p)W_{xx}(p+1)f_x(p) \qquad (8.11a)$$

$$B_pQ_{xu}(p)+f_x^T(p)W_{xx}(p+1)f_u(p) \qquad (8.11b)$$

$$C_p=Q_{uu}(p)+f_u^T(p)W_{xx}(p+1)f_u(p) \qquad (8.11c)$$

The value of $\Delta u(p)$ that minimizes (8.10) is given by

$$\Delta u(p)=-C_p^{-1}\{(\ell_u(p)+W_x(p+1)f_u(p))^T+B_p^T\Delta x(\ell)+h_u^T\mu\} \qquad (8.12)$$

where we choose μ so that the constraint (8.8) is satisfied. Hence

$$0=h+h_u\Delta u(p)=h-h_uC_p^{-1}((\ell_u(p)+W_x(p+1)f_u(p))^T+B_p^T\Delta x(p))-h_uC_p^{-1}h_u^T\mu$$

which gives

$$\mu=(h_uC_p^{-1}h_u^T)^{-1}(h-h_uC_p^{-1}(\ell_u(p)+W_x(p+1)f_u(p)+B_p\Delta x(p))) \qquad (8.13)$$

This value of μ is inserted in (8.12) which gives us

$$\Delta u(p)=-C_p^{-1}\{h_u^T(h_uC_p^{-1}h_u^T)^{-1}\ h+$$

$$+(I-h_u^T(h_uC_p^{-1}h_u^T)^{-1}h_uC_p^{-1})((\ell_u(p)+W_x(p+1)f_u(p))^T+$$

$$+B_p\Delta x(p))\} \qquad (8.14)$$

If we define the matrix I_o as

$$I_o=I-C_p^{-1}h_u^T(h_uC_p^{-1}h_u^T)^{-1}h_u \qquad (8.15a)$$

and $\delta u(p)$ and β_p as

$$\delta u(p)=-C_p^{-1}h_u^T(H_uC_p^{-1}h_u^T)^{-1}h-I_oC_p^{-1}(\ell_u(p)+W_x(p+1)f_u(p))^T \qquad (8.15b)$$

$$\beta_p=I_oC_p^{-1}\cdot B_p^T \qquad (8.15c)$$

then equation (9.14) reduces to

$$\Delta u(p)=\delta u(p)-\beta_p\cdot\Delta x(p). \qquad (8.16)$$

If we insert (8.16) in (8.10) (using (8.8)) we get

$$J^*(\Delta x(p),p)=a(p+1)+(\ell_x(p)+W_x(p+1)f_x(p))\Delta x(p)+$$

$$+(\ell_u(p)+W_u(p+1)f_u(p))(\delta u(p)-\beta_p\Delta x(p))+$$

$$+ \frac{1}{2}(\Delta x^T(p)A_p\Delta x(p)+2\Delta x^T(p)B_p(\delta u(p)-\beta_p\Delta x(p))+$$

$$+(\delta u(p)-\beta_p\Delta x(p))^T C_p(\delta u(p)-\beta_p\Delta x(p))=$$

$$=a(p+1)+(\ell_u(p)+W_x(p+1)f_u(p))\cdot\delta u(p)+$$

$$+ \frac{1}{2}\delta u^T(p)C_p\delta u(p)+$$

$$+(\ell_x(p)+W_x(p+1)f_x(p)-(\ell_u(p)+W_x(p+1)f_u(p)\beta_p+$$

$$+\delta u^T(p)B_p^T-\delta u(p)C_p\beta_p)\Delta x(t)+$$

$$+ \frac{1}{2}\Delta x^T(p)(A_p-B_p\beta_p-\beta_p^TB_p^T+\beta_pC_p\beta_p)\Delta x(p) \qquad (8.17)$$

Identifying the coefficients we then get

$$a(p)=a(p+1)+(\ell_u(p)+W_p(p+1)f_u(p))\delta u(p)+$$

$$+ \frac{1}{2}\delta u^T(p)C_p\delta u(p) \qquad (8.18a)$$

$$W_x(p)=\ell_x(p)+W_x(p+1)f_x(p)-(\ell_u(p)+W_x(p+1)f_u(p))\beta_p))\beta_p+$$

$$+\delta u^T(p)(B_p^T-C_p\beta_p) \qquad (8.18b)$$

$$W_{xx}(p)=A_p-B_p\beta_p-\beta_p^TB_p+\beta_p^TC_p\beta_p=$$

$$=A_p-B_p(I_oC_p^{-1}+C_p^{-1}I_o^T-C_p^{-1}I_o^TC_pI_oC_p^{-1})B_p \qquad (8.18c)$$

We now return to the simple constraints defined by (8.7). Specializing the formulas we get,

$$h=0 \qquad (8.19a)$$

and

$$h_u=(e_{r1},e_{r2},\ldots,e_{rp})^T \tag{8.19b}$$

where e_r is the r:th column in the unit matrix. We assume that $r_1, r_2,\ldots,r_p$ are the indices in $I_k(p)$. (See (8.7)).

The value of $\delta u(t)$ in (8.15b) becomes

$$\delta u(t)=-I_o\cdot C_p^{-1}(\ell_u(p)+W_x(p+1)f_u(p))^T \tag{8.20}$$

The difference equation for $W_x(p)$ reduces to

$$W_x(p)=\ell_x(p)+W_x(p+1)f_x(p)-(\ell_u(p)+W_x(p+1)f_u(p))\beta_p. \tag{8.21}$$

The variables which do not belong to I_k(or $I_k(t)$,t=0,...,N-1) are now calculated from

$$\Delta x(0)=0 \tag{8.22a}$$

$$\left.\begin{aligned}\Delta u(t)&=\delta u(t)-\beta_t\Delta x(t) \qquad (8.22b)\\ x(t+1)&=f_x(t)\Delta x(t)+f_u(t)\Delta u(t) \qquad (8.22c)\end{aligned}\right\} \quad t=0,\ldots,N-1$$

After these calculations we set

$$(\Delta u(t))_i=-(\ell_u(t)+V_x(t+1)f_u(t))_i;\ i\in I_k(t) \tag{8.23}$$

to obtain the steepest descent direction among the variables indexed by I_k.

The step length procedure is modified in the following way: Consider the following controls

$$u(t,\alpha)=[u_k(t)+\alpha\Delta u(t)]^+;t=0,\ldots,N-1 \tag{8.24}$$

where the expression $u=[\bar{u}]^+$ means $u_i=\max(0,\bar{u}_i)$ where $u=(u_1,\ldots,u_{n_u})^T$ and $\bar{u}=(\bar{u}_1,\ldots,u_{n_u})^T$. Hence the operator $[\quad]^+$ defines the projection of $u_k(t)+\Delta u(t)$ into the feasible region

(8.2d). From proposition 1.35 on page 78 in Bertsekas (1982) we then know that $u_k(t), t=0,\ldots,N-1$ is a Kuhn-Tucker-point of the problem (8.2) if and only if

$$u(t,\alpha)=u_k(t), \alpha \geq 0, t=0,\ldots,N-1 \tag{8.25}$$

Also, if $u_k(t), t=0,\ldots,N-1$, is not a Kuhn-Tucker point of (8.2), there is an $\bar{\alpha}>0$, such that $J(u(\alpha))<J(u_k)$ when $\alpha \in (0,\bar{\alpha})$. Hence $\Delta u(t)$ defines a descent direction for the problem defined by (8.2).

The algorithm outlined in Bertsekas (1982) for the next control sequence now gives

$$u_{k+1}(t)=u(t,\alpha_k) \tag{8.26}$$

where $$\alpha_k=\beta^{m_k} \tag{8.27}$$

and m_k is the smallest nonnegative integer m such that

$$J(u(\beta^m)) \leq J(u_k)+\sigma \cdot \Delta J(\beta^m) \tag{8.28}$$

where $\Delta J(\beta^m)$ is defined as

$$\Delta J(\beta^m)=\sum_{t=0}^{N-1} \left(\beta^m \cdot \sum_{j \notin I_k(t)} \frac{\partial J}{\partial (u(t))_j} \cdot (\Delta u(t))_j + \right.$$

$$\left. + \sum_{j \in I_k(t)} \frac{\partial J}{\partial (u(t))_j} (u(t,\alpha)-u_k(t))_j \right) \tag{8.29}$$

Now we assume that the problem (8.12) satisfies the following conditions.

C8.1. The derivatives $V_z(0)$ are Lipschitz continuous on each bounded set in the control-space.

C8.2. There exist positive scalars λ_1 and λ_2 and nonnegative integers q_1 and q_2 such that

$$\lambda_1 w_k^{q_1}|z|^2 \leq z^T V_{zz}(0) z \leq \lambda_2 w_k^{q_2}|z|^2$$

for all z in the control-space and where $w_k=|z_k(0)-[z_k(0)-MV_z^T(0)]^+|$ for a positive definite diagonal matrix M.

We have the following global convergence condition

Theorem 8.1. If the conditions C8.1 and C8.2 above are satisfied, then every limit point of the control sequence $\{u_k\}$ generated by the iteration (8.26) is Kuhn-Tucker point of the problem (8.2).

Proof: Follows from Bertsekas (1982), p. 86.

□

8.3 A hybrid algorithm.

In this section we propose an algorithm for solving constrained nonlinear optimal control problems, where the constraints consist of both general constraints and box constraints. The idea is to handle the general constraints using our basic method and the box constraints using the technique given in the previous section.

We therefore consider the problem

$$\underset{u}{\text{minimize}} \quad J(u)=\ell(x(N),N)+\sum_{t=0}^{N-1}\ell(x(t),u(t),t) \tag{8.30a}$$

$$\text{subject to} \quad x(t+1)=f(x(t),u(t),t) \tag{8.30b}$$

$$x(0)=x_0 \tag{8.30c}$$

$$h^i(x(t_i),u(t_i))=0,\ i=0,\ldots,p \tag{8.30d}$$

$$0\leq u(t),\ t=0,\ldots,N-1 \tag{8.30e}$$

The constraints (8.30e) are now treated in the way described in the previous section. Hence we solve the Riccati equation (8.18) and then calculate $\Delta u_0(t)$ according to (8.22). After that we adjust the variables in $I_k(t)$ such that

$$(\delta u(t))_i=\Delta u(t)_i=-(u(t))_i \text{ if } i\in I_k(t) \tag{8.31}$$

Note that we can not use $\ell_u(t)+V_x(t+1)\ f_u(t)$ to define the index-sets $I_k(t)$. Instead we use the derivatives of the Lagrangian for identifying the possible active box constraints. As Lagrange multipliers we choose the ones obtained at iteration k-1. Hence the index sets $I_k(t)$ are now defined by

$$I_k(t)=\{j:(u(t))_j<\epsilon_k \text{ and } (\ell_u(t)+\tilde{V}_x(t+1)f_u(t)+\gamma_u(t))_j>0\} \tag{8.32}$$

Here $\tilde{V}_x(t)$ is given as in section 4.3.3 by (4.51) and $\gamma(t)$ by (4.50) except that only the general constraints (8.29d) are used. From section 5.4 we know that solving the Riccati equation (8.18) is actually a way to factorize a matrix. Applying the arguments in connection with equations (5.55) to our box constraints we find that in this case the factorization is defined by the matrices $f_x(t)$, $f_u(t)$, $I_o\cdot C_p^{-1}$ and β_t. Hence C_p^{-1} is replaced by $I_oC_p^{-1}$. The only change when we calculate the influence of the general constraints is that for (8.29) the equations (5.57d) and (5.57f) are given by

$$\delta u_i(t_i)=-I_oC_{t_i}^{-1}h_u^i(t_i)^T \tag{8.33a}$$

$$\delta u_i(t)=-I_oC_t^{-1}f_u^T(t)W_i(t+1),t=0,\ldots,t_i-1. \tag{8.33b}$$

The new control is then given by

$$u_{k+1}(t)=u_k(t)+\alpha_k \Delta u(t) \qquad (8.34)$$

where α_k is chosen according to the rules for the Watchdog technique given in section 7. In Chapter 11 we will show examples where the method is used.

9. SUMMARY AND IMPLEMENTATION OF THE OPTIMIZATION ALGORITHM

9.1 Introduction

In the previous chapters we have described and discussed different aspects of algorithms for solving static and dynamic optimization problems. The formal development of these aspects has been filled with many long expressions and technical arguments. It is the purpose of the present chapter to summarize this discussion into a single and final algorithm. That will be done in section 9.2. In section 9.3 we discuss the relationship between our algorithm and other algorithms. Our algorithm has been implemented on a DEC-20 computer. A flow chart for the organization of this program is given in section 9.4, while details of the user aspects of the program are given in section 9.5.

A computer program has also been developed for the static optimization problem (2.11) which is suitable for large scale problems. This program is described in section 9.6. In the following two chapters some numerical experience with the computer programs, when applied to various problems, will be reported.

9.2 The algorithm for solving constrained optimal control problems.

The fundamental idea behind the development in this thesis has been to formulate the optimal control problem (4.1)-(4.4) as a static optimization problem, and to derive the Newton-type algorithm in a consistent manner. Clearly, in order to obtain an efficient algorithm, the inherent structure in the problem formulation must be utilized when the problem is viewed as a static optimization problem.

This means that the developed algorithm from a conceptual point of view is Wilson's method applied to the particular problem (4.1)-(4.4). The control signals $u(t), t=0,..,N-1$ are considered as free variables. The sequence of state vectors $x(t)$ is eliminated using (4.1). From Theorem 4.1 we know that then the $\Delta u(t), t=0,...,N-1$ given by the solution to (4.53) define the

Wilson direction for the problem (4.1)-(4.4). The problem is thus to solve (4.53), and we shall now summarize the discussion in chapter 5, sections 6.2 and 8.3 of how to solve this equation. This procedure will consequently include both regularization and techniques for handling simple constraints.

Assume that we initially are given a control sequence $u_0(t)$, $t=0,\dots,N-1$ and a p-dimensional vector of nonnegative multipliers μ_{-1}.

I. At step k of the algorithm we have a control sequence $u_k(t), t=0,\dots,N-1$. Using equations (4.1) and (4.2) we can calculate and store the sequence of states $x_k(t), t=0,\dots,N$ that corresponds to this control sequence.

II. Let $\tilde{V}_x(N)$ and $\tilde{V}_{xx}(N)$ be given by

$$\tilde{V}_x(N)=\ell_x(N)+\gamma_x(N,\mu_{k-1}) \tag{9.1a}$$

$$W_x(N)=\ell_x(N) \tag{9.1b}$$

$$\tilde{W}_{xx}(N)=\ell_{xx}(N)+\gamma_{xx}(N,\mu_{k-1}) \tag{9.1c}$$

III. For $t=N-1,\dots,0$ solve the difference equations

$$\tilde{V}_x(t)=\ell_x(t)+\tilde{V}_x(t+1)f_x(t)+\gamma_x(t,\mu_{k-1}) \tag{9.2a}$$

$$W_x(t)=\ell_x(t)+W_x(t+1)f_x(t)-(\ell_u(t)+W_x(t+1)f_u(t))\beta_t \tag{9.2b}$$

$$\tilde{W}_{xx}(t)=A_t-B_t(I_o C_t^{-1}+C_t^{-1}I_o^T+C_t^{-1}I_o^T C_t I_o C_t^{-1})B_t \tag{9.2c}$$

where

$$A_t=\ell_{xx}(t)+\tilde{V}_x(t+1)f_{xx}(t)+$$

$$+f_x^T(t)\tilde{W}_{xx}(t+1)f_x(t)+\gamma_{xx}(t,\mu_{k-1}) \tag{9.3a}$$

$$B_t=\ell_{xu}(t)+\tilde{V}_x(t+1)f_{xu}(t)+$$

$$+f_x^T(t)\tilde{W}_{xx}(t+1)f_u(t)+\gamma_{xu}(t,\mu_{k-1}) \tag{9.3b}$$

$$C_t=D(t)+\ell_{uu}(t)+\tilde{V}_x(t+1)f_{uu}(t)+$$

$$+f_u^T(t)\tilde{W}_{xx}(t+1)f_u(t)+\gamma_{uu}(t,\mu_{k-1}) \tag{9.3c}$$

$$\beta_t=I_o\cdot C_t^{-1}\cdot B_t^T \tag{9.3d}$$

and where I_o is defined from the active simple constraints described in section 8.4. In (9.3c) the matrix D(t) is chosen in accordance with the ideas described in section 8.2. Store the values

$$\delta u_0(t)=-I_o C_t^{-1}(\ell_u(t)+W_x(t+1)f_u(t))^T \tag{9.4a}$$

$$\beta_t \tag{9.4b}$$

$$I_o C_t^{-1} \tag{9.4c}$$

For every active simple constraint at time t put $(\delta u_0(t))_i$ at its boundary.

