THE APPLICATION OF GRADIENT ALGORITHMS TO THE OPTIMIZATION OF CONTROLLED VERSIONS OF THE WORLD 2 MODEL OF FORRESTER

J.L. de Jong
Department of Mathematics
Eindhoven University of Technology

and

J.W.Dercksen
Netherlands Organization for the Advancement of Pure Research (Z.W.O.)
Department of Physics
Eindhoven University of Technology
P.O.Box 513, Eindhoven, The Netherlands

1. INTRODUCTION

In early 1972, shortly after the results in Forrester's book "World Dynamics" (Forrester (1971)) had arosed the interest of many people in the study of world models, a project group, named "Global Dynamics" was started in the Netherlands (cf.Rademaker (1972)) which set itself as one of its goals to study the effects of the incorporation of controls into the world models considered by the M.I.T. groups of Forrester and Meadows under sponsorship of the Club of Rome (cf. Meadows (1972)).

One way to get a better understanding of a controlled system is to determine the optimal controls given suitably chosen optimization criteria and to study the sensitivity of these optimal controls to changes in model and criterion parameters. An essential tool in such a study is an efficient algorithm (or better: computer program) for the numerical solution of optimal control problems of the particular type at hand. In case of the "Global Dynamics" project, in which several Dutch universities and companies cooperated, several groups set out to test different classes of known numerical optimal control algorithms in order to select the one best suited to generate the many optimal solutions required for the project. Two of these groups already reported their results (cf. Olsder & Strijbos (1973), Dekker & Kerckhoffs (1974)).

At Eindhoven University of Technology a special experimental program was set up to compare the performance of different known gradient type algorithms. These were applied to the common test problem of the project which consisted of a simplified version of the controlled world model of Forrester (with 4 instead of 5 state variables and with linear approximations of the sectionally linear table functions in Forrester's model). The results of this experimental program as well as the results of the application of the better algorithms to the complete controlled World 2 model are presented in this paper.

The outline of the paper is as follows: In Chapter 2 a precise statement is given of the complete controlled World 2 model and of the test problem, the simplified controlled World 2 model. In Chapter 3 an outline is given of the different gradient algorithms considered in the experimental program together with a discussion of the two different techniques tried out to take into account the bounds on the values of the control variables. Also in this chapter some remarks are made on the scaling of the variables. In Chapter 4 the numerical results for the different applications of the algorithms are presented and discussed. A short summary of the conclusions, an acknowledgement, a list of references, 5 tables and 4 figures conclude the paper.

2. *THE CONTROLLED WORLD 2 MODEL*

2.1 *The World 2 model of Forrester*

The World 2 model which Forrester developed for the Club of Rome and which formed the basis of the results in his book "World Dynamics" (Forrester (1971)) consists of a set of 5 interacting nonlinear difference equations which describe the evolution of 5 "level" or state variables:

P : Population
CI : Capital Investment
CIAF : Capital Investment in Agriculture Fraction
POL : Pollution
NR : Natural Resources

Differential equations in a notation more common to control engineers and equivalent to the difference equations of Forrester were given in Cuypers (1973)

$$
\begin{aligned}
\dot{P} &= 0.04.P.F_3(MSL).F_{16}(CR).F_{17}(FR).F_{18}(POLR) \\
&\quad -0.028.P.F_{11}(MSL).F_{12}(POLR).F_{13}(FR).F_{14}(CR) \\
\dot{CI} &= -0.025.CI + 0.05.P.F_{26}(MSL) \\
\dot{CIAF} &= -(CIAF - F_{36}(FR).F_{43}[F_{38}(MSL)/F_{40}(FR)])/15 \\
\dot{POL} &= - POL/F_{34}(POLR) + P.F_{32}(CIR) \\
\dot{NR} &= - P.F_{42}(MSL)
\end{aligned}
\qquad (2.1)
$$

The functions $F_k(\cdot)$ in these equations are *coupling functions* given by Forrester as sectionally linear functions of their arguments. (The index k corresponds to the number of the section in Chapter 3 of Forrester (1971) in which the corresponding coupling function is presented). The arguments of these functions are, respectively, the normalized variables:

CR = P/PS (PS = Population Standard = $3.5775.10^9$)
CIR = CI/P
POLR = POL/POLS (POLS = Pollution Standard = $3.6.10^9$)
NRFR = NR/NRI (NRI = Natural Resources Initial = 9.10^{11})

and the auxiliary variables MSL (= Material Standard of Living) and FR (= Food Ratio) defined as

$$MSL = (CI/P)((1-CIAF)/(1-CIAFN)).F_6(NRFR) \tag{2.2}$$

and

$$FR = F_{20}(CR).F_{21}(CIRA).F_{28}(POLR) \tag{2.3}$$

where

$$CIRA = (CI/P)(CIAF/CIAFN) \qquad (CIAFN = CIAF\ Normal = 0.3)$$

Initial conditions for the differential equations (2.1) were specified by Forrester for the year 1900. Integration of the differential equations up to the year 1970 yields the following initial conditions for the year 1970 (cf. Cuypers (1973)).