IV. Put $\Delta x_0(t)=0$ and for t=0,...,N-1 calculate and store $\Delta x_0(t+1)$ and $\Delta u_0(t)$ given by

$$\Delta u_0(t)=\delta u_0(t)-\beta_t\Delta x_0(t) \tag{9.5a}$$

$$\Delta x_0(t+1)=f_x(t)\Delta x_0(t)+f_u(t)\Delta u_0(t) \tag{9.5b}$$

V. Choose the active constraints according to the rules given in section 3.3.
For each active constraint i, do the following calculations:

Store the values:

$$\delta u_i(t)=0 \quad t>t_i \tag{9.6a}$$

$$\delta u_i(t_i)=-I_o C_t^{-1} h_u^i(t_i) \tag{9.6b}$$

For $t=t_i-1,\ldots,0$ solve the difference equation

$$W_x^i(t)=W_x^i(t+1)(f_x(t)-f_u(t)\beta_t) \tag{9.7a}$$

with the boundary value

$$W_x^i(t_i)=h_x^i(t_i)-h_u^i(t_i)\beta_{t_i} \tag{9.7b}$$

and during these calculations compute and store

$$\delta u_i(t)=-I_o C_t^{-1} f_u^T(t) W_x^i(t+1)^T. \tag{9.8}$$

Put $\Delta x_i(0)=0$ and for $t=0,\ldots,N-1$ calculate $\Delta x_i(t+1)$ and $\Delta u_i(t)$ from

$$\Delta u_i(t)=\delta u_i(t)-\beta_t \Delta x(t_i) \tag{9.9a}$$

$$\Delta x_i(t+1)=f_x(t)\Delta x_i(t)+f_u(t)\Delta u_i(t) \tag{9.9b}$$

During these calculations compute and store the element a_{ij} given by

$$a_{ij}=h_x^j(t_j)\Delta x_i(t_j)+h_u^j(t_j)\Delta u_i(t_j) \tag{9.10}$$

for every constraint j in the active set.

The elements a_{ij} in (9.10) define the matrix A in (3.9) in section 3.3. The elements of the right hand side vector in (3.10) are calculated as

$$d_i=h^i(t_i)+h_x^i(t_i)\Delta x_0(t_i)+h_u^i(t_i)\Delta u_0(t_i) \qquad (9.11)$$

After having solved (3.8) for the current set of active constraints, put $\Delta x(0)=0$ and solve the equations

$$\Delta u(t)=\delta u_0(t)+\sum_{i=1}^{p'}\tilde{\mu}_i^!\cdot\delta u_i(t)-\beta_t\Delta x(t) \qquad (9.12a)$$

$$\Delta x(t+1)=f_x(t)\Delta x(t)+f_u(t)\Delta u(t) \qquad (9.12b)$$

During these calculations, calculate the values of ΔJ_{zk} in (7.7) and also check if the non-active constraints of (4.53d) are violated.

If the matrix A becomes singular during these calculations, the assumption A2 given in section 5.2 is violated. Then special measures must be taken (cf section 6.2).

VI. Let $\Delta x(t)$ and $\Delta u(t)$ be the result of the previous step and μ_k the value of the Lagrange multipliers for the constraints (4.53d) obtained at the solution.
If $\sum_{t=0}^{N-1}|\Delta u(t)|^2<\varepsilon$ for some given $\varepsilon>0$ go to step VIII.

VII. Let

$$u_{k+1}(t)=u_k(t)+\alpha_k\Delta u(t),t=0,\ldots,N-1 \qquad (9.13)$$

where α_k is chosen according to the rules given in section (7.4). Put k=k+1 and go back to step I.

VIII. Accept $u_k(t),t=0,\ldots,N-1$ as the optimal control.

9.3 Relationship to other algorithms

After describing our algorithm for discrete time optimal control problems we will now make a comparison with two other proposed methods in this field, namely the DDP-method in Jacobson and

Mayne (1970) and the new DDP-algorithm by Ohno (1978). We will here consider the version of the DDP-method denoted as small variations in control. Moreover we assume that in Ohno's method, the Newton method is used as the local iteration procedure for finding a solution to the Kuhn-Tucker necessary condition.

These three methods (our method, and the two DDP-methods) all give an increment of the type

$$\Delta u(t)=\delta u(t)-\beta_t \cdot dx(t) \tag{9.14}$$

but the choices of $\delta u(t)$, β_t and $dx(t)$ differ for the methods. We first assume that we have no constraints of the type (4.4). In both the DDP-methods, the backward equations are solved as in (9.2) but in the expressions (9.3), the vector $\tilde{V}_x(t)$ is replaced by $W_x(t)$. Hence it is not necessary to solve the difference equation for $V_x(t)$. The calculation of β_t is made in accordance with (9.3d) for all three methods but again in the expressions for β_t and C_t, $V_x(t)$ is exchanged for $W_x(t)$ for the DDP-methods. This is also the only difference in the calculation of $\delta u(t)$. Another difference appears in the calculation of the value of $dx(t)$ in (9.14). Our algorithm uses $dx(t)=\Delta x(t)$ where $\Delta x(t)$ is obtained from the solution to (9.5). In both the DDP-methods $dx(t)$ is calculated as

$$dx(t)=x_{k+1}(t)-x_k(t) \tag{9.15}$$

where $x_k(t)$ denotes the states corresponding to the controls $u_k(t)$ and $x_{k+1}(t)$ denotes the states corresponding to the controls $u_{k+1}(t)=u_k(t)+\Delta u(t)$. Hence again our method requires the solution of an extra set of difference equations in order to get an exact Newton direction. (If we refrain from using an exact Newton method we could of course also use (9.15)).

To obtain global convergence in our algorithm, we adjust the variable α_k in (9.13) so that a sufficient reduction is obtained in the objective function. In Jacobson and Mayne the new control sequence is given in a similar way, namely by

$$u_{k+1}(t)=u_k(t)+\varepsilon\delta u(t)-\beta_t(x_{k+1}(t)-x_k(t)) \qquad (9.16)$$

where ε is chosen so that a decrease is obtained in the objective function. Ohno's method is only supposed to be applied close to a solution, so he uses

$$u_{k+1}(t)=u_k(t)+\Delta u(t) \qquad (9.17)$$

without any step-size reduction procedure at all (cf Theorem 7.2).

As shown above, our method requires the solution of two extra sets of difference equations. But because the variables in these difference equations are of dimension n_x, the extra computer work for solving these equations is small compared to the solution of the Riccati equation which is $n_x \times n_x$. Also these two extra difference equations are necessary to obtain an exact Newton direction and hence the reason why we easily prove global convergence and locally quadratic convergence.

Assume that we have constraints of the type (4.4). Both the DDP-methods require that these constraints are explicit functions of the controls. If these methods are used on problems which have pure state constraints, the state transition equation must be used once or several times to get a constraint that is an explicit function of u. Both the DDP-methods take care of the constraints in the same way as we handle simple constraints in Section 8.2. In our method, two difference equations are solved for each active constraint (cf (9.7) and (9.9)) and for each set of active constraints we have to solve the difference equation (9.12). Hence, again our method has to solve extra difference equations. On the other hand, our method gives a better estimate of the active constraints, because in the other methods, the choice of active constraints is made when solving the Riccati equation, and this choice cannot be changed during the rest of the iteration. A result of this fact can be an increased number of iterations for the DDP-methods.

The DDP-method with global variations in control has a slightly different structure than the three methods we have discussed so far in this section. At time t, in the backwards iteration, a global minimization in u(t) is made of the function

$$\ell(x_k(t),u(t),t)+W_x(t+1)f(x_k,u(t),t)+$$

$$+\frac{1}{2}\Delta f^T(u(t),t)W_{xx}(t+1)\Delta f(u(t),t) \qquad (9.18)$$

where $\Delta f(u(t),t)$ is the difference

$$\Delta f(u(t),t)=f(x_k(t),u(t),t)-f(x_k(t),u_k(t),t) \qquad (9.19)$$

The u(t) that minimizes (9.18) is denoted by $u^*(t)$. Then the backward equations are solved with $(x_k(t),u^*(t))$ as arguments instead of $(x_k(t),u_k(t))$. The new controls are then given by

$$u_{k+1}(t)=u^*(t)-\beta_t dx(t) \qquad (9.20)$$

where β_t and dx(t) are calculated in the usual manner. The possibility of convergence of this method is increased by the following step-size adjustment procedure: First the control (9.20) is used over the whole time horizon. If a sufficient reduction in the cost function is obtained, the new control sequence is accepted and the iterations proceed. If a reduction is not obtained, the controls (9.20) are used on smaller and smaller time intervals until this happens. This step-size adjustment method is also used in the continuous-time DDP-method.

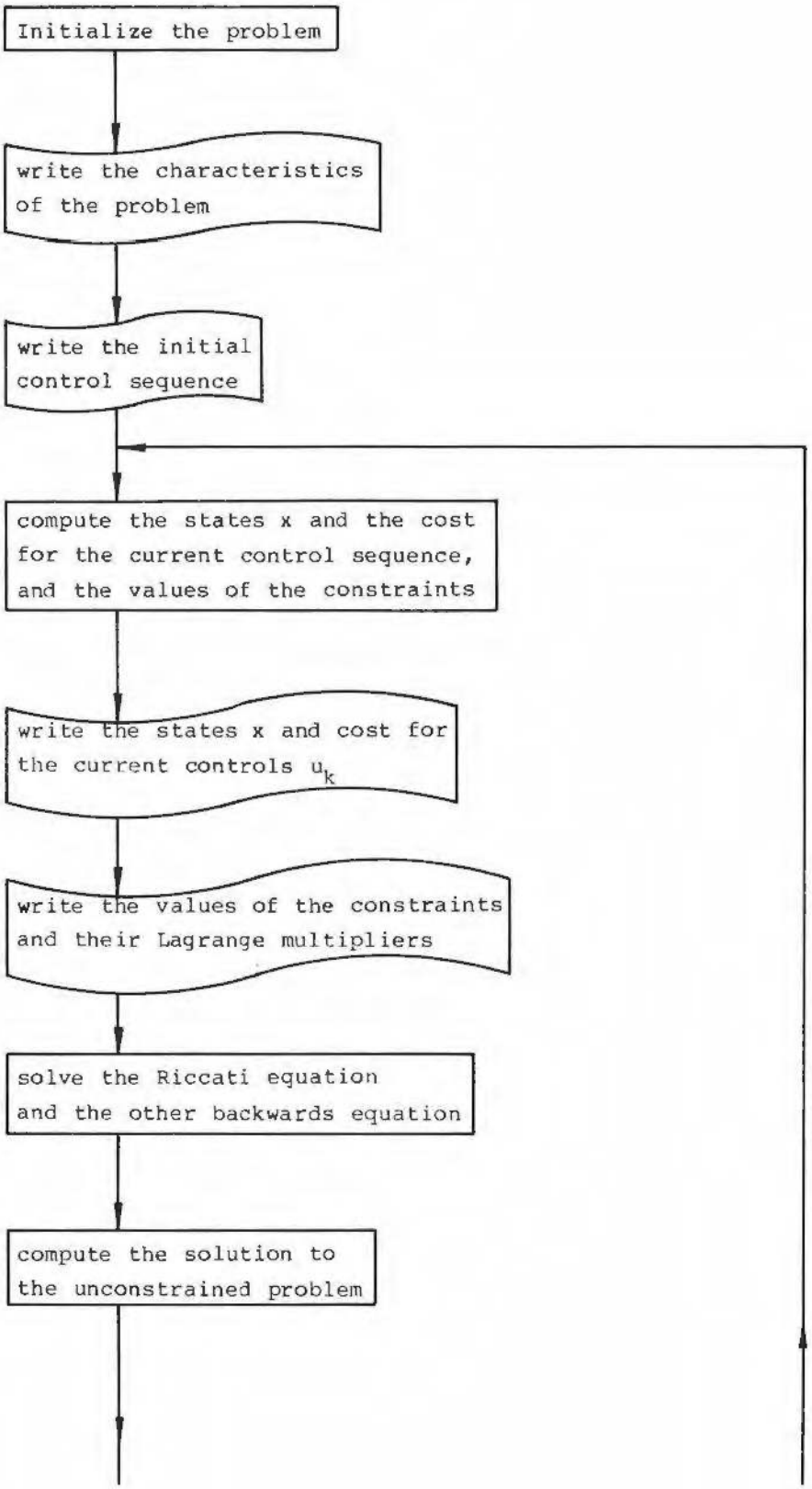
9.4 A flow chart of a computer program for implementing the algorithm

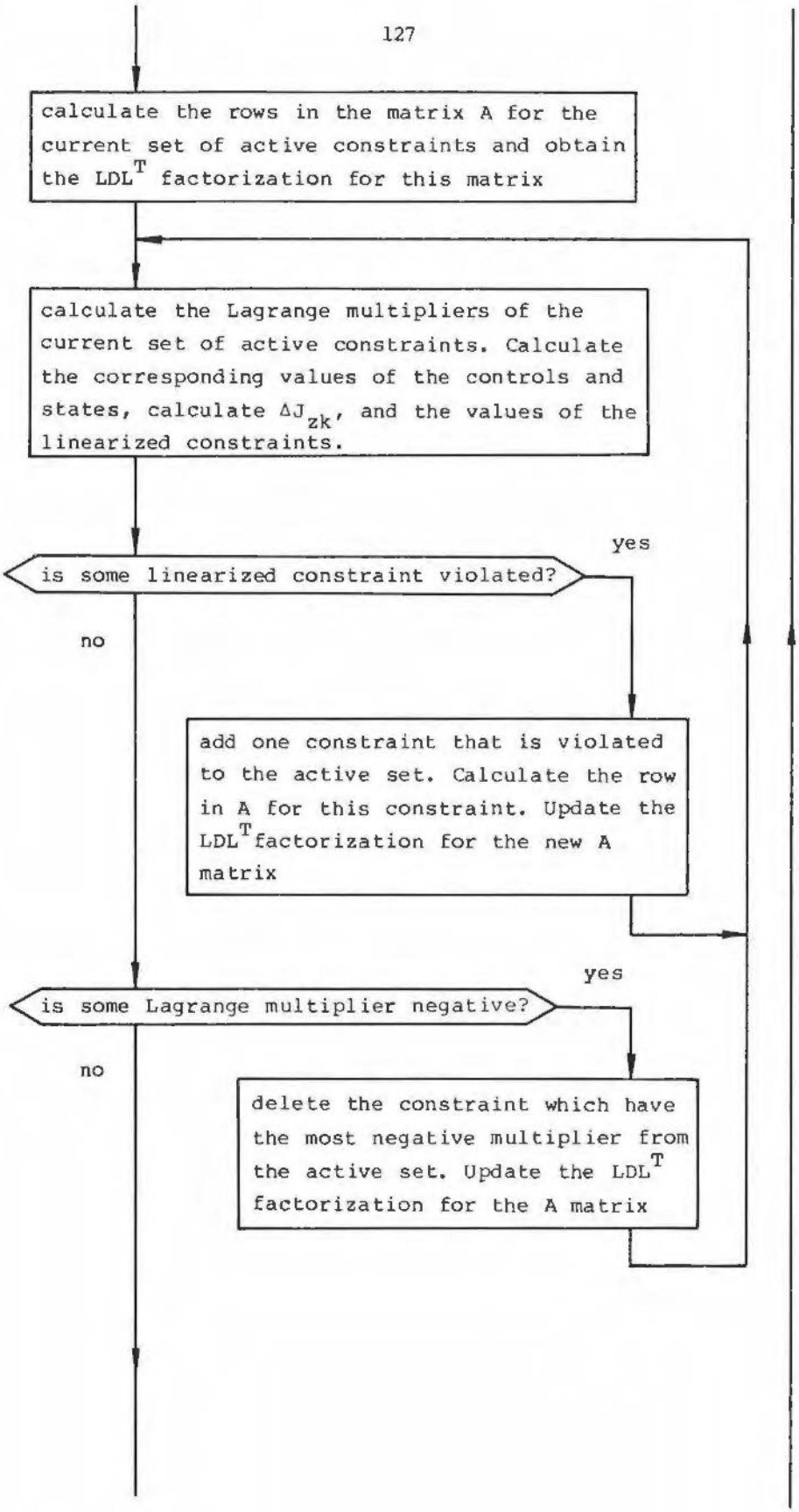
Two computer programs have been written for solving discrete time optimal control problems. The first program solves optimal control problems, that are directly formulated in discrete time such as (4.1)-(4.4). The other program treats problems that are