$$\begin{aligned} P(1970) &= 3.67830938.10^9 & POL(1970) &= 2.88957159.10^9 \\ CI(1970) &= 3.83097633.10^9 & NR(1970) &= 7.7680742.10^{11} \\ CIAF(1970) &= 0.28031694 \end{aligned} \tag{2.4}$$

2.2 *The complete controlled World 2 model*

The most natural way to introduce regulating or control variables into this model (cf. Burns & Malone (1974)) is to assume that the magnitude of some of the coefficients in the differential equations (2.1) can be manipulated within certain bounds. The basis of the introduction of control variables into the World 2 model in case of the "Global Dynamics" project was the assumption that fractions U_P, U_{CI}, U_{POL} and U_{NR} of the total amount of goods and services not designated for agriculture, which amount was defined as

$$\begin{aligned} ISO &= CI.(1-CIAF).F_6(NRFR).U_r \\ &= P.MSL.(1-CIAFN).U_r \end{aligned} \tag{2.5}$$

(where ISO stands for Industrial and Service Output and where U_r is an efficiency factor (= the reciprocal of the capital coefficient with the standard value $U_r = 1/3$), can be allocated for respectively i) birthcontrol, ii) reinvestment, iii) pollution control and iv) protection of the natural resources. In addition, it was assumed that for the items i), iii) and iv) a law of diminishing returns would apply. Thus, the following *control multipliers* were postulated.

$$\begin{aligned} G_1(U_P) &= \exp\ (-\gamma_1.U_P.MSL) \\ G_3(U_{POL}) &= \exp\ (-\gamma_3.U_{POL}.(MSL/F_{32}(CIR))) \\ G_4(U_{NR}) &= \exp\ (-\gamma_4.U_{NR}) \end{aligned} \tag{2.6}$$

where γ_1, γ_3 and γ_4 are constants with the standard values

$$\gamma_1 = 25 \qquad \gamma_3 = 10 \qquad \gamma_4 = 3.5 \tag{2.7}$$

The assumed possibility to control the fraction of the ISO for reinvestment was realized by replacing the second differential equation of (2.1) by

$$\begin{aligned} \dot{CI} &= -0.025.CI + ISO.U_{CI} \\ &= -0.025.CI + P.MSL.(1-CIAFN).U_r.U_{CI} \end{aligned} \tag{2.8}$$

Given the standard values CIAFN = 0.3 and $U_r = 1/3$, the state equations of the controlled World 2 model become

$$\begin{aligned} \dot{P} &= 0.04.P.F_3(MSL).F_{16}(CR).F_{17}(FR).F_{18}(POLR).\exp(-\gamma_1 U_P.MSL) \\ &\quad - 0.028\ P.F_{11}(MSL).F_{12}(POLR).F_{13}(FR).F_{14}(CR) \\ \dot{CI} &= -0.025.CI + (0.7/3).P.MSL.U_{CI} \\ \dot{CIAF} &= -(CIAF - F_{36}(FR).F_{43}[F_{38}(MSL)/F_{40}(FR)])/15 \\ \dot{POL} &= -POL/F_{34}(POLR) + P.F_{32}(CIR).\exp(-\gamma_3 U_{POL}(MSL/F_{32}(CIR))) \\ \dot{NR} &= -P.F_{42}(MSL).\exp(-\gamma_4 U_{NR}) \end{aligned} \tag{2.9}$$

As part of the numerical investigations of the "Global Dynamics" project polynomial approximations were determined of the coupling functions $F_k(\cdot)$ which could replace the sectionally linear functions of Forrester in the ranges of interest for the optimization. The coefficients of these polynomials are given in *Table 2.1.*

Given the meaning of the control variables the following control constraints are self evident

$$U_P \geq 0 \qquad U_{CI} \geq 0 \qquad U_{POL} \geq 0 \qquad U_{NR} \geq 0 \tag{2.10}$$

and

$$U_P + U_{CI} + U_{POL} + U_{NR} \leq 1 \tag{2.11}$$

In addition, in order to prevent the optimization procedures to generate unrealistic values, the only control variable appearing linearly in the differential equation was given a simple upper and lower limit

$$0.198 \leq U_{CI} \leq 0.242 \tag{2.12}$$

To measure the quality of different controls a performance criterion should be defined. In case of the "Global Dynamics" project several criteria were considered of which the following, Bolza-type criterion became the standard one

$$J[u] = \int_{1970}^{2100} QL(\tau)P(\tau)d\tau + \lambda_P.P(2100) + \lambda_{POL}.POL(2100) + \lambda_{NR}.NR(2100) \tag{2.13}$$

In this expression the symbol QL (= Quality of Life) stands for almost the same performance measure as introduced by Forrester

$$QL = F_{38}(CMSL).F_{39}(CR).F_{40}(FR).F_{41}(POLR), \tag{2.14}$$

the difference being that the argument of the coupling function $F_{38}(\cdot)$ is not MSL but CMSL (= Consumption Material Standard of Living) which was defined by

$$CMSL = MSL.(1-U_P-U_{CI}-U_{POL}-U_{NR})/0.7828 \tag{2.15}$$

The constants λ_P, λ_{POL} and λ_{NR} in (2.13) were given the standard values

$$\lambda_P = 10 \quad \lambda_{POL} = -0.5P(1970)/POLS \quad \lambda_{NR} = 100P(1970)/NR(1970) \tag{2.16}$$

The optimal control problem thus derived, which will be called the *complete controlled World 2 model* to distinguish it from the simplified controlled World 2 model to be discussed in the next section, can now be summarized as follows:

"Given the state equations (2.9) with the initial conditions (2.4), find the control variables U_P, U_{CI}, U_{POL} and U_{NR} as functions of the time which satisfy the control constraints (2.10), (2.11) and (2.12) and which maximize (or minimize the negative of) the performance criterion (2.13)".

2.3 The simplified controlled World 2 model

The presence in the state equations (2.9) of the coupling functions, the values of which are to be determined by interpolation or polynomial approximation, considerably increase the computer time required for integration. For that reason, it was decided in an early phase of the numerical optimization experiments to make use of a simpler model which should have roughly the same characteristics as the original model but would be much easier to integrate. This object was realized by first linearizing all coupling functions around the standard uncontrolled trajectory and thereafter simplifying the complex of linear coupling functions in such a way, that in the uncontrolled case the results of Forrester were reasonably reproduced. Following this approach it was found that the state variable CIAF, which stayed fairly constant under standard conditions, could be replaced by a constant. Thus, the number of state equations was reduced from 5 to 4. Similarly, a number of coupling functions could be omitted as their values under standard conditions hardly differed from 1.0. This led to the following simple state equations

$$\begin{aligned}
\dot{P} &= 0.04.P.f_1(POL).f_2(CMSL).\exp(-25U_P.MSL)-0.028.P.f_3(POL).f_4(CMSL)\\
\dot{CI} &= -0.025\ CI + P.MSL.U_{CI}\\
\dot{POL} &= -POL/f_7(POL) + P.f_6(CI/P).\exp(-10U_{POL})\\
\dot{NR} &= -P.MSL.\exp(-3.5U_{NR})
\end{aligned} \tag{2.17}$$

where

$$\begin{aligned}
f_1(POL) &= 1.015 - 0.015\ POL & f_4(CMSL) &= 2.6 - 1.6\ CMSL\\
f_2(CMSL) &= 1.15 - 0.15\ CMSL & f_6(CI/P) &= -1.0 + 2(CI/P)\\
f_3(POL) &= 0.95 + 0.05\ POL & f_7(POL) &= 0.8333 + 0.1667.POL
\end{aligned} \tag{2.18}$$

and

$$MSL = (CI/P)(NR/NR(1970)) \tag{2.19}$$

and

$$CMSL = MSL.(0.7 - U_P - U_{CI} - U_{POL} - U_{NR})/0.7 \tag{2.20}$$

The corresponding initial conditions became

$$P(1970) = 1.0 \quad CI(1970) = 1.0 \quad POL(1970) = 1.0 \quad NR(1970) = 800/3.6 \tag{2.21}$$

and the control constraints

$$U_P \geq 0 \qquad U_{POL} \geq 0 \qquad U_{NR} \geq 0 \tag{2.22}$$

$$0.04027 \leq U_{CI} \leq 0.05527 \tag{2.23}$$

and

$$U_P + U_{CI} + U_{POL} + U_{NR} \leq 0.7 \tag{2.24}$$

As performance criterion was chosen

$$J[u] = \int_{1970}^{2100} QL(\tau)P(\tau)d\tau + 5.P(2100) - 0.05.POL(2100) + 0.4NR(2100) \tag{2.25}$$

where QL was defined as

$$QL = (0.8+0.2CMSL)(1.5-0.5P)(1.02-0.02P) \tag{2.26}$$

Thus, in summary, the following optimal control problem, to be called *the simplified controlled World 2 model* resulted

"Given the state equations (2.17) and the initial conditions (2.21), find the control variables U_P, U_{CI}, U_{POL} and U_{NR} as functions of time which satisfy the control constraints (2.22) - (2.24) and which maximize (or minimize the negative of) the performance criterion (2.25)".