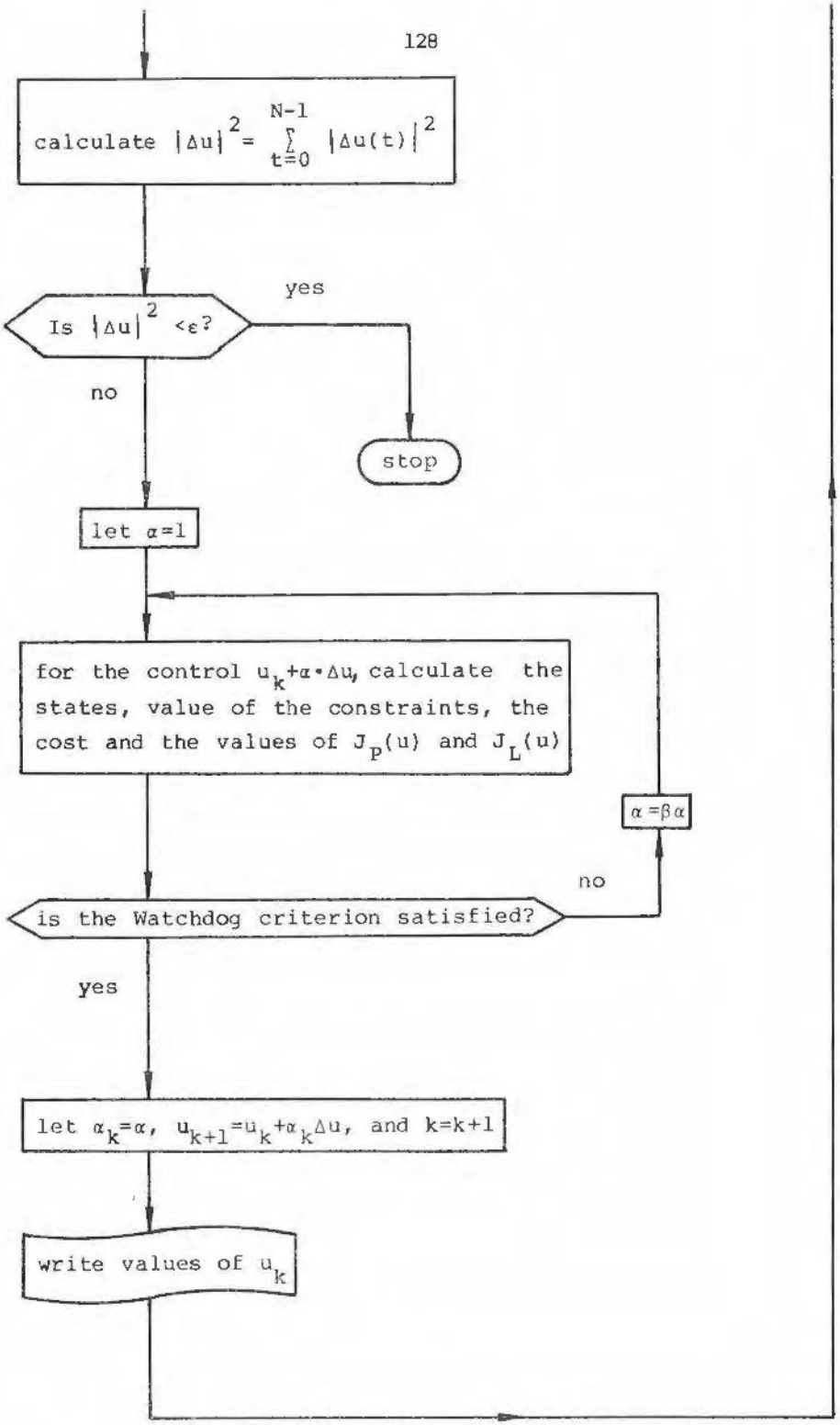
formulated in continuous time, while the control is constrained to be piecewise constant between sampling instants. This problem, along with relevant equations, is further discussed in the appendix.

The basic steps in these two programs are the same. In fact, they have the same main program - only certain subroutines differ between the two programs.

The organization of the calculations basically follow the summary of the algorithm, given in the Section 9.2. A flow chart of the main program has the following structure:


Initialize the problem
write the characteristics of the problem
write the initial control sequence
compute the states x and the cost for the current control sequence, and the values of the constraints
write the states x and cost for the current controls u_k
write the values of the constraints and their Lagrange multipliers
solve the Riccati equation and the other backwards equation
compute the solution to the unconstrained problem


calculate the rows in the matrix A for the current set of active constraints and obtain the LDL^T factorization for this matrix
calculate the Lagrange multipliers of the current set of active constraints. Calculate the corresponding values of the controls and states, calculate ΔJ_zk, and the values of the linearized constraints.
is some linearized constraint violated?
yes
no
add one constraint that is violated to the active set. Calculate the row in A for this constraint. Update the LDL^T factorization for the new A matrix
is some Lagrange multiplier negative?
yes
no
delete the constraint which have the most negative multiplier from the active set. Update the LDL^T factorization for the A matrix


calculate $|\Delta u|^2 = \sum_{t=0}^{N-1} |\Delta u(t)|^2$
Is $|\Delta u|^2 < \varepsilon$?
yes
no
stop
let $\alpha=1$
for the control $u_k+\alpha\cdot\Delta u$, calculate the states, value of the constraints, the cost and the values of $J_p(u)$ and $J_L(u)$
$\alpha=\beta\alpha$
no
is the Watchdog criterion satisfied?
yes
let $\alpha_k=\alpha$, $u_{k+1}=u_k+\alpha_k\Delta u$, and $k=k+1$
write values of u_k

The main program that administers this flow-chart consists of 180 FORTRAN statements, which corresponds to about 2100 bytes when implemented on a DEC 20 computer. All calculations, indicated in the flow-chart are performed in subroutines. These subroutines consist of 1300 FORTRAN statements, corresponding to approximately 9900 bytes. The differential equations in the Appendix are solved by a fourth order Runge-Kutta method (p. 346 in Dahlquist, Björck (1974)).

The program is written to handle both equality and inequality constraints of the type (4.4). In the quadratic subproblems, the active constraints are chosen in the following way. Steps 0-2 make it possible to start outside the region defined by the constraints.

0. Let all equality constraints and all inequality constraints which have a positive Lagrange multiplier be in the active set. Put ISAT=0.

1. Compute the solution for the current set of active constraints. If all constraints are satisfied for this solution put ISAT=1.

2. If ISAT=1 go to step 3. Else add the most violated constraint to the active set and go back to step 1.

3. If all constraints are satisfied then go to step 4, else add a constraint to the active set using the method described in section 3.2

4. If all Lagrange multipliers of the inequality constraints in the active set are positive then go to step 5. If not delete the inequality constraint with the most negative Lagrange multipler from the active set and go back to step 1.

5. The solution has been found.

The idea is to add violated inequality constraints to the active set until all constraints are satisfied. Then we can use the algorithm in Section 3.2 (steps 3-5).

9.5 User-program interface.

The organization of the program is such that the user specifies his non-linear optimal control problem in a separate FORTRAN subroutine. This FORTRAN subroutine communicates with the main program through four named COMMON blocks. In these COMMON blocks the following information should be given:

n_x, the number of states
n_u, the number of controls
N, the number of sampling instants
p, the number of inequality constraints.
$\bar{\mu}$, the values of the penalty parameters in the Watchdog technique (7.3).

ε, Parameter to determine the termination criterion.

Moreover the values of the time instants, $t_1,\ldots, t_p$ at which the constraints are given must be specified, as well as lower and upper bounds on the control signals.

Also, for given x and u the state transition function f (see (4.1)) as well as its first and second order derivatives should be calculated. Similarly, for given x and u the criterion function ℓ along with its first and second derivatives with respect to x and u should also be specified as well as the constraint functions h^i together with their first and second order derivatives with resepct to x and u. Finally, the initial values for the control-signals and for the Lagrange multipliers must be specified.

A typical subroutine is depicted in Figure 9.1

```
            SUBROUTINE USER
C
C    SUBROUTINE FOR THE PROGRAM  QADAPP, WHICH SOLVES DISCRETE TIME
C    OPTIMAL CONTROL PROBLEMS.
C
C    IN THIS SUBROUTINE THE PROBLEM IS DEFINED.
C
      REAL F(6), X(74), U(2), FX(6,6), FU(6,2), FXX(6,6,6), FXU(6,6,2),
     *  FUU(6,2,2), LX(6), LU(2), LXX(6,6),
     *  LXU(6,2), LUU(2,2), FIX(6), FIXX(6,6), DU0(2,201)
C
      REAL UNOM(2,201), VNOM(201), XIN(6), XNOM(6,201),
     *    UL(2,201), UU(2,201)
C
      REAL GCON(80), GXCON(80,6), GUCON(80,2), GXXCON(80,6,6),
     *   GXUCON(80,6,2), GUUCON(80,2,2), LACON(80),
     *  LAMBDA(80)
C
      INTEGER ITEQ(80), IPEQ(80)
C
      COMMON /QAD1/  IGRAD, KONT, ITEMAX, LP, N, NACT, NEW,
     *   NPRINT, NSTEP, NEND,
     *  NU, NEQ, NIEQ, NAEQ, INDEL, IFAIL, ILOOP
C
      COMMON /QAD2/ EPSIL, ETA1, ETA2, IT, T, TSAMP, UNORM, VNEW,
     *  VOLD, PCIC
C
      COMMON /QAD3/ UNOM, VNOM, XIN, XNOM, DU0, ITEQ, IPEQ,
     *  GCON, LACON, LAMBDA, UL, UU
C
      COMMON /QAD4/ SF, SL, F,  U, FX, FU, FXX, FXU, FUU,
     *     LX,  LU, LXX, LXU, LUU, FIX, FIXX, GXCON, GUCON,
     *    GXXCON, GXUCON, GUUCON, X
C
C
C
```

Figure 9.1 Program head for the subroutine to be written by the user of the optimization program.

When the program is run, the main program gives, for each iteration, the current values of the control signals, the states, the values of the constraints and the Lagrange multipliers and also the current cost. This information can be given on the terminal screen, on a line-printer or stored on disc-area.

For further information about how to use the program, consult the program manual, Jonson (1983).

9.6 A computer program for solving static nonlinear constrained programming problems.

In chapter 2 we described a general static non-linear constrained programming problem (2.11), which has the form

$$\text{minimize} \quad \ell(z) \tag{9.21a}$$

$$\text{subject to } g(z)=0 \tag{9.21b}$$

$$h(z)\leq 0 \tag{9.21c}$$

In that chapter we also showed that given the variables z_k, λ_{k-1}, μ_{k-1}, the Newton direction towards the Kuhn-Tucker-point of (9.21) is obtained by solving the following problem with linear constraints:

$$\underset{d}{\text{minimize}} \quad \ell_z(z_k)d+\frac{1}{2}d^T L_{zz}(z_k,\lambda_{k-1},\mu_{k-1})d \tag{9.22a}$$

$$\text{subject to } g(z_k)+g_z(z_k)d=0 \tag{9.22b}$$

$$h(z_k)+h_z(z_k)d\leq 0 \tag{9.22c}$$

where $L(z,\lambda,\mu)$ is the Lagrangian to the problem (9.13). The solution to (9.22) defines the variables d_k, λ_k and μ_k. The next value z_{k+1} is then determined from

$$z_{k+1}=z_k+\alpha_k d_k \tag{9.23}$$

where α_k is chosen according to the Watchdog technique described in section 7.4.

A computer program has been developed for the problem (9.21) using the iterations (9.23). Similarly to the program for optimal control problems, a subroutine has to be written by the user. This subroutine should specify the following values:

n, the number of variables
m_1, the number of equality constraints
m_2, the number of inequality constraints.

Also for any given value of z, the values of the functions $\ell(z)$, $g(z)$, and $h(z)$ (defined in (9.21)) along with their first and second order derivatives with respect to z have to be given.

When the program is run the values of z_k, $g(z_k)$, $h(z_k)$, λ_k and μ_k are printed on a file for each iteration.

In the program the quadratic sub-problems (9.22) are solved using the technique described in section 3.3. The NAG-routines FO1BRF and FO4AXF are used for factorizing the system matrix in (3.6) and (3.7) and for solving this system of linear equations.

10. AN OPTIMAL POWER DISTRIBUTION PROBLEM

10.1 Introduction

In this chapter we consider an electrical network of the same kind as that of example 2.1. It will however be of more realistic complexity. We assume that the network has a number of nodes. Some of the nodes are connected by electrical lines. Not all nodes need to have direct lines between each other, but all nodes are connected. At each node there are power generators, power consumers or both. A node can have more than one generator. Each generator is allowed to produce power within a given interval only. The consumers require a known amount of electric power. On some of the lines there may also be transformers. To each generator there is assigned a cost function, whose value is determined by the amount of power generated.

The problem is now to adjust the generated power so that the total cost, summed over each power generator, is minimized when the power demand from the consumers is satisfied.

This problem is of the type (2.11). More specifically, we may define the problem as follows:

Let the vector z consist of all values that describe the network with the following quantitites:

V_i the voltage amplitude at each node

φ_i the phase angle of each node (except the slack-node).

t_j the amplitude ratio for transformer j

P_{Gk} the real power produced by generator k

$Q_{G\ell}$ the complex power produced by generator ℓ.

Objective function

We assume that only the real power generators have associated costs. Hence our cost function is

$$\ell(z) = \sum_k C_k(P_{Gk}) \qquad (10.1)$$

The index k on C_k implies that the cost functions may be different for the different generators.

Equality constraints

Each node must be in balance, both with respect to real and to complex power. This gives us the constraints for each node. As in example 2.1 we get that for node 1 these constraints are (see (2.2))

$$0 = P_{G1} - P_{D1} - V_1^2 \cdot (Y_1 \cos\varphi_1 + Y_{12}\cos\delta_{12}) + V_1 \cdot V_2 \cdot \cos(\varphi_1 - \varphi_2 - \delta_{12}) \qquad (10.2a)$$

$$0 = Q_{G1} - Q_{D1} + V_1^2 (Y_1 \sin\varphi_1 + Y_{12}\sin\delta_{12}) + V_1 \cdot V_2 \cdot Y_{12} \cdot \sin(\varphi_1 - \varphi_2 - \delta_{12}) \qquad (10.2b)$$

Inequality constraints

In order to operate safely, the lines must not be overloaded. That is to each line there is an upper limit on the heat produced in the line. (cf equation (2.7)).

Also, the network must be stable. Therefore the phase angle between two nodes at each end of a line must not differ by more than $\frac{\pi}{2}$. For one of the nodes the phase angle is fixed to zero. This node is usually called the slack node.

Simple constraints

The generators have limited capacity. Therefore for each generator, the generated power must be within a given interval.

$$P_{\ell k} \leq P_{Gk} \leq P_{uk} \tag{10.3a}$$

$$Q_{\ell \ell} \leq Q_{G\ell} \leq Q_{u\ell} \tag{10.3b}$$

10.2 The test example

We choose an example given in Arriatti (1971), which is often used as a test example for algorithms designed to solve elelctrical power distribution problems. It consists of 10 nodes and 13 transmission lines. In 5 of these lines there is a transformer. The total number of power generators is 14, allocated at 7 different nodes. At each of these 7 nodes there is also a reactive generator. This gives a total of 45 free variables, 20 equality constraints, 20 inequality constraints and 41 simple constraints. The varibles are: 10 voltage amplitudes, 2 phase angles, 5 transformers, 7 reactive power generators and 14 real power generators. The 22 inequality constraints consist of 13 from maximum capacity at the 13 lines and 8 from phase angle differences. The 41 simple constraints are: 10 for the voltage amplitudes, 6 phase angles directly connected to the slack node, 5 transformers, 14 real power generators and 7 reactive generators. The geometry of the network is given by figure 10.1.

This problem is an example of a so-called sparse problem. Sparsity here means that the Hessian of the objective function and the constraints, and the Jacobian of the constraints contain many zeros.

The figures 10.2a and 10.2b show non-zero elements in the Hessian of the Lagrangian of the problem (equations (10.1) and (10.2)) and in the Jacobian of the equality constraints (equation (10.2ab)) and inequality constraints (equation (2.7)). The variables are ordered in the following way:

10	voltage amplitudes
5	transformers
9	voltage phase angles (node 4 is slack node)
7	reactive power generators
14	real power generators

The voltage levels have the simple constraints

$$205 \leq V_i \leq 240 \tag{10.4}$$

where the unit is kV.