It should be noted that although the standard (uncontrolled) behavior of this simplified model compared quite well with the results of Forrester, the optimal behavior turned out to be quite different from the optimal behavior of the complete controlled World 2 model. One of the main reasons for this was the coupling function $f_4(MSL)$, which for values of MSL larger than 1.625 have unrealistic negative values. This turned out to have a large influence on the optimal behavior. After the discovery of the imperfection the use of the model was continued for reason of its good properties as a test problem.

3. OUTLINE OF THE ALGORITHMS TESTED

3.1 Gradient algorithms for solving optimal control problems

Both optimal control problems specified in the preceding sections were of the following basic form:

"Given the state equations

$$\dot{x} = f(x,u) \qquad x:[t_b,t_f] \rightarrow R^n, \; u:[t_b,t_f] \rightarrow R^m \tag{3.1}$$

and the initial conditions

$$x(t_b) = x_b \tag{3.2}$$

find the control vector $u(t), t\varepsilon[t_b,t_f]$ which satisfies the constraints

$$u_{i,min} \leq u_i(t) \leq u_{i,max} \qquad t\varepsilon[t_b,t_f] \ , \ i=1,\ldots,m \tag{3.3}$$

and which generates the least value of the performance criterion

$$J[u] = k(x(t_f)) + \int_{t_b}^{t_f} \ell(x(\tau),u(\tau))d\tau \quad " \tag{3.4}$$

From a computational point of view this type of optimal control problem is rather simple: The initial and final times are fixed and there are no terminal constraints. Except for the presence of the constraints on the values of the control variables, a problem which will be dealt with below in a special section, this control problem formulation is well suited for the gradient type of algorithms, as will be seen.

Gradient methods for solving optimal control problems are iterative methods in which the control vector function is modified in each iteration so as to improve the performance criterion. Most of the algorithms contain the following basic steps

(o) assume $u^{(o)}(t), t\varepsilon[t_b,t_f]$, given and set i: = 0;

(i) evaluate the performance criterion $J[u^{(i)}]$ corresponding to $u^{(i)}$ (by integrating the state equations (3.1) forward) and the gradient $\nabla_u J^{(i)}(t), t\varepsilon[t_b,t_f]$ as to be discussed below (i.e. by integrating the costate equations (3.7) backward);

(ii) test: if $u^{(i)}$ optimal, stop; otherwise:

(iii) determine a new search direction $d^{(i)}(t), t\varepsilon[t_b,t_f]$;

(iv) set $u(t)$: = $u^{(i)}(t)+\alpha d^{(i)}(t)$ and determine the scalar value $\alpha^{(i)}$ of α for which the performance criterion considered as a function of α reaches its minimum value (or in some algorithms: reaches a lower value which satisfies certain specifications)

(v) set $u^{(i+1)}(t)$: = $u^{(i)}(t)+\alpha^{(i)}d^{(i)}(t)$, set i: = i+1 and return to step (i).

The step in this algorithm by which the different algorithms are distinguished is step (iii). Over the years a great number of search directions have been proposed, most of which, however, have in common that they make use of the gradient (with respect to the control) of the performance criterion (considered as a functional of the control only). This gradient is, as is well known (cf. Bryson & Ho (1969)), at each time instant equal to

$$\nabla_u J^{(i)}(t) = H_u^{(i)T}(t) = (\ell_u^T+f_u^T\lambda)^{(i)}(t) \tag{3.5}$$

where H_u is the partial derivative with respect to the control of the *Hamiltonian*, which is defined as:

$$H(x,u,\lambda) = \ell(x,u)+\lambda^T f(x,u) \tag{3.6}$$

and where $\lambda(t), t\varepsilon[t_b,t_f]$ is the *costate* or *adjoint vector* which is the solution of the costate or adjoint equation

$$\dot{\lambda} = -f_x^T\lambda - \ell_x^T \tag{3.7}$$

with the "initial" condition

$$\lambda(t_f) = k_x^T(x(t_f)) \tag{3.8}$$

The gradient $\nabla_u J^{(i)}(t)$ corresponding to a particular $u^{(i)}$ can be computed by one backward integration of the costate equations (corresponding to that $u^{(i)}$).

3.2 Methods tested

Most gradient methods in use for solving optimal control problems may be considered the infinite dimensional equivalents of the better known gradient methods for solving unconstrained finite dimensional minimization problems. The methods actually tested in the numerical experiments to be described were the infinite dimensional equivalents of the following finite dimensional methods (cf. Murray (1972), Jacoby, Kowalik & Pizzo (1972)):

a) SD(= Steepest Descent) method
b) PARTAN (= Parallel Tangents) method
c) CGI (= Conjugate Gradient I) method (of Fletcher-Reeves)
d-e) CGII (= Conjugate Gradient II) method (of Hestenes-Stiefel)
f) DFP (= Davidon-Fletcher-Powell) method

Given the definitions of the infinite dimensional inner product and the corresponding norm (in $\mathcal{L}_2^m[t_b,t_f]$)

$$\langle g^{(i)},h^{(i)}\rangle = \int_{t_b}^{t_f} g^{(i)T}(\tau)h^{(i)}(\tau)d\tau \qquad ||v|| = \langle v,v\rangle^{\frac{1}{2}} \tag{3.9}$$

the search directions of the infinite dimensional counterparts of the methods a) - e) are, respectively, given by

a') SD-method (cf. Kelley (1962) Bryson & Denham (1962)):

$$d^{(i)}(t) := -\nabla_u J^{(i)}(t) \tag{3.10}$$

b') PARTAN-method (cf. Wong, Dressler & Luenberger (1971)):

$$d^{(2i)}(t) := -\nabla_u J^{(2i)}(t) \qquad i = 0,1,2,\ldots \tag{3.11}$$

$$d^{(2i+1)}(t) := \begin{cases} ||\nabla_u J^{(2i)}|| \dfrac{(u^{(2i+1)}(t)-u^{2(i-1)}(t))}{||u^{(2i+1)}-u^{2(i-1)}||} & i = 1,2,\ldots \\ 0 & i = 0 \end{cases} \tag{3.12}$$

c') CGI-method (cf. Lasdon, Mitter & Waren (1967)):

$$d^{(i)}(t) = -\nabla_u J^{(i)}(t)+\beta^{(i)} d^{(i-1)}(t) \tag{3.13}$$

where

$$\beta^{(i)} = \frac{<\nabla_u J^{(i)},\nabla_u J^{(i)}>}{<\nabla_u J^{(i-1)},\nabla_u J^{(i-1)}>} \tag{3.14}$$

d') CGIIA-method (cf. Pagurek & Woodside (1968))

$$d^{(i)}(t) := -\nabla_u J^{(i)}(t)+\beta^{(i)} d^{(i-1)}(t) \tag{3.15}$$

with

$$\beta^{(i)} := \frac{<\nabla_u J^{(i)},v^{(i)}>}{<d^{(i-1)},v^{(i)}>} \tag{3.16}$$

where $v^{(i)}(t)$ (which is the infinite dimensional equivalent of the matrix-vector product $G^{(i)}d^{(i-1)}$ where $G^{(i)}$ is the local Hessian), can be determined from

$$v^{(i)}(t) := f_u{}^T w^{(i)}(t)+H_{ux} z^{(i)}(t)+H_{uu} d^{(i-1)}(t) \tag{3.17}$$

where $z^{(i)}(t)$ is the solution of

$$\dot{z}^{(i)} = f_x z^{(i)}(t)+f_u d^{(i-1)}(t) \qquad z^{(i)}(t_b) = 0 \tag{3.18}$$

and $w^{(i)}(t)$ is the solution of

$$\dot{w}^{(i)} = -f_x{}^T w^{(i)}-H_{xx} z^{(i)}-H_{xu} d^{(i-1)} \qquad w^{(i)}(t_f) = k_{xx} z^{(i)}(t_f) \tag{3.19}$$

e') CGIIB-method (cf. Sinnott & Luenberger (1967)):
As CGII-A-method with the replacement of H_{ux}, H_{uu} in (3.17) and H_{xx} and H_{xu} in (3.19) by respectively ℓ_{ux}, ℓ_{uu}, ℓ_{xx} and ℓ_{xu}

f') DFP-method (cf. Tripathi & Narendra (1968)):

$$d^{(i)}(t) = -\nabla_u J^{(i)}(t) - \sum_{k=0}^{i-1} \frac{<s^{(k)},\nabla_u J^{(i)}>}{<s^{(k)},y^{(k)}>} s^{(k)}(t) + \sum_{k=0}^{i-1} \frac{<a^{(k)},\nabla_u J^{(i)}>}{<a^{(k)},y^{(k)}>} a^{(k)}(t) \tag{3.20}$$

where

$$a^{(k)}(t) := y^{(k)}(t) + \sum_{j=0}^{k-1} \frac{<s^{(j)},y^{(k)}>}{<s^{(j)},y^{(j)}>} s^{(j)}(t) - \sum_{j=0}^{k-1} \frac{<a^{(j)},y^{(k)}>}{<a^{(j)},y^{(i)}>} a^{(j)}(t) \tag{3.21}$$

with

$$s^{(j)}(t) := u^{(j+1)}(t)-u^{(j)}(t) \qquad , \qquad y^{(j)}(t) := \nabla_u J^{(j+1)}(t)-\nabla_u J^{(j)}(t) \tag{3.22}$$

In the process of executing this DFP-algorithm, it is required that in each iteration two new vector functions, $s^{(i)}(t)$ and $a^{(i)}(t)$, are stored. This implies that the

required computer memory increases with the number of iterations. To cure this, it is customary to restart the algorithm periodically after a fixed number of iterations.