The ratio of the transformers have the limits

$$0.9 \leq t_i \leq 1.1 \tag{10.5}$$

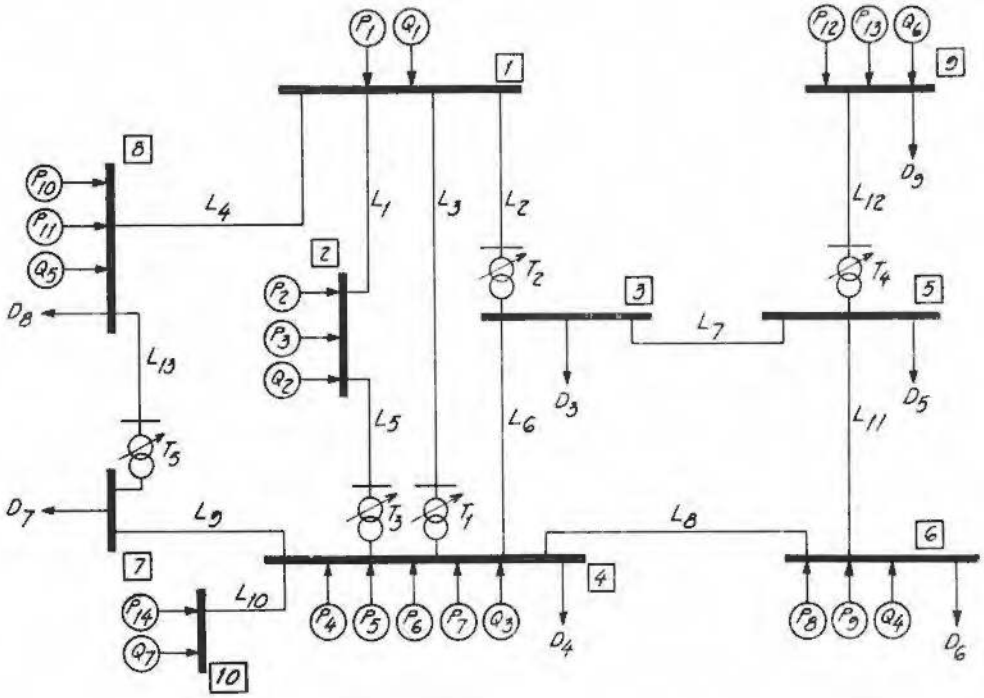


Figure 10.1 The structure of the network in the test example. Numbers in squares denote node numbers. P means real power generators. Q means reactive power generators. L denotes the lines, T the transformers and D is the demand.

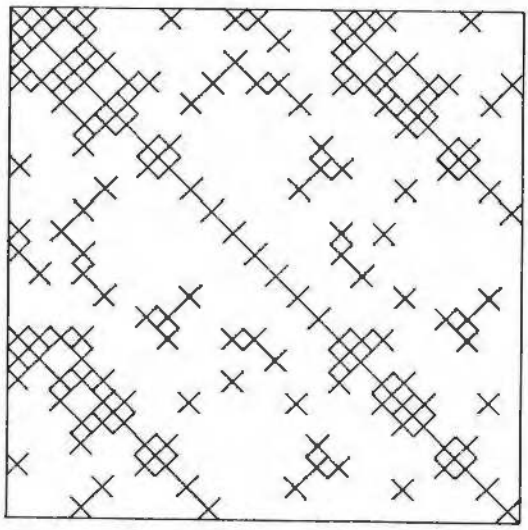


Figure 10.2a The non-zero elements in the Hessian of the problem. Only the left upper 32x32 part of the matrix is shown. The rest of the matrix is zero except for the diagonal.

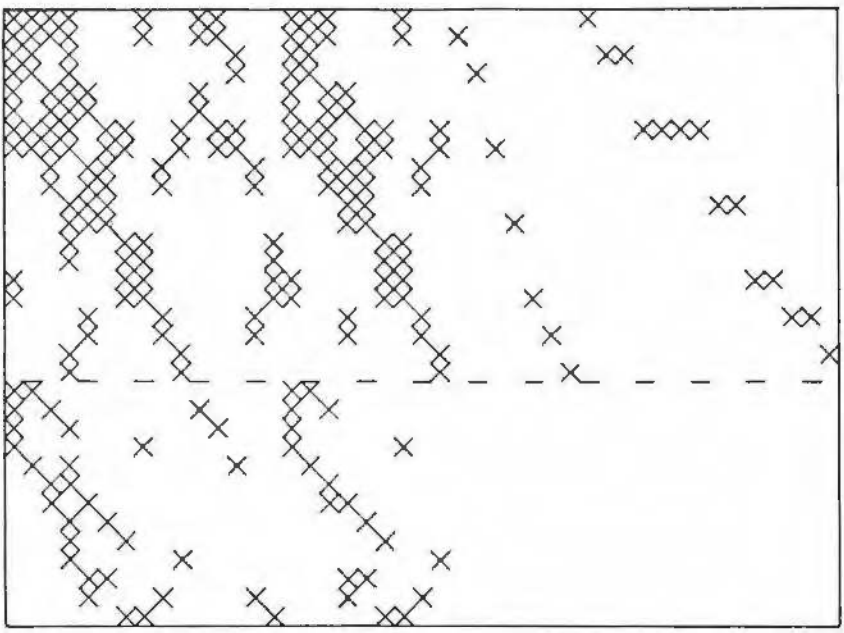


Figure 10.2b The non-zero elements in the Jacobian of the matrix. The upper part corresponds to the 20 equality constraints and the lower part, to the 13 inequality constraints.

Generator number	Lower bound (MVAr)	Upper bound (MVAr)	Initial value (MVAr)	Value at the solution (MVAr)
1	-24	120	0.0	120.00
2	-48	240	0.0	5.72
3	-78	390	0.0	390.00
4	-30	150	0.0	150.00
5	-89	195	0.0	160.46
6	-23	115	0.0	104.44
7	-24	120	0.0	28.44

Table 10.1 Limits of generated reactive power.

Generator number	Cost coefficients A_1	A_2	Limits lower (MW)	upper (MW)	Initial value (MW)	Value at the solution (MW)
1	1.0	0.000461	80	217	150.0	121.40
2	1.0	0.000461	80	217	150.0	148.19
3	1.0	0.000461	80	217	150.0	148.19
4	1.0	0.000461	80	217	150.0	217.00
5	1.0	0.000461	80	217	150.0	217.00
6	1.0	0.000926	40	108	80.0	108.00
7	1.0	0.000926	40	108	80.0	108.00
8	1.0	0.000926	40	108	80.0	108.00
9	1.0	0.000926	40	108	80.0	108.00
10	1.0	0.000461	80	217	150.0	193.13
11	1.0	0.000926	40	108	80.0	95.53
12	1.0	0.000139	30	72	50.0	72.00
13	1.0	0.000926	40	108	80.0	108.00
14	1.0	0.000461	80	217	150.0	217.00

Table 10.2 Cost coefficients and limits for the real power generators.

Line number	Reactance (ohms)	Shunt admittance (mhos)	Maximum load (MW)	Load at the solution (MW)
1	5.00+j 24.50	j 0.0002	25.0	6.7
2	5.75+j 28.00	j 0.0002	14.0	14.0
3	6.00+j 39.50	j 0.0003	25.0	11.0
4	22.80+j 62.60	j 0.0002	25.0	0.9
5	5.00+j 24.50	j 0.0001	14.0	14.0
6	24.70+j 97.00	j 0.0002	25.0	5.9
7	24.70+j 97.00	j 0.0002	25.0	3.0
8	3.75+j 24.75	j 0.0001	25.0	0.5
9	8.25+j 33.00	j 0.0003	25.0	1.1
10	2.00+j 10.00	j 0.0002	25.0	8.6
11	9.50+j 31.80	j 0.0002	25.0	13.8
12	6.00+j 39.50	j 0.0003	25.0	10.3
13	8.25+j 32.30	j 0.0003	25.0	5.2

Table 10.3 Reactances and shunt admittances of the lines.

Node number	Power demand (MW)	(MVAr)
3	250	150
4	1000	630
5	150	75
6	100	35
7	100	50
8	250	150
9	100	30

Table 10.4 Power demand at different nodes.

The upper and lower limits of the complex generators are given in Table 10.1. In that table the initial values given to the algorithm are also shown, and also their values at the solution.

The cost functions for the real power generators are of the form

$$C_K(P_{GK})=A_1 P_{GK}+A_2 P_{GK}^2 \qquad (10.5)$$

where the values of A_1 and A_2 for the different generators are given in Table 10.2. That table also contains the lower and upper limits on the generated real power for the different real power generators and also the initial values and the optimal values obtained for the real powers.

The transmission lines are represented by their equivalent π. Hence each line has a length reactance and a shunt admittance at its ends. The values of these reactances and shunt admittances are given in Table 10.3. The power demand at the nodes is given by Table 10.4.

Initially the voltage amplitudes were set to 225.0kV and the voltage phase angles were set to zero. The ratio of the transformers were set to one. The initial values gave that the absolute value of the constraint vector was 955.5 MW.

When we tried to solve this problem, the situation described in Figure 6.2 occured. Therefore all equality constraints were scaled during the first iterations according to the idea of Powell (1977). Then the program managed to find a local minimum point which took 10.5 CPU-seconds and 14 iterations. The absolute value of the equality constraints was reduced to 0.11 (MW) and the value of the objective function was 2148.5 units. The values of the variables at the solution can be found in the tables 10.1, 10.2 and 10.5.

	Voltage		Transformer
	amplitude	phase angle	ratio
1	240.0	0.019	1.089
2	240.0	0.072	0.927
3	217.2	-0.064	0.982
4	236.7	———	1.003
5	222.5	-0.055	0.933
6	240.0	-0.003	
7	240.0	-0.022	
8	234.5	0.012	
9	240.0	-0.007	
10	240.0	0.037	

Table 10.5 The values of voltages and transformers at the solution.

Usually this type of problems are solved by Reduced Gradient Methods (p. 262 in Luenberger (1973)). However, these methods cannot handle the inequality constraints (2.7) effectively. Therefore also Reduced Gradient Methods and Quasi Newton versions of Wilson's method have been tried. (Talukdar and Giras (1982)).

In this method, m_1 of the variables are eliminated using the m_1 equality constraints. The remaining, inequality constrained optimization problem, is solved by Wilson's method, when the Hessian of the Lagrangian is updated, using Quasi-Newton techniques (Powell (1977), Han (1975)).

11. NUMERICAL EXAMPLES OF OPTIMAL CONTROL PROBLEMS

In this chapter we shall apply the computer program, described in Section 9.3 to some optimal control problems. These will include both simple test cases (Example 11.1), "tricky" small problems, (Example 11.2, 11.3 and 11.4) and a more realistically formulated application example (Example 11.5).

It would of course be interesting to compare our method with other methods designed for solving discrete time optimal control problems, but this is not easy, because very few other algorithms have resulted in computer programs. Therefore, as a comparison we also solve some of the corresponding problems in continuous time, with the continuous time DDP-methods by Jacobson and Mayne (1970). The constraining hyperplane technique for this problem is described in Mårtensson (1972). Note however that these algorithms solve completely different types of problems. We also claim that our formulation of the problem is a more realistic one for digital control and that the DDP-mehod gives a solution that is more difficult to implement on a digital computer.

Note, however, that for our method the state constraints might be violated between the sample instants.

Example 11.1 The discrete time double integrator

We consider a system with two states and one control variable. The dynamics of the system is assumed to be

$$x(t+1)=Fx(t)Gu(t) \tag{11.1}$$

where the matrices F and G are

$$F=\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix} \; ; \; G=\begin{pmatrix}1/2\\ 1\end{pmatrix} \tag{11.2}$$

This system is the result of sampling the differential equation $\ddot{y}=u$, where $y=x_1$, with the sampling interval equal to one time unit.

Consider the following performance index for the system

$$J(u)=\frac{1}{2}\sum_{t=0}^{N-1} u^2(t) \tag{11.3}$$

with the number of steps N equal to 10. Assume that the initial value of the state is

$$x(0)=\begin{pmatrix}0\\10\end{pmatrix} \tag{11.4}$$

and that we have the following state constraints:

$$x(N)=\begin{pmatrix}0\\-10\end{pmatrix} \tag{11.5a}$$

and

$$x_1(t)\leq 10, t=0,\ldots,N \tag{11.5b}$$

At iteration 0 the initial values of the controls were set to $u(t)=-2, t=0,\ldots,9$, which is the solution to the problem if the constraint (11.5b) is removed. The initial values of the Lagrange multipliers were chosen to be $\mu_{-1}=0$. Since this is a constrained linear quadratic (CLQ) problem, the algorithm will find the solution to this problem after one iteration only. The only problem is to choose the correct set of active constraints. How this choice is made is illustrated in Figure 11.1ab and Figure 11.2ab. First the solution was obtained, when only the constraint (10.5a) was included. This solution makes the constraint (10.5b) for t=5 to be the most violated one. This constraint is included in the active set and a new solution is calculated. For this solution the constraint (11.5b) for t=3 is the most violated one. This constraint is also included in the active set and a new solution is obtained, which leads to inclusion of the constraint at t=7 into the active set. For the next solution, all constraints are satisfied but the multiplier corresponding to the constraints at t=5, is negative, so we delete that constraint from the active set and obtain a new solution. This solution satisfies all Kuhn-Tucker conditions, and is accepted as the solution to the CLQ-problem. For these calculations, 1.09 CPU-seconds were required, on the DEC-20 computer.

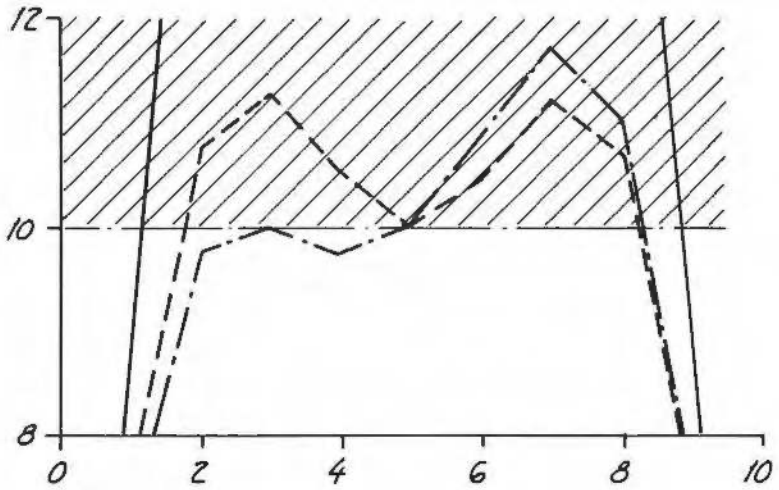


Figure 11.1a The state $x_1(t)$,t=0,...,10. The area outside the constraint is shaded. The solid line corresponds to the initial control u(t)=-2. The solution when $x_1(5)$=10 (1:st iteration) is given by the dashed line and the solution when $x_1(3)$=10 and $x_1(5)$=10 (2:nd iteration) is given by the chain-dotted line.

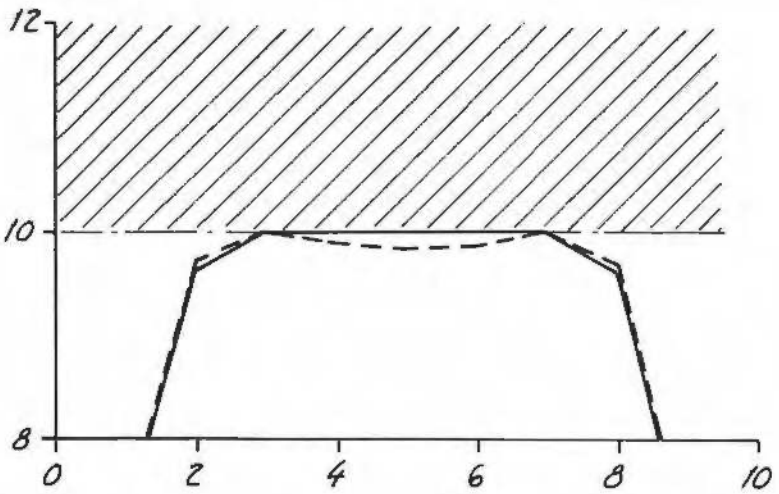


Figure 11.1b The state $x_1(t)$,t=0,...,10. The area outside the constraint is shaded. The solid line is the solution when $x_1(3)$=10,$x_1(5)$=10 and $x_1(7)$=10 (3:rd iteration) and the dashed line corresponds to the optimal control of the problem (4:th iteration).