It may be noticed that in both methods, the CGIIA method and the CGIIB method, one extra forward integration (of (3.18)) and one extra backward integration (of (3.19)) are required to evaluate $\beta^{(i)}$. The CGIIB method has as advantage over the CGIIA method that no second order partial derivatives of the state equations are required which implies less programming effort and less computing time for integration.

3.3 Techniques for bounded controls

a) Clipping-off-technique

The first technique which was used for taking care of bounds on the values of the control components is known as the clipping-off-technique (cf. Quintana & Davison (1974)) and amounts to setting the control components back at their bounds as soon as these are violated in the search for a line minimum. This implies the following modification in step (iv) of the standard algorithm: Evaluate

$$u_{j,\text{unclipped}}(t) := \bar{u}_j^{(i)}(t)+\alpha d^{(i)}(t) \tag{3.23}$$

and set

$$\begin{aligned} \bar{u}_j(t) &:= u_{j,\max} && \text{if} && u_{j,\text{uncl}}(t) \geq u_{j,\max} \\ &:= u_{j,\text{uncl}}(t) && \text{if} && u_{j,\min} \leq u_{j,\text{uncl}}(t) \leq u_{j,\max} \\ &:= u_{j,\min} && \text{if} && u_{j,\text{uncl}}(t) \leq u_{j,\min} \end{aligned} \tag{3.24}$$

In case of no bounds on the values of the control components, the gradient tends to zero when the minimum is approached. Most gradient algorithms make implicitly use of this fact. When the minimum is attained at the boundary of the feasible region, the corresponding gradient (component) does not become small. This may spoil the search direction calculations. For instance, without modification, the values of the inner products in $\beta^{(i)}$ in (3.14) would almost completely be determined by the large gradient components corresponding to the control components at their bounds, and $\beta^{(i)}$ erroneously would get the value of approximately 1.0 in all iterations. In order to cure that situation the algorithms a') - f') were modified with the aid of *clipped functions* which are defined as

$$\begin{aligned} \bar{q}_j^{(i)}(t) &:= 0 && \text{if} && \bar{u}_j^{(i)}(t) \textit{ and } \bar{u}_j^{(i-1)}(t) \text{ at boundary} \\ &:= q_j^{(i)}(t) && && \text{otherwise} \end{aligned} \tag{3.25}$$

With this definition the modified search directions may be written as:

a') method of steepest descent

no change

b') PARTAN-method

$$d^{(2i)}(t) := -\nabla_u J^{(2i)}(t) \qquad i = 0,1,2,\ldots$$

$$d^{(2i+1)}(t) := \|\nabla_u J^{(2i)}(t)\| \frac{(\bar{u}^{(2i+1)}(t)-\bar{u}^{(2i-1)}(t))}{\|(\bar{u}^{(2i+1)}(t)-\bar{u}^{(2i-1)}(t)\|} \qquad i = 1,2,\ldots \quad (3.26)$$

$$= 0 \qquad i = 0$$

c') CGI-method

$$d^{(i)}(t) := -\nabla_u J^{(i)}(t) + \frac{<\overline{\nabla_u J}^{(i)}, \overline{\nabla_u J}^{(i)}>}{<\overline{\nabla_u J}^{(i-1)}, \overline{\nabla_u J}^{(i-1)}>} d^{(i-1)}(t) \quad (3.27)$$

d'-e') CGII-methods

$$d^{(i)}(t) := -\nabla_u J^{(i)}(t) + \frac{<\overline{\nabla_u J}^{(i)}, \tilde{v}^{(i)}>}{<s^{(i-1)}, \tilde{v}^{(i)}>} s^{(i-1)}(t) \quad (3.28)$$

where

$$\tilde{v}^{(i)}(t) := f_u^T \tilde{w}^{(i)}(t) + H_{ux}^{(\ell)} \tilde{z}^{(i)}(t) + H_{uu}^{(\ell)} s^{(i-1)}(t) \quad (3.29)$$

with $\tilde{z}^{(i)}(t)$ satisfying

$$\dot{\tilde{z}}^{(i)} = f_x \tilde{z}^{(i)} + f_u s^{(i-1)} \qquad \tilde{z}^{(i)}(t_b) = 0 \quad (3.30)$$

and $\tilde{w}^{(i)}(t)$ satisfying

$$\dot{\tilde{w}}^{(i)} = -f_x^T \tilde{w}^{(i)} - H_{xx}^{(\ell)} \tilde{z}^{(i)} - H_{xu}^{(\ell)} s^{(i-1)} \qquad \tilde{w}^{(i)}(t_f) = k_{xx} \tilde{z}^{(i)}(t_f) \quad (3.31)$$

f') DFP-method

$$d^{(i)}(t) := -\nabla_u J^{(i)}(t) - \sum_{k=0}^{i-1} \frac{<s^{(k)}, \overline{\nabla_u J}^{(i)}>}{<s^{(k)}, \bar{y}^{(k)}>} s^{(k)}(t) + \sum_{k=0}^{i-1} \frac{\bar{a}^{(k)}, \overline{\nabla_u J}^{(i)}}{\bar{a}^{(k)}, \bar{y}^{(k)}} \bar{a}^{(k)}(t) \quad (3.32)$$

with

$$a^{(k)}(t) := y^{(k)}(t) + \sum_{j=0}^{k-1} \frac{<s^{(j)}, \bar{y}^{(k)}>}{<s^{(j)}, \bar{y}^{(j)}>} s^{(j)}(t) - \sum_{j=0}^{k-1} \frac{<\bar{a}^{(j)}, \bar{y}^{(k)}>}{<\bar{a}^{(j)}, \bar{y}^{(j)}>} \bar{a}^{(j)}(t) \quad (3.33)$$

It may be remarked that, in line with the replacement of $d^{(i-1)}(t)$ by $s^{(i-1)}(t)$ in the formulae (3.28)-(3.31) of the CGII-methods, the replacement of $d^{(i-1)}(t)$ by $(\alpha^{(i-1)})^{-1} s^{(i-1)}(t)$ in the CGI-method would have been logical (and conform the essence of one of the suggestions of Quintana and Davison (1974)). However, numerical experiments with this alternative showed that the convergence behavior was worse with replacement of $d^{(i-1)}(t)$ than without. The numerical evidence of this will be presented in the next chapter.

b) Transformation technique

The second well-known technique (cf. Jacoby, Kowalik & Pizzo (1972)) for taking care of bounded controls in gradient algorithm is the transformation technique. This technique consists of replacing the original control variables by new variables by means of a transformation which guarantees that the bounds on the original variables are automatically satisfied while the new variables are unconstrained. In particular, in case of a lower bound only, e.g. $u_j(t) \geq 0$, a common transformation is

$$u_j(t) = k_j v_j^2(t) \tag{3.34}$$

and similarly, in case of a lower and an upper bound, e.g. $a \leq u_j(t) \leq b$, a common transformation is

$$u_j(t) = \tfrac{1}{2}(a+b) - \tfrac{1}{2}(b-a)\cos(\pi k_j v_j(t)) \tag{3.35}$$

in which expressions the k_j's are arbitrary scale factors. The transformations in these cases have the property that whenever a control component approaches its bound in the original system, the corresponding gradient component with respect to the new variables tends to zero.

Against the advantage of having unconstrained instead of constrained variables, the transformation technique was found to have three smaller disadvantages for application in connection with control problems:

i) whenever a control component is at its boundary on a particular time interval at some instant during the iteration process, then there is no way when using gradient methods to leave that boundary. This property eliminates in particular a number of otherwise useful startsolutions

ii) the transformation "distorts" the object function (3.4) very severely in the neighborhood of the bounds which impairs the rate of convergence whenever the optimum happens to be near or partly on the boundary.

iii) the transformation implies an extra programming effort, which, especially in case of the CGII methods, is considerable.

One aspect of the minimization procedure which became clear when using the transformation technique was the importance of good scaling for the convergence behavior. This will be discussed in more detail in the next section.

3.4 Scaling

The convergence behavior of gradient algorithms depends, as is well known, very much on the scaling of the variables relative to the function to be minimized. This phenomenon may be explained with the observation that in gradient algorithms steps are taken which are more or less proportional to the gradient. Whenever a certain gradient

vector component is large relative to the other components, which means that the object function is very sensitive to changes in the corresponding variable, then a step proportional to the gradient implies a large change in that particular variable, while the opposite would be desirable. The idea behind scaling is therefore to try to make all gradient components of the same order of magnitude, or equivalently, to make the object function equally sensitive to changes in all the variables.