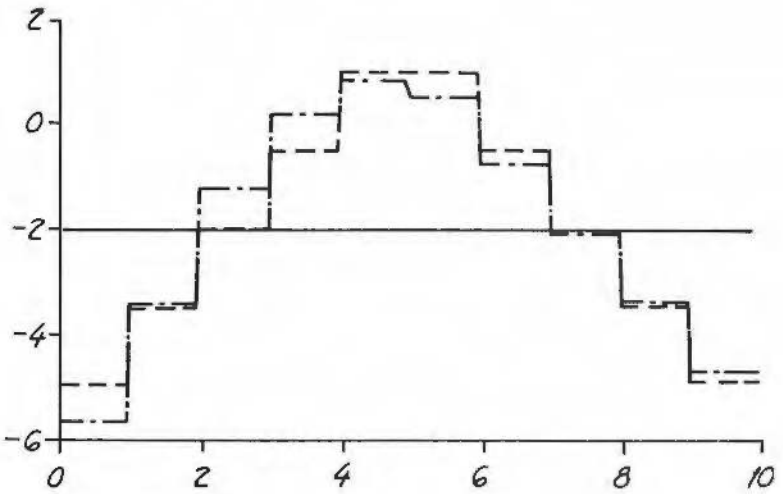


Figure 11.2a The control signal u(t),t=0,...,9. The solid line is the initial control u(t)=-2. The dashed line is the control when $x_1(5)$ =10 and the chain-dotted line is the control when $x_1(3)$=10 and $x_1(5)$=10.

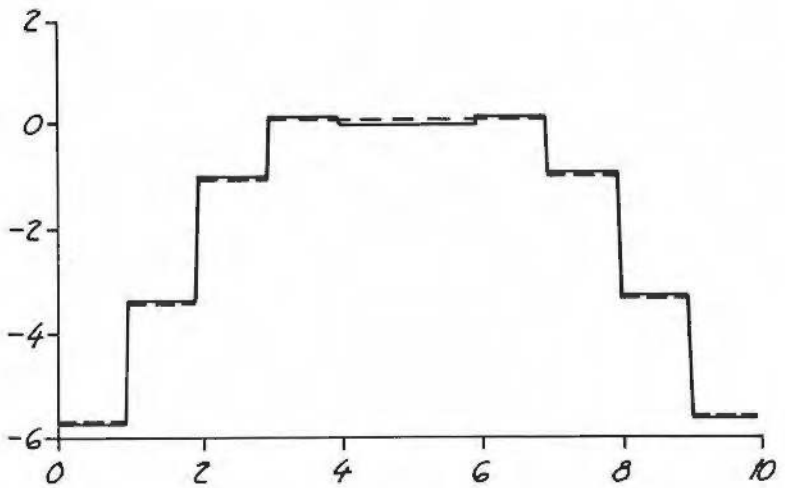


Figure 11.2b The control signal u(t),t=0,...,9. The solid line is the solution when we have 3 constraints in the active set namely $X_1(3)$=10,$X_1(5)$=10 and $X_1(7)$=10. The dashed line is the optimal control of the problem.

Example 11.2 A problem where ℓ_{xx} is not positive definite.

We consider the following system with one state variable and one control variable:

$$x(t+1)=2\cdot x(t)+x^2(t)+u(t) \tag{11.6}$$

The objective function is

$$J(u)=x(N)+\sum_{t=0}^{N-1}\left(\frac{1}{4}\,u^2(t)-u(t)-x(t)-\frac{1}{2}\,x^2(t)\right) \tag{11.7}$$

and the initial value

$$x(0)=0. \tag{11.8}$$

No constraints are given for this problem. The value of N was chosen as 10. The optimal solution to this problem is $u(t)=0,t=0,\ldots,N-1$, but because $\ell_{xx}=-1$ in (11.7), this is a very difficult problem to solve. We started with $u(t)=0.01,t=0,\ldots,9$ which gave the value $J(u)=8.1\cdot 10^3$ for the cost function and $x(10)=16.3\cdot 10^3$. The value of ε in the stopping-criterion was chosen as $1.0\cdot 10^{-12}$.

We will now show the rate of convergence for our method. Let e_k be the value

$$e_k=\left|u_k-u^*\right| \tag{11.9}$$

and assume that close to the solution we have

$$e_{k+1}=K\cdot e_k^p \tag{11.10}$$

where we want to estimate p. Let φ_k be the logarithm of e_k. From (11.10) we then get

$$\hat{p}_k=\frac{\varphi_k-\varphi_{k-1}}{\varphi_{k-1}-\varphi_{k-2}} \tag{11.11}$$

where $\hat{p}_k$ is the estimate of p at iteration k.

The computer needed 23 iterations to satisfy the stopping criterion, and then 2.83 CPU-seconds was used. In Figure 11.3 the value of $\hat{p}_k$ defined by (11.11) is plotted versus iteration number.

The figure shows that $\hat{p}_k$ tends to 2 as k tends to infinity. The abrupt jump from 2 to 1 from iteration 22 to iteration 23 is due to underflow when calculating e_k when k=23.

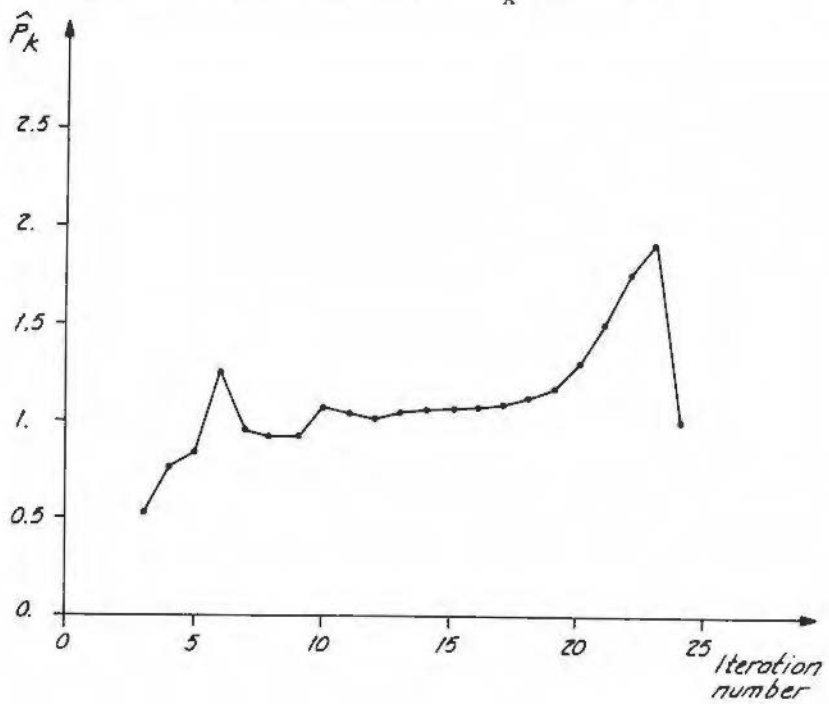


Figure 11.3 The estimate of the rate of convergence $\hat{p}_k$, when the Newton method is used.

Example 11.3 Adapted from Mårtensson (1972).

Consider the continuous time system with two states and one control signal

$$\dot{x}(\tau)=Ax(\tau)+Bu(\tau), \tag{11.12}$$

where the matrices A and B are

$$A=\begin{pmatrix}0 & 1\\ 0 & 0\end{pmatrix};\ B=\begin{pmatrix}0\\ 1\end{pmatrix} \tag{11.13}$$

The criterion function is

$$J=\frac{1}{2}\int_0^1 u^2(\tau)d\tau \tag{11.14}$$

and the initial condition is

$$x(0)=\begin{pmatrix}0\\ 0\end{pmatrix} \tag{11.15}$$

We have the terminal constraints

$$x(1)=\begin{pmatrix}1\\ 0\end{pmatrix} \tag{11.16}$$

and the state inequality constraints

$$x_1(\tau)-8(\tau-0.5)^2+0.5\leq 0 \tag{11.17}$$

We solve this problem using the program based on the algorithm described in the Appendix. The sampling time was chosen as $T_s=0.05$ which makes N=20. The constraint (11.14) is then approximated by

$$x_1(s)-8(s-0.5)^2+0.5\leq 0;\ s=T_s\cdot t;t=0,\ldots,20. \tag{11.18}$$

Since this is actually a quadratic problem with linear constraints our algorithm solves this problem in one iteration. The

initial sequence to the algorithm was u(t)=0; t=0,...,19, which gave a non feasible solution.

In the subproblem, 8 iterations were needed to obtain the correct set of active constraints. The computer needed 2.0 CPU-seconds for the calculations. At the optimum, three of the inequality constraints (11.19) are active, namely for s=0.55, s=0.80 and s=0.85.

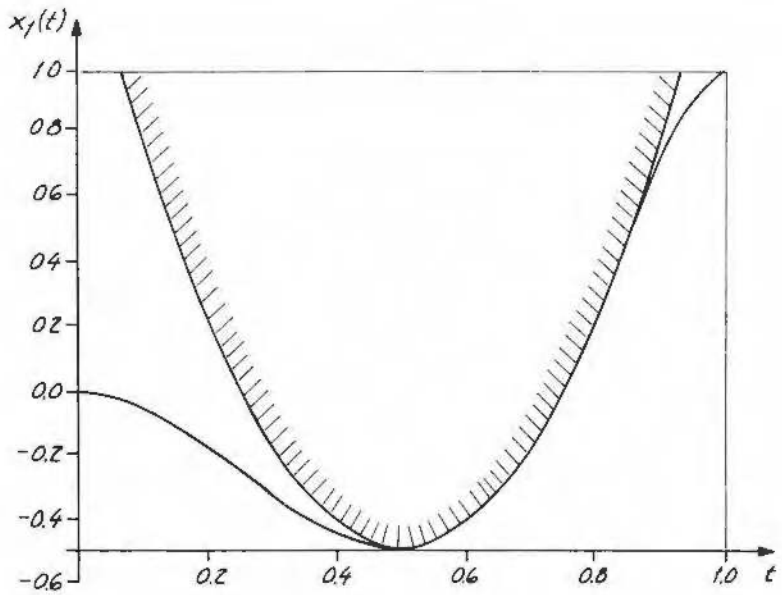


Figure 11.4 The state $x_1(s)$,s=0.05t,t=0,...,20 at the solution. The state touches the infeasible region for s=0.55, s=0.80 and s=0.85.

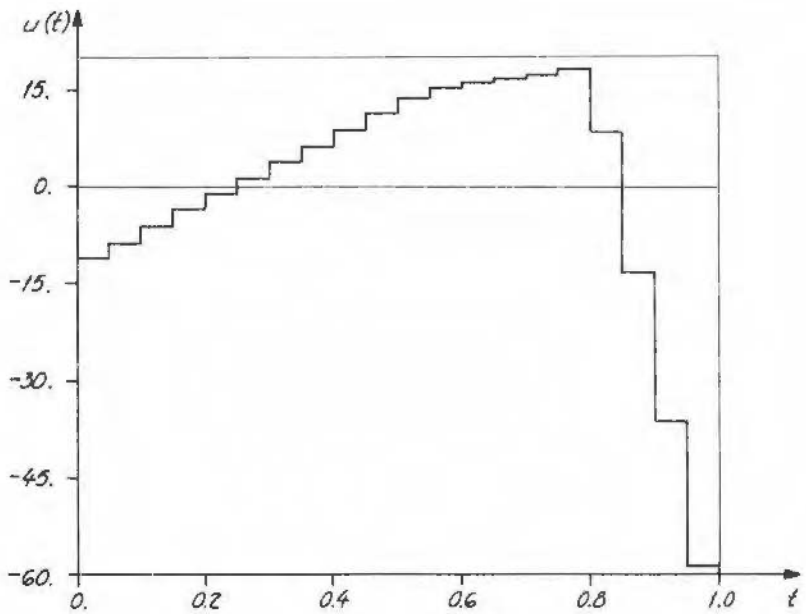


Figure 11.5 The optimal control for the problem in example 11.3

To compare with DDP, we also solved this problem using T_s=0.01 which makes N=100. For this problem our algorithm required 5.3 CPU-seconds whereas the DDP-method needed 8.5 CPU-seconds.

Example 11.4 The constrained Van der Pol equation.

Consider the system

$$\dot{x}_1 = x_2 \tag{11.19a}$$

$$\dot{x}_2 = (1-x_1^3)x_2 - x_1 + u \tag{11.19b}$$

with the initial value

$$x(0) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{11.20}$$

and the criterion function

$$J(u) = \frac{1}{2} \int_0^1 (x_1^2 + x_2^2 + u^2) d\tau \tag{11.21}$$

The constraints are

$$x(1) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{11.22}$$

and

$$-3 \leq u(t) \leq 3 \tag{11.23}$$

This problem was solved with our sampling program where the sampling time was chosen as 0.01. Hence N=100. The constraints on the controls are of the type discussed in section 8.2. Therefore we solve this problem using the algorithm proposed in section 8.3. The initial controls were $u(t)=0$ and $\mu_{-1}=0$. We also solved the continuous version of this problem with the continuous DDP-method. The result is summarized in the following table:

	Our program	DDP
Number of iterations	4	6
Value of criterion function	2.934	2.938
Value of terminal constraints	0.0006 0.0008	0.0008 0.0009
Solution time (CPU-seconds)	9.27	6.74

Example 11.5 A container crane.

This example is based on a problem described in Mårtensson (1973).

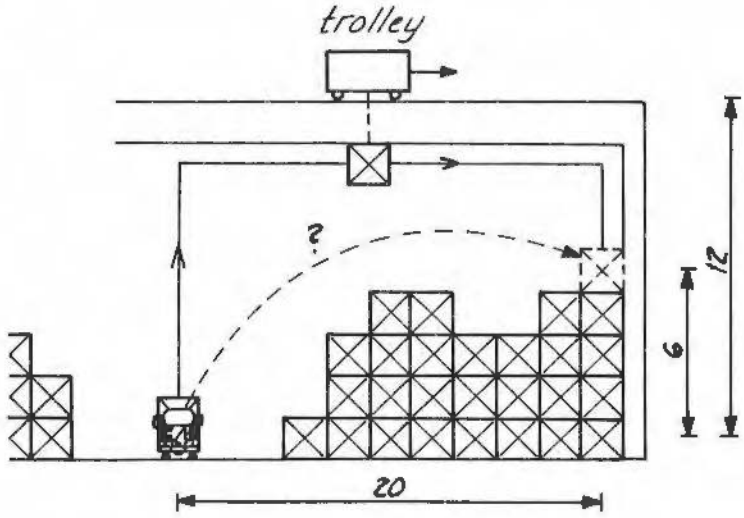


Figure 11.6

A truck delivers containers to a storage house. A crane in this storage house is supposed to lift the container from the truck and stack it in a predetermined position. During the transportation, the crane may have to lift the container over other, already stacked, containers. We assume that we can control the accelerations of the trolley and the winch separately.

We use the following notation.

$(\xi_1, 0)$	position of the trolley
(ξ_2, η_2)	position of the load
θ	angular deviation
T_c	tension in the cable
L_c	length of the cable
m	mass of load
g	gravity acceleration ($9.81 m/s^2$)

See also Figure 11.7. From the figure we find that

$$\xi_2=\xi_1+L_c\cdot\sin\theta \tag{11.24a}$$

$$\eta_2=-L_c\cos\theta \tag{11.24b}$$

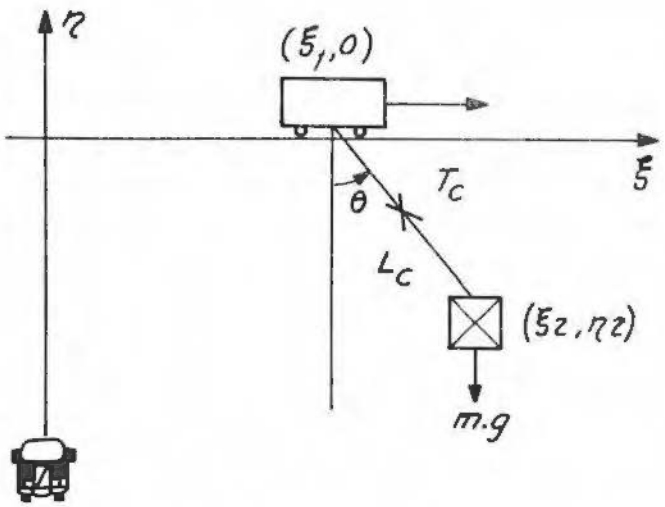


Figure 11.7 Definition of the variables for the container crane.

If we let u(t) be the acceleration of the trolley and $u_2(t)$ be the control signal to the winch we get the following differential equations

$$\ddot{\xi}_1=u_1 \tag{11.25a}$$

$$\ddot{L}_c=u_2 \tag{11.25b}$$

For the load we get

$$m\ddot{\xi}_2=-T_c\cdot\sin\theta \tag{11.26a}$$

$$m\ddot{\eta}_2=T_c\cos\theta-mg \tag{11.26b}$$

If we multiply (11.26b) by sinθ and then eliminate T_c using (11.26a) we get

$$\ddot{\xi}_2\cos\theta+\ddot{\eta}_2\sin\theta=-g\ \sin\theta \tag{11.27}$$

By differentiation of (11.24) twice with respect to τ we get:

$$\ddot{\xi}_2=\ddot{\xi}_1+(\ddot{L}_c-\dot{\theta}^2)\sin\theta+2\dot{L}_c\dot{\theta}\cos\theta+L_c\ddot{\theta}\ \cos\theta \tag{11.28a}$$

$$\ddot{\eta}_2\ =-(\ddot{L}_c-\dot{\theta}^2)\cos\theta+2\dot{L}_c\dot{\theta}\sin\theta+L_c\ddot{\theta}\ \sin\theta \tag{11.28b}$$

If we use (11.26) to eliminate $\ddot{\xi}_2$ and $\ddot{\eta}_2$ in (11.28) we get

$$\ddot{\theta}\ =-\ \frac{1}{L_c}\ (g\ \sin\theta+2\dot{L}_c\ \dot{\theta}+\xi_1\cos\theta) \tag{11.29}$$

We now define the state variables as

$$x_1=\xi_1 \tag{11.30a}$$

$$x_2=\dot{\xi}_1 \tag{11.30b}$$

$$x_3=\theta \tag{11.30c}$$

$$x_4=\dot{\theta} \tag{11.30d}$$

$$x_5=L_c \tag{11.30e}$$

$$x_6=\dot{L}_c \tag{11.30f}$$

The dynamics of the system is then described by the following system of differential equations.

$$\dot{x}_1 = x_2 \tag{11.31a}$$

$$\dot{x}_2 = u_1 \tag{11.31b}$$

$$\dot{x}_3 = x_4 \tag{11.31c}$$

$$\dot{x}_4 = -(g \sin x_3 + u_1 \cos x_3 + 2x_4 x_6)/x_5 \tag{11.31d}$$

$$\dot{x}_5 = x_6 \tag{11.31e}$$

$$\dot{x}_6 = x_2 \tag{11.31f}$$

We assume that the system is initiated at

$$x(0) = (0,0,0,0,12,0)^T \tag{11.32a}$$

and after 20 seconds the system should be in the state

$$x(20) = (20,0,0,0,6,0)^T \tag{11.32b}$$

We assume that we can approximate the already stored containers by the parabola

$$\eta = -4 - 0.25(\xi - 0.25)^2 \tag{11.33}$$

The load must therefore be above this parabola. This gives us the state inequality constraint

$$x_5 \cos x_3 - 4 - 0.25(x_1 + x_5 \sin x_3 - 12.5)^2 \leq 0 \tag{11.34}$$

When we solved this problem with our algorithm, the sampling time was chosen as 0.2 seconds which gives the value N=100.

The initial sequence was chosen as

$$u(\tau)=\begin{pmatrix} 0.2 \\ -0.04 \end{pmatrix} \; ; \; 0 \leq \tau < 10 \qquad (11.30a)$$

$$u(\tau)=\begin{pmatrix} -0.2 \\ 0.03 \end{pmatrix} \; ; \; 10 \leq \tau < 20 \qquad (11.30b)$$

which makes a trajectory that satisfies only the first two of the terminal constraints (11.32b).

The iterations were terminated when the value of $|\Delta u|$ was smaller than 0.1. The algorithm needed 3 iterations to find the solution and took approximately 55 CPU-seconds.

The position of the load is plotted versus time in Figures 11.8 both for the initial control seqence (Figure 11.8a) and the optimal control sequence (Figure 11.8b).

We also solved this problem in its continuous time counterpart using the DDP-method, Jacobson and Mayne (1970), in combination with a constraining hyperplane technique to handle the state constraints (11.29) as in Mårtensson (1973). Also in this case the time interval $[0,20]$ was divided into 100 steps. The DDP-program needed 14 iterations and 103 CPU-seconds to solve this problem. The value of the criterion function was in this case 0.6119 whereas our algorithm gave the value 0.6404. The most violated terminal constraint had the absolute value 0.0008 for the DDP program and 0.0004 for our program.

To solve this problem, it was necessary to use the regularization technique proposed in Section 6.2. We chose the diagonal elements so that the matrix C became positive definite, and so that the absolute values of the elements in β did not exceed 1.5. Also, for both our method and for the DDP-method it was necessary to add the square of the terminal constraints to make the programs operate in a suitable way. (This is a trick often employed when solving optimal control problems.)

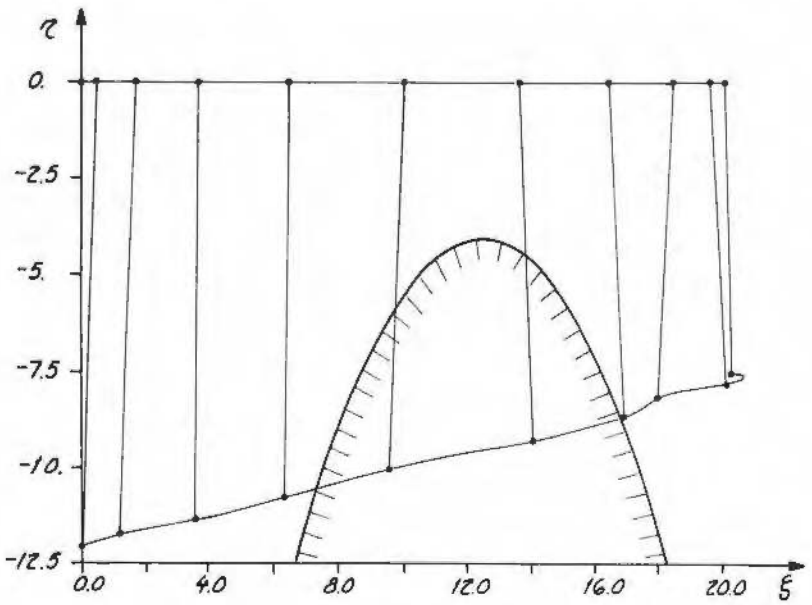


Figure 11.8a The position of load and trolley at every other second for the initial control sequence.

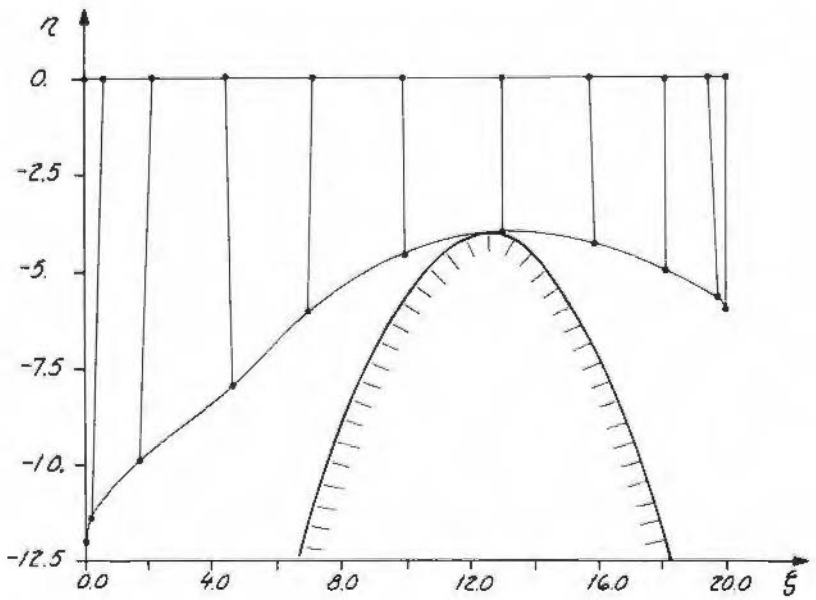


Figure 11.8b The position of load and trolley at every other second for the optimal control sequence.

Summary

We have shown that our program and the algorithm upon which it is based is capable of solving non-linear, discrete-time optimal control problems of different kinds, in particular problems with state constraints. The program has also been applied to other problems, see e.g. Trulsson (1983), and has exposed robust practical convergence properties.

Comparisons with other numerical schemes, such as DDP show that our method is competitive. It must however be remarked that it is difficult to evaluate comparisons between the continuous-time DDP-algorithm and our discrete-time algorithm. A fair comparison should be carried out against a discrete-time counterpart of the DDP-algorithm, which can handle state constraints. Such an algorithm has, however, apparently never been explicitly reported in the literature.

The numerical expressions also reveal that some care must be exercised in the choice of numerical algorithms in the subroutines, that mechanize the different equations that are to be solved in the course of the optimization scheme. Replacing the straightforward implementation of the Riccati-equation by more sophisticated and numerically sound versions is probably the most important possible improvement of this kind.

12 CONCLUSIONS

Many problems in engineering can be formulated as optimization problems. This thesis focuses on two particular types of optimization problems, namely nonlinear programming problems, i.e. nonlinear optimization problems with equality and inequality constraints, and optimal control problems in discrete time with state and control constraints. It has been the aim throughout the thesis to use a unified approach to these two problems.

To that end an algorithm has been developed which can be looked at from two different viewpoints. On the one hand it is exactly Wilson's method, i.e. a Newton method for constrained problems, when the optimal control problem is regarded as a static nonlinear optimization problem. On the other hand it is a method that takes advantage of the peculiar structure of optimal control problems and has a form similar to well known optimal control algorithms for problems without state and control constraints. A benefit from the approach is that it suggests a method of handling constraints, in particular pure state constraints, which are notoriously difficult to treat.

To get a deeper understanding of the relation between the two points of view, the regularity conditions of the nonlinear programming problem have been interpreted in the optimal control framework. It has been shown that the constraint qualification leads to a controllability condition and that a certain definiteness condition can be checked from the behaviour of a Riccati equation.

Our approach leads to a simple treatment of convergence questions. Due to the mathematical programming interpretation of our algorithm we have been able to establish both global convergence and a locally quadratic convergence rate.

A program has been developed which uses our algorithm to solve a completely general nonlinear discrete time optimal control problem. Both problems that are discrete-time problems in themselves and problems formulated as piecewise constant control

of continuous-time systems (usually arising from computer control) can be handled. The program has been used successfully to solve a number of problems of varying complexity. Our approach has thus been shown to have not only theoretical interest, but practical usefulness as well.

APPENDIX

SAMPLING OF CONTINUOUS TIME SYSTEMS:

1. Introduction

The main purpose of this thesis is to describe and analyze a method for solving discrete time nonlinear constrained optimal control problems. However most of these problems originate from continuous time systems, but are for practical reasons or by tradition defined as discrete time systems. For instance most economic systems are usually described as discrete time systems. Also, when a digital computer is used for controlling a system, a discrete time description of the system is suitable, because when a digital computer is used, the controls are usually held constant between the sample times.

Hence we need a method to get a discrete time description of the system we are interested in. In this thesis we have assumed that this discrete time description is given or easily obtained. In this appendix we will describe how to solve the discrete time description of the system even when it is impossible to explicitly find the functions f and ℓ in (4.1) and (4.3).

We assume that we have a continuous time description of the system, namely

$$\dot{x}(\tau)=F(x(\tau),u(\tau),\tau);0\leq\tau\leq t_F \tag{A.1}$$

and that we want to minimize the performance index

$$J(u)=\ell(x(t_F),t_F)+\int_0^{t_F} L(x(\tau),u(\tau))d\tau \tag{A.2}$$

We assume that the initial values of the states are given by

$$x(0)=x_0 \tag{A.3}$$

The problem is now that we are only allowed to change the values of the controls at certain times, the sampling times.

We will assume here that these sampling times are equidistant and also that the distance between two consequtive sampling times is 1. (This is not necessary. We make these assumptions to get easier notations). Hence the sampling times are the integers $0,1,\ldots,N-1$. We also assume that at these sampling times we have constraints of the type

$$h^i(x(t_i),u(t_i))\leq 0, i=1,\ldots,p. \tag{A.4}$$

We also assume that the controls are right continuous, i.e.

$$u(\tau)=u(t) \text{ if } t\leq\tau<t+1. \tag{A.5}$$

We also assume that the solutions to all the differential equations in this appendix exist, are unique and are bounded on the intervals on which they are defined.

If the state $x(t)$ and the control $u(t)$ are known, we can calculate $x(t+1)$ by solving the differential equation (A.1), since from (A.5) we know the controls $u(\tau)$. Hence the function F in (A.1) implicitly defines the function $f(x(t),u(t),\ell)$ in the discrete-time description (4.1) of the system. Also in (4.3) the values of the function $\ell(x(t),u(t),t)$ could be evaluated as

$$\ell(x(t),u(t),t)=\int_t^{t+1} L(x(\tau),u(\tau),\tau)d\tau \tag{A.6}$$

So we conclude that the problem defined by (A.1-5) is actually of the same type as that described in chapter 4, but usually the functions f and ℓ are impossible to get explicitly.

In the following we will use the notations $\alpha(t^+)$ and $\alpha(t^-)$ when we mean the right hand limit and the left hand limit of the piece-wise continuous function $\alpha(t)$.

Hence if $\varepsilon>0$ we have

$$\alpha(t^+)=\lim_{\varepsilon\to 0}\ \alpha(t+\varepsilon) \tag{A.7a}$$

$$\alpha(t^-)=\lim_{\varepsilon\to 0}\ \alpha(t-\varepsilon). \tag{A.7b}$$

In the next section we derive the necessary equations when there are no constraints of the type (A.4) and in section 3 we extend these results to include these constraints.

2. Solving the unconstrained problem

We will derive a technique for solving the optimal control, defined in the previous section. When we derive this technique we will closely follow the same lines as those in section 4.3.1, when we derived the technique for discrete time systems. Consequently we let $z(t)$ consist of the controls from time t up to $N-1$ as in (4.6). Hence

$$z(t)=(u^T(N-1),\ldots,u^T(t))^T. \tag{A.8}$$

In (4.8) we defined the function $V(x(t),z(t),t)$ as the cost from time t up to time N starting at $x(t)$ and using the control defined by $z(t)$. So here we, for $t\leq\tau<t+1$, define

$$V(x(\tau),z(t),\tau)=\ell(x(N),N)+\int_{\tau}^{N} L(x(s),u(s),s)ds \tag{A.9}$$

where $x(s)$, for $s\geq\tau$, is the solution to (A.1) starting at $x(\tau)$, and $u(s)$ is defined by (A.8) and (A.5). As in section 4.3.1 we now want to calculate the derivatives of $V(t^+)$ with respect to x, z, xx, xz, and zz. We do that in the following way. The derivative of the function $V(\tau)$ in (A.9) with respect to τ is

$$\dot{V}(\tau)=V_\tau(\tau)+V_x(\tau)F(\tau)=-L(\tau) \tag{A.10}$$

if $t<\tau<t+1$. (The term $V_z(\tau)\dot{z}(t)$ in the expression above vanishes because $z(t)$ is independent of τ on the interval.)