In the simplified controlled World 2 model the original control variables turned out to be reasonably well scaled and no effort was put in to obtain a better scaling. As soon as the transformed variables v(t) (3.34)-(3.35) were introduced instead of the original control variables u(t), the need for scaling became more apparent: The gradient components relative to the new variables become

$$(\nabla_v J(t))_j = (\nabla_u J(t))_j . 2k_j v_j \qquad j = 1,3,4 \qquad (3.36)$$

and

$$(\nabla_v J(t))_2 = (\nabla_u J(t))_2 . \tfrac{1}{2}(b-a)\pi k_2 \sin(\pi k_2 v_2(t)) \qquad j = 2 \qquad (3.37)$$

Given the situation that the original gradient vector components $(\nabla_u J(t))_j$ are of roughly the same size, the new gradient vector will also be of the same size if

$$k_2/k_j \simeq \frac{2v_j}{\frac{1}{2}(b-a)\pi} \qquad (3.38)$$

for the simplified controlled World 2 model, where $v_j \simeq 0.1$ and $(b-a) = 0.015$ a reasonable scaling was obtained with the scale factor values

$$k_1 = k_3 = k_4 = 1 \qquad k_2 = 10 \qquad (3.39)$$

In the complete controlled World 2 model the gradient components were no longer of the same order of magnitude. In particular, the gradient component corresponding to the population control variable U_P turned out to become much larger than the other components. A closer look at the control multipliers (2.6) explained this: With MSL $\simeq$ 12 and $F_{32}(CI/P) \simeq 8$ in the neighborhood of the optimal solution, these control multipliers became

$$\begin{aligned} G_1(U_P) &= \exp(-\gamma_1 U_P \mathrm{MSL}) \simeq \exp(-300\ U_P) \\ G_2(U_{CI}) &= U_{CI} \\ G_3(U_{POL}) &= \exp(-\gamma_3 U_{POL}(\mathrm{MSL}/F_{32})) \simeq \exp(-15\ U_{POL}) \\ G_4(U_{NR}) &= \exp(-\gamma_4 U_{NR}) = \exp(-3.5\ U_{NR}) \end{aligned} \qquad (3.40)$$

An obvious way to scale the control variables in this particular case was to reformulate the optimal control problem with as new control variables

$$\tilde{u}_1 = 25\ U_P^{MSL} \qquad \tilde{u}_3 = 10\ U_{POL}^{MSL}/F_{32}(CI/P)$$
$$\tilde{u}_2 = -\ln(U_{CI}) \qquad \tilde{u}_4 = 3.5\ U_{NR} \tag{3.41}$$

This approach, which will be called the *reformulation technique*, used in conjunction with the clipping-off technique to take into account the translated bounds on the $\tilde{u}$-variables, turned out to improve the convergence of the application of the gradient algorithms considerably. Numerical evidence of this will be discussed in the next chapter.

4. *NUMERICAL RESULTS*

4.1 *Optimization results*

The optimal control histories and the corresponding optimal state space trajectories are given in Fig.4.1 for the simplified model and in Fig.4.2 for the complete model.The optimal state space trajectories can be compared with the trajectories in case of no control (i.e. Forrester's standard results) which are presented by dotted curves in the same figures. A discussion of these results falls outside the scope of this paper: for this the reader is referred to Rademaker (1972). One remark should be made, however, and that is, that a comparison of the optimal control and state space trajectories for the two different models shows that at most only the tendencies in the behaviors roughly compare. The actual results are quite different. In fact, the optimal criterion values of the simplified model satisfies

$$J[u^*] > 178.911 \tag{4.1}$$

whereas for the complete model

$$J[u^*] > 500.042 \tag{4.2}$$

For the larger part this difference between the results for the two models can be attributed to the difference in coupling functions. In the case of the complete model much larger values of the CMSL (2.17), and through the CMSL much larger values of the QL(2.16), are generated than in the case of the simplified model. This underlines the fact that the models are indeed quite different.

4.2 *Comparison of the application of different methods to the simplified controlled World 2 problem*

In order to compare their relative efficiency all methods to be applied on the simplified model were programmed as special subroutines within one general computer program for solving optimal control problems. Two versions of this general program were used, one of which made use of the clipping-off technique for taking into account the bounds on the values of the control variables, the other one making use of the transformation technique. The aim of this approach was to obtain a comparison of the methods which should be independent of the particular way of programming of the algorithm. The drawback of such an approach was of course the fact that none of the methods was pro-

grammed in an optimally efficient way.

In the general program the *integration* of the differential equations was carried out by a standard fourth order Runge-Kutta routine. After some experimentation a stepsize of 2 years was found to be the best compromise between accuracy and required computer time. For the *line search* use was made of a quadratic search routine in which first three points on the line are determined which include the line minimum. For the initial stepsize in this search routine, which influences of course the number of function calls, two strategies were tried out, the first one consisting of using in every new line search the same small initial