From (A.10) we get the equality

$$V_\tau(t)+L(\tau)+V_x(\tau)F(\tau)=0. \tag{A.11}$$

By differentiating (A.11) with respect to x, z, xx, xz and zz we get the following equalities.

$$V_{x\tau}(\tau)+L_x(\tau)+F^T(\tau)V_{xx}(\tau)+V_x(\tau)F_x(\tau)=0 \tag{A.12a}$$

$$V_{z\tau}(\tau)+(0,L_u(\tau))+F^T(\tau)V_{xz}(\tau)+(0,V_x(\tau)F_u(\tau))=0 \tag{A.12b}$$

$$V_{xx\tau}(\tau)+L_{xx}(\tau)+F_x^T(\tau)V_{xx}(\tau)+F^T(\tau)V_{xxx}(\tau)+$$

$$+V_{xx}(\tau)F_x(\tau)+V_x(\tau)F_{xx}(\tau)= 0 \tag{A.12c}$$

$$V_{xz\tau}(\tau)+(0,L_{xu}(\tau))+F^T(\tau)V_{xxz}(\tau)+(0,V_{xx}(\tau)F_u(\tau))+$$

$$F_x^T(\tau)V_{xz}(\tau)+(0,V_x(\tau)F_{xu}(\tau)) \tag{A.12d}$$

$$V_{zz\tau}(\tau)+\begin{pmatrix}0 & 0\\ 0 & L_{uu}(\tau)\end{pmatrix}+F^T(\tau)V_{xzz}(\tau)+\begin{pmatrix}0\\ F_u^T(\tau)V_{xz}(\tau)\end{pmatrix}+$$

$$+(0,V_{zx}(\tau)F_u(\tau))+\begin{pmatrix}0 & 0\\ 0 & V_x(\tau)F_{uu}(\tau)\end{pmatrix}=0 \tag{A.12e}$$

Using the equalities in (A.12) we can derive the following differential equalities in the interval $t<\tau<t+1$.

$$-\dot{V}_x(\tau)=-V_{x\tau}(\tau)-F^T(\tau)V_{xx}(\tau)=L_x(\tau)+V_x(\tau)F_x(\tau) \tag{A.13a}$$

$$-\dot{V}_z(\tau)=-V_{z\tau}(\tau)-F^T(\tau)V_{xz}(\tau)=(0,L_u(\tau)+V_x(\tau)F_u(\tau)) \tag{A.13b}$$

$$-\dot{V}_{xx}(\tau)=-V_{xx\tau}(\tau)-F^T(\tau)V_{xxx}(\tau)=L_{xx}(\tau)+F_x^T(\tau)V_{xx}(\tau)$$

$$+V_{xx}(\tau)F_x(\tau)+V_x(\tau)F_{xx}(\tau) \tag{A.13c}$$

$$-\dot{V}_{xz}(\tau)=-V_{xz\tau}(\tau)-F^T(\tau)V_{xxz}(\tau)=$$

$$=F_x^T(\tau)V_{xz}(\tau)+(0,L_{xu}(\tau)+V_x(\tau)F_{xu}(\tau)) \qquad \text{(A.13d)}$$

$$-\dot{V}_{zz}(\tau)=-V_{zz\tau}(\tau)-F^T(\tau)V_{xzz}(\tau)=$$

$$=\begin{pmatrix}0\\ F_u^T(\tau)V_{xz}(\tau)\end{pmatrix}+(0,V_{zx}(\tau)F_u(\tau))+\begin{pmatrix}0 & 0\\ 0 & L_{uu}(\tau)+V_x(\tau)F_{uu}(\tau)\end{pmatrix}$$

(A.13d)

From the definition (A.9) of the function $V(\tau)$ and the construction (A.8) of the vector $z(t)$, it is obvious that we have the following boundary conditions.

$$V_x(N^-)=\ell_x(N) \qquad \text{(A.14a)}$$

$$V_z(N^-)=0 \qquad \text{(A.14b)}$$

$$V_{xx}(N^-)=\ell_{xx}(N) \qquad \text{(A.14c)}$$

$$V_{xz}(N^-)=0 \qquad \text{(A.14d)}$$

$$V_{zz}(N^-)=0 \qquad \text{(A.14e)}$$

and for $t=1,\ldots,N-1$ we have

$$V_x(t^-)=V_x(t^+) \qquad \text{(A.14f)}$$

$$V_z(t^-)=(V_z(t^+),0) \qquad \text{(A.14e)}$$

$$V_{xx}(t^-)=V_{xx}(t^+) \qquad \text{(A.14g)}$$

$$V_{xz}(t^-)=(V_{xz}(t^+),0) \qquad \text{(A.14h)}$$

$$V_{zz}(t^-)=\begin{pmatrix} V_{zz}(t^+) & 0 \\ 0 & 0 \end{pmatrix} \qquad (A.14i)$$

Following the ideas of chapter 4.3.1 we continue by defining $W(\Delta x(\tau),\Delta z(t),\tau)$ as the second order Taylor expansion of the function $V(\tau)$ around the point $(x(\tau),z(t))$ on the interval $t<\tau<t+1$. Hence

$$W(\Delta x(\tau),\Delta z(t)\Delta z(t),\tau)=V(\tau)+V_x(\tau)\cdot\Delta x(\tau)+$$

$$V_z(\tau)\cdot\Delta z(t)+\frac{1}{2}\,(\Delta x^T(\tau)V_{xx}(\tau)\Delta x(\tau)+$$

$$2\cdot\Delta x(\tau)V_{xz}(\tau)\Delta z(t)+\Delta z^T(t)V_{zz}(\tau)\Delta z(t) \qquad (A.15)$$

We now assume that $\Delta x(\tau)$ satisifies the differential equation

$$\Delta\dot{x}(\tau)=F_x(\tau)\Delta x(\tau)+F_u(\tau)\Delta u(t) \qquad (A.16)$$

on the interval $t<\tau<t+1$. (cf the role of the auxiliary variable $D(t+1)$ in (4.16).) From the variation of constants formula in Brockett (1970), p.40, we can write $\Delta x(t+1)$ as

$$\Delta x(t+1)=\varphi(t+1,\tau)\Delta x(\tau)+\int_{\tau}^{t+1}\varphi(t+1,s)F_u(s)ds\Delta u(t) \qquad (A.17)$$

where the transfer matrix $\varphi(\tau,s)$ is defined by

$$\frac{d}{d\tau}\,\varphi(\tau,s)=F_x(\tau)\varphi(\tau,s);\varphi(s,s)=I. \qquad (A.18)$$

We now try to find the analogy of equation (4.18) in which $W(t)$ is expressed as a function of $W(t),\Delta x(t)$ and $\Delta u(t)$. Hence we put up the following expression for $W(\tau)$

$$W(\Delta x(\tau),\Delta z(\tau),\tau)=W(\Delta x(t+1),\Delta z(t+1),t+1)+$$

$$+\ell(\tau)+\ell_x(\tau)\Delta x(\tau)+\ell_u(\tau)\Delta u(t)+\frac{1}{2}\,(\Delta x^T(\tau)Q_{xx}(\tau)\Delta x(\tau)$$

$$+2\Delta x^T(\tau)Q_{xu}(\tau)\Delta u(t)+\Delta u^T(t)Q_{uu}(\tau)\Delta u(t)) \qquad (A.19)$$

where $\ell(\tau),\ell_x(\tau),\ell_u(\tau),Q_{xx}(\tau),Q_{xu}(\tau)$ and $Q_{uu}(\tau)$ are to be determined.

We can do that in the following way. Eliminate $\Delta x(t+1)$ in (A.19) by using the expression (A.17). We then, after some manipulations, get a quadratic expression in the variables $\Delta x(\tau),\Delta z(t+1)$ and $\Delta u(t)$. By identifying the coefficients in this expression with the coefficients in (A.15) we get

$$\ell(\tau)=V(\tau)-V(t+1) \qquad (A.18a)$$

$$\ell_x(\tau)=V_x(\tau)-V_x(t+1)\varphi(t+1,\tau) \qquad (A.18b)$$

$$\ell_u(\tau)=[V_z(\tau)]_{u(t)}-V_x(t+1)\int_\tau^{t+1}\varphi(t+1,s)F_u(s)ds \qquad (A.18c)$$

$$Q_{xx}(\tau)=V_{xx}(\tau)-\varphi^T(t+1,\tau)V_{xx}(t+1)\varphi(t+1,\tau) \qquad (A.18d)$$

$$Q_{xu}(\tau)=[V_{xz}(\tau)]_{u(t)}-$$

$$-\varphi^T(t+1,\tau)V_{xx}(t+1)\int_\tau^{t+1}\varphi(t+1,s)F_u(s)ds \qquad (A.18e)$$

$$Q_{uu}(\tau)=[V_{zz}(\tau)]_{u(t)}-$$

$$-\left(\int_\tau^{t+1}\varphi(t+1,s)F_u(s)ds\right)^T V_{xx}(t+1)\left(\int_\tau^{t+1}\varphi(t+1,s)F_u(s)ds\right) \qquad (A.18f)$$

Here $[\quad]_{u(t)}$ means that we only use the part of the matrix within the brackets that corresponds to u(t). If we differentiate relations (A.18) with respect to τ and use (A.10) and (A.13) we obtain the following differential equations.

$$-\dot{\ell}(\tau)=L(\tau) \qquad (A.19a)$$

$$-\dot{\ell}_x(\tau)=L_x(\tau)+\ell_x(\tau)F_x(\tau) \tag{A.19b}$$

$$-\dot{\ell}_u(\tau)=L_u(\tau)+\ell_x(\tau)F_u(\tau) \tag{A.19c}$$

$$-\dot{Q}_{xx}(\tau)=L_{xx}(\tau)+V_x(\tau)F_{xx}(\tau)+F_x^T(\tau)Q_{xx}(\tau)+Q_{xx}(\tau)F_x(\tau) \tag{A.19d}$$

$$-\dot{Q}_{xu}(\tau)=L_{xu}(\tau)+V_x(\tau)F_{xu}(\tau)+F_x^T(\tau)Q_{xu}(\tau)+Q_{xx}(\tau)F_u(\tau) \tag{A.19e}$$

$$-\dot{Q}_{uu}(\tau)=L_{uu}(\tau)+V_x(\tau)F_{uu}(\tau)+F_u^T(\tau)Q_{xu}(\tau)+Q_{ux}(\tau)F_u(\tau) \tag{A.19f}$$

By taking the limits of the expressions (A.18) as τ tends to $(t+1)^-$ we get that all variables are zero at $(t+1)^-$.

Again we use the function $J^*(\Delta x(t),t)$ defined in section (4.3.1) as

$$J^*(\Delta x(t),t)= \min_{\Delta z(t)} W(\Delta x(t^*),\Delta(t),t^*) \tag{A.20}$$

This function is only defined at the times $t=0,\ldots,N-1$. From (A.15) we see that the $\Delta z(t)$ that minimizes $W(\Delta x(t^+),\Delta z(t),t^+)$ is given by

$$\Delta z(t)=-V_{zz}(t)^{-1}(V_z^T(t)+V_{xz}^T(t)\Delta x(t)) \tag{A.21}$$

If we put this value in (A.15) we find that $J^*(\Delta x(t),t)$ is a quadratic function of $\Delta x(t)$. We therefore put up the following expression for $J^*(\Delta x(t),t)$.

$$J^*(\Delta x(t),t)=a(t)+W_x(t)\Delta(t)+\frac{1}{2}\Delta x^T(t)W_{xx}(t)\Delta x(t) \tag{A.22}$$

Now assume that we have a rule for calculating $\Delta z(t)$ in (A.21) for $t=N,\ldots,p+1$. (For $t=N$ it is trivial!) We will then calculate $\Delta u(t)$ for $t=p$ in the following way. For $p<\tau<p+1$ we define the function $W^O(\Delta x(\tau),\Delta u(t),\tau)$ as

$$W^{o}(\Delta x(\tau),\Delta u(p),\tau)=J^{*}(\Delta x(p+1),p+1)+$$

$$+\ell(\tau)+\ell_{x}(\tau)\Delta x(\tau)+\ell_{u}(\tau)\Delta u(p)+$$

$$+\frac{1}{2}\ (\Delta x^{T}(\tau)Q_{xx}(\tau)\Delta x(\tau)+2\Delta x^{T}(\tau)Q_{xu}(\tau)\Delta u(p)+$$

$$+\Delta u^{T}(p)Q_{uu}(\tau)\Delta u(p)) \qquad (A.23)$$

That is $W^{o}(\tau)$ is the value of the function $W(\tau)$ given by (A.15), but when $\Delta z(p+1)$is chosen so that $W(\Delta x(p+1),\ \Delta z(p+1),p+1)$ is minimized.

We now put up the following expression for $W^{o}(\Delta x(\tau),\Delta u(p),\tau)$.

$$W^{o}(\Delta x(\tau),\Delta z(p),\tau)=a^{o}(\tau)+$$

$$+W_{x}^{o}(\tau)\Delta x(\tau)+W_{u}^{o}(\tau)\Delta u(p)+$$

$$+\frac{1}{2}\ (\Delta x^{T}(\tau)W_{xx}^{o}(\tau)\Delta x(\tau)+2\Delta x^{T}(\tau)W_{xu}^{o}(\tau)\Delta u(p)+$$

$$+\Delta u^{T}(p)W_{uu}^{o}(\tau)\Delta u(p)) \qquad (A.24)$$

If we now again use the Variation of Constants Formula (A.17) to eliminate the variables $\Delta x(p+1)$ in (a.23) and identify the coefficients we get

$$a^{o}(\tau)=\ell(\tau)+a(p+1) \qquad (A.25a)$$

$$W_{x}^{o}(\tau)=\ell_{x}(\tau)+W_{x}(p+1)\varphi(p+1,\tau) \qquad (A.25b)$$

$$W_{u}^{o}(\tau)=\ell_{u}(\tau)+W_{x}(p+1)\int_{\tau}^{p+1}\varphi(p+1,s)F_{u}(s)ds \qquad (A.25c)$$

$$W_{xx}^{o}(\tau)=Q_{xx}(\tau)+\varphi^{T}(p+1,\tau)W_{xx}(p+1)\varphi(p+1,\tau) \qquad (A.25d)$$

$$W^{o}_{xu}(\tau)=Q_{xu}(\tau)+\varphi^{T}(p+1,\tau)W_{xx}(p+1)\int_{\tau}^{p+1}\varphi(p+1,s)F_{u}(s)ds \qquad (A.25e)$$

$$W^{o}_{uu}(\tau)=Q_{uu}(\tau)+(\int_{\tau}^{p+1}\varphi(p+1,s)F_{u}(s)ds)^{T}\cdot W_{xx}(p+1)\cdot$$

$$\cdot\int_{\tau}^{p+1}\varphi(p+1,s)F_{u}(s)ds \qquad (A.25f)$$

By differentiating the equalities in (A.25) we get the following differential equation,

$$-\dot{a}^{o}(\tau)=L(\tau) \qquad (A.26a)$$

$$-\dot{W}^{o}_{x}(\tau)=L_{x}(\tau)+W^{o}_{x}(\tau)F_{x}(\tau) \qquad (A.26b)$$

$$-\dot{W}^{o}_{u}(\tau)=L_{u}(\tau)+W^{o}_{x}(\tau)F_{u}(\tau) \qquad (A.26c)$$

$$-\dot{W}^{o}_{xx}(\tau)=L_{xx}(\tau)+V_{x}(\tau)F_{xx}(\tau)+F^{T}_{x}(\tau)W^{o}_{xx}(\tau)+W^{o}_{xx}(\tau)F_{x}(\tau) \qquad (A.26d)$$

$$-\dot{W}^{o}_{xu}(\tau)=L_{xu}(\tau)+V_{x}(\tau)F_{xu}(\tau)+F^{T}_{x}(\tau)W^{o}_{xu}(\tau)+W^{o}_{xx}(\tau)F_{u}(\tau) \qquad (A.26e)$$

$$-\dot{W}^{o}_{uu}(\tau)=L_{uu}(\tau)+V_{x}(\tau)F_{uu}(\tau)+F^{T}_{u}(\tau)W^{o}_{xu}(\tau)+W^{o}_{ux}(\tau)F_{u}(\tau) \qquad (A.26f)$$

If we let τ tend to $(p+1)^{-}$ in (A.25) we see that the differential equations in (A.26) have the limits

$$a^{o}((p+1)^{-})=a(p+1) \qquad (A.27a)$$

$$W^{o}_{x}((p+1)^{-})=W_{x}(p+1) \qquad (A.27b)$$

$$W^{o}_{u}((p+1)^{-})=0 \qquad (A.27c)$$

$$W^{o}_{xx}((p+1)^{-})=W_{xx}(p+1) \qquad (A.27d)$$

$$W^{o}_{xu}((p+1)^{-})=0 \qquad (A.27e)$$

$$W^o_{uu}((p+1)^-)=0 \tag{A.27f}$$

We now solve the differential equations (A.26) with the limits (A.27) to obtain the value of the function $W^o(\tau)$ in (A.24) for $\tau=p^+$. At this time the minimizing $\Delta u(p)$ is given by

$$\Delta u(p)=-W^o_{uu}(p^+)^{-1}(W^o_u(p^+)^T+W^o_{xu}(p^+)^T\Delta x(p^+)) \tag{A.28}$$

and hence for t=p the coefficients in (A.22) is given by,

$$a(p)=a^o(p^+)-W^o_u(p^+)W^o_{uu}(p^+)^{-1}W^o_u(p^+)^T \tag{A.29a}$$

$$W_x(p)=W^o_x(p^+)-W^o_u(p^+)W^o_{uu}(p^+)^{-t}W^o_{ux}(p^+) \tag{A.29b}$$

$$W_{xx}(p)=W^o_{xx}(p^+)-W^o_{xu}(p^+)W^o_{uu}(p^+)^{-1}W^o_{ux}(p^+) \tag{A.29c}$$

if we introduce the notations

$$\delta u(p)=-W^o_{uu}(p^+)^{-1}-W^o_u(p^+)^T \tag{A.30a}$$

$$\beta_p=-W^o_{uu}(p^+)^{-1}W^o_{xu}(p^+)^T \tag{A.30b}$$

the equation (A.28) reduces to

$$\Delta u(p)=\delta u(p)-\beta_p\Delta x(p) \tag{A.31}$$

From theorem 4.1 we conclude that the Newton step towards the solution of the problem (A.1)(A.2)(A.3) and (A.5) is given by (A.28) where $\Delta x(\tau)$ is the solution to the differential equation (A.16) and the other coefficients in (A.28) are given from (A.26), (A.27) and (A.29).

Hence the algorithm to solve, (A.1-3),(A.5) is as follows.

1. Assume that we have a control sequence $u_0(t)$, t=0,...,N-1. Put k=0.

2. For the control sequence $u_k(t), t=0,\ldots,N-1$ calculate the corresponding states $x_k(\tau)$ using (A.1) and (A.3) and the value $J(u_k)$ of the criterion function in (A.2).

3. For $t=N-1,\ldots,0$ solve the differential equations (A.13a) and (A.26) on the intevals $[t,t+1]$. During these calculations compute and store the values of $\delta u(t)$ and β_t given by (A.30). If the matrix $W^o_{uu}(t^+)$ becomes singular or indefinite for some t, use the method proposed in section 6.2.

4. For $t=0,\ldots,N-1$, calculate and store the value of $\Delta u(t)$ from (A.31) and solve the differential equation (A.16) on the interval $[t,t+1]$.

5. If $|\Delta u(t)|<\varepsilon$ $\forall t$, where ε is a small positive number, go to step 7.

6. Put $u_{k+1}(t)=u_k(t)+\alpha_k\Delta u(t), t=0,\ldots,N-1$, where α is chosen according to the rules given in section 7. Put k=k+1 and go back to step 2.

7. Stop. $u_k(t)$ is probably close to a local minimizing control sequence to the problem (A.1), (A.2), (A.3) and (A.5).

3. Solving the constrained problem.

The constraints only influence the equations at times $0,1,\ldots,N$ and not between these times. Hence, the differential equations remain the same, only the boundary conditions are changed. As in section 4.3.3 we use the function $\gamma(t)$, defined by (4.50), to obtain the influence on the constraints. Hence,

$$\gamma(t)=\gamma(x(t),u(t),\mu,t)=\sum_{i\in I(t)}\mu_i h^i(x(t_i),u(t_i)) \qquad (A.32)$$

where $I(t)$ is defined by (4.49). The values of μ are the ones

obtained from the previous iteration.

From the resemblance between the equations in section (4.3) and in this appendix it is obvious that the Riccati equations now become

$$V_x(N^-)=\ell_x(N)+\gamma_x(N) \tag{A.33a}$$

$$W^o_{xu}(N^-)=\ell_{xx}(N)+\gamma_{xx}(N) \tag{A.33b}$$

$$W^o_{xu}(N^-)=0 \tag{A.33c}$$

$$W^o_{uu}(N^-)=0 \tag{A.34d}$$

Between the sampling times these variables satisfy the following differential equations. (cf equations (A.13a) and (A.26)).

$$-\dot{V}_x(\tau)=L_x(\tau)+V_x(\tau)F_x(\tau) \tag{A.34a}$$

$$-\dot{W}^o_{xx}(\tau)=L_{xx}(\tau)+V_x(\tau)F_{xx}(\tau)+ F^T_x(\tau)W^o_{xx}(\tau)+W^o_{xx}(\tau)F_x(\tau) \tag{A.34b}$$

$$-\dot{W}^o_{xu}(\tau)=L_{xu}(\tau)+V_x(\tau)F_{xu}(\tau)+F^T_x(\tau)W^o_{xu}(\tau)+W^o_{xx}(\tau)F_u(\tau) \tag{A.34c}$$

$$-\dot{W}^o_{uu}(\tau)=L_{uu}(\tau)+V_x(\tau)F_{uu}(\tau)+F^T_u(\tau)W^o_{xu}(\tau)+W^o_{ux}(\tau)F_u(\tau) \tag{A.34d}$$

At the times t=N-1,...,0, calculate and store the matrices C_t^{-1} and β_t given by

$$C_t=W^o_{uu}(t^+)+\gamma_{uu}(t) \tag{A.35a}$$

$$\beta_t=C_t^{-1}(W^o_{ux}(t^+)+\gamma_{ux}(t)) \tag{A.35b}$$

Again if C_t is indefinite or singular we use the method proposed in section 6.2 to make it positive definite. At the times

t=N-1,...,1 we have the following boundary conditions.

$$V_x(t^-)=V_x(t^+)+\gamma_x(t) \tag{A.36a}$$

$$W^o_{xx}(t^-)=W^o_{xx}(T^+)+\gamma_{xx}(t)-(W^o_{xu}(t^+)+\gamma_{xu}(t))\beta_t \tag{A.36b}$$

$$W^o_{xu}(t^-)=0 \tag{A.36c}$$

$$W^o_{uu}(t^-)=0 \tag{A.36d}$$

The trajectory corresponding to the unconstrained optimum is now obtained from the following two steps.

Step 1. Start with

$$W^o_x(N^-)=\ell_x(N) \tag{A.37a}$$

$$W^o_u(N^-)=0 \tag{A.37b}$$

For t=N-1,...,0 find the solution to

$$-\dot{W}^o_x(\tau)=L_x(\tau)+W^o_x(\tau)F_x(\tau) \tag{A.38a}$$

$$-\dot{W}^o_u(\tau)=L_u(\tau)+W^o_x(\tau)F_u(\tau) \tag{A.38b}$$

On the interval (t,t+1)

and let $\delta u_0(t)$ be given by

$$\delta u_0(t)=-C_t^{-1}W^o_u(t^+)^T \tag{A.39}$$

For t=N-1,...,1 we have the boundary conditions

$$W^o_x(t^-)=W^o_x(t^+)-W^o_u(t^+)\beta_t \tag{A.40a}$$

$$W_u^o(t^-)=0 \tag{A.40b}$$

Step 2. Start with $\Delta x_0(0)=0$ and for $t=0,\ldots,N-1$ let $\Delta u_0(t)$ be given by

$$\Delta u_0(t)=\delta u_0(t)-\beta_t \Delta x_0(t) \tag{A.41}$$

and solve the differential equation

$$\Delta \dot{x}_0(\tau)=F_x(\tau)\Delta x_0(\tau)+F_u(\tau)\Delta u_0(t) \tag{A.42}$$

Store the values of $\Delta u_0(t), t=0,\ldots,N-1$ and $\Delta x_0(t), t=0,\ldots,N$.

For each constraint i in the active set we perform the following two steps.

Step 1. For $t>t_i$ put

$$\delta u_i(t)=0 \tag{A.43}$$

For $t=t_i$ we have the conditions

$$\delta u_i(t_i)=-C_{t_i}^{-1} h_u^i(t_i)^T \tag{A.44a}$$

$$W_x^i(t_i^-)=h_x^i(t_i)-h_u^i(t_i)\beta_{t_i} \tag{A.44b}$$

$$W_x^i(t_i^-)=0 \tag{A.44c}$$

For $t=t_i-1,\ldots,0$ we solve on the intervals $(t,t+1)$ the differential equations

$$-\dot{W}_x^i(\tau)=W_x^i(\tau)F_x(\tau) \tag{A.45a}$$

$$-\dot{W}_u^i(\tau)=W_x^i(\tau)F_u(\tau) \tag{A.45b}$$

For $t=t_i-1,\ldots,1$ the boundary conditions are

$$W_x^i(t^-)=W_x^i(t^+)-W_u^i(t^+)\beta_t \tag{A.45a}$$

$$W_u^i(t^-)=0 \tag{A.46b}$$

For $t=t_i-1,\ldots,0$ we calculate

$$\delta u_i(t)=-C_t^{-1}W_u^i(t^+)^T \tag{A.47}$$

During these calculations we store the values of $\delta u_i(t)$, $t=0,\ldots,N-1$.

Step 2. Start with $\Delta x_i(0)=0$ and for $t=0,\ldots,N-1$ let $\Delta u_i(t)$ be given by

$$\Delta u_i(t)=\delta u_i(t)-\beta_t\Delta x_i(t) \tag{A.48}$$

and solve the differential equation

$$\Delta\dot{x}_i(\tau)=F_x(\tau)\Delta x_i(\tau)+F_u(\tau)\Delta u_i(t) \tag{A.49}$$

on the interval $(t,t+1)$. During these calculate for each constraint j in the active set, the value

$$a_{ij}=h_x^j(t_j)\Delta x_i(t_j)+h_u^j(t_j)\Delta u_i(t_j). \tag{A.50}$$

The values a_{ij} in (A.50) define the elements in the matrix A given by (3.9). The elements in the vector d in (3.10) are given by

$$d_i=h^i(t_i)+h_x^i(t_i)\Delta x_0(t_i)+h_u^i(t_i)\Delta u_0(t_i) \tag{A.51}$$

It is trivial to implement the method for simple constraints on the controls given in section (8.3), because we only need to use the matrix $I_oC_t^{-1}$ instead of C_t^{-1} in the equations (A.35b),

(A.39), (A.44a) and (A.47). The boundary conditions for $W^{o}_{xx}(\tau)$ in (A.36b) are changed to

$$W^{o}_{xx}(t^{-})=W^{o}_{xx}(t^{+})+\gamma_{xx}(t)-$$

$$-(W^{o}_{xu}(t^{+})+\gamma_{xu}(t))(I_{o}C_{t}^{-1}+C_{t}^{-1}I_{o}^{T}-C_{t}^{-1}I_{o}^{T}C_{t}I_{o}C_{t}^{-1})\cdot$$

$$\cdot(W^{o}_{ux}(t^{+})+\gamma_{ux}(t)). \qquad (A.52)$$

which follows from (8.18c).

REFERENCES

Arriatti, eds. (1971)
Preliminary report on optimization studies. CIGRE Study Committee 32. Working Group A1. Rome April 8th, 1971.

Bartels, R.H., G.H. Golub and M.A. Saunders. (1970)
Numerical Techniques in Mathematical Programming. In J.B. Rosen, O.L. Mangasarian and K. Ritter, eds., Nonlinear Programming, Academic Press, New York, pp. 123-176.

Bellman, R.E. and S.E. Dreyfus. (1962)
Applied Dynamic Programming. Princeton University Press, Princeton.

Bertsekas, D.P. (1982)
Constrained Optimization and Lagrange Multiplier Methods. Academic Press, New York.

Brockett, R.W. (1970)
Finite Dimensional Linear Systems. John Wiley and Sons Inc., New York.

Bryson, A.E. and Y.-C. Ho. (1969)
Applied Optimal Control. Ginn and Company, Waltham, Mass.

Bunch, J.T. and L. Kaufman (1980)
A Computational Method for the Indefinite Quadratic Programming Problem. Linear Algebra Applications, Vol. 34, pp. 348-370.

Canon, M.D., C.D. Cullum and E. Polak (1970)
Theory of Optimal Control and Mathematical Programming. McGraw-Hill Book Company, New York.

Chamberlain, R.M. , C. Lemarechal, H.C. Pedersen and M.J.D. Powell (1979)
The Watchdog Technique for Forcing Convergence in Algorithms for Constrained Optimization. Tenth International Symposium on Mathematical Programming, Montreal, August 1979.

Dahlquist, G. and Å. Björck (1974)
Numerical Methods. Prentice-Hall Inc., Englewood Cliffs, New Jersey.

Dyer, P. and S.R. McReynolds (1970)
The Computation and Theory of Optimal Control. Academic Press, New York.

Fletcher, R. (1980)
Practical Methods of Optimization. Volume 1. John Wiley and Sons, New York.

Fletcher, R. (1981)
Practical Methods of Optimization. Volume 2. John Wiley and Sons, New York.

Gill, P.E., G.H. Golub, W. Murray and M.A. Saunders (1974)
Methods for Modifying Matrix Factorizations. Mathematics of Computation, Vol. 28, No. 126 pp. 505-535.

Gill, P.E. and W. Murrary (1978)
Numerically Stable Methods for Quadratic Programming. Mathematical Programming, 14, pp. 349-372.

Gill, P.E., W. Murrary and S.M. Picken. (1972)
The Implementation of Two Modified Newton Algorithms for Unconstrained Optimization. NPL Report NAC24.

Goldfarb, D. (1975)
Matrix Factorizations in Optimization of Nonlinear Functions Subject to Linear Constraints. Mathematical Programming 10, pp. 1-31.

Han, S.-P. (1975)
Superlinearly Convergent Variable Metric Algorithms for General Nonlinear Programming Problems. TR 75-233, Cornell University.

Han, S.-P. (1977)
A Globally Convergent Method for Nonlinear Programming. TR 75-257, Cornell University. / J. of Optimization Theory and Applications, Vol. 22, pp. 297-309.

Jacobson, D.H. and D.Q. Mayne (1970)
Differential Dynamic Programming. American Elsevier Publishing Company, Inc., New York, 1970.

Järmark, B.S.A. (1976)
Convergence Control in Differential Dynamic Programming Applied to Air-to-air Combat. AIAA J. 14(1), pp. 118-121.

Kwakernaak, H. and R. Sivan (1972)
Linear Optimal Control Systems. Wiley-Interscience, New York.

Lasdon, L.S. (1979)
Optimization Theory for Large Systems. MacMillan Publishing Company Inc., New York.

Luenberger, D.G. (1973)
Introduction to Linear and Nonlinear Programming. Addison Wesley, Reading, Mass.

Luenberger, D.G. (1979)
Introduction to Dynamic Systems. John Wiley and Sons, New York.

Mayne, D. (1966)
A Second-Order Gradient Method for Determining Optimal Trajectories of Non-Linear Discrete-Time Systems. Int. J. Control, Vol. 3, No.1, pp. 85-95.

McReynolds, S.R. and A.E. Bryson. (1965)
A Successive Sweep Method for Solving Optimal Programming Problems. Joint Autom. Control Conf., pp. 551-555. Troy, New York, June 1965.

Mitter, S.K. (1966)
Successive Approximation Methods for the Solution of Optimal Control Problems. Automatica, Vol. 3, pp. 135-149.

Ohno, K. (1978)
A New Approach to Differential Dynamic Programming for Discrete Time Systems. IEEE Trans. on Automatic Control, Vol. AC-23, No. 1, pp. 37-47.

Paige, C.C. and M.A. Saunders (1975)
Solution of Sparse Indefinite Systems of Linear Equations. SIAM J. Numer. Anal. Vol. 12, No. 4, pp. 617-629.

Polak, E. (1973)
An Historical Survey of Computational Methods in Optimal Control. SIAM Review, Vol. 15, No.2, pp. 553-584.

Powell, M.J.D. (1978)
A Fast Algorithm for Nonlinearly Constrained Optimization. In Numerical Analysis, Dundee, 1977, Lecture Notes in Mathematics 630. Springer Verlag, Berlin, pp. 144-157.

Powell, M.J.D. (1981)
An Upper Triangular Matrix Method for Quadratic Programming. In O.L. Mangasarian et.al. eds., Nonlinear Programming 4. Academic Press, New York. pp.1-24.

Rudin, W. (1964)
Principles of Mathematical Analysis. 2nd ed. McGraw-Hill Book Company, New York.

Talukdar, S.N. and T.C. Giras (1982)
Robust Variable Metric Method for Optimum Power Flows. IEEE Trans Power Apparatus and Systems, Vol PAS-101 No 2.

Trulsson, E. (1983)
Adaptive Control based on Explicit Criterion Minimization. Ph.D. Dissertation, Linköping University

Wilson, R.B. (1963)
A Simplicial Method for Concave Programming. Ph.D. dissertation, Harvard University, Cambridge, Mass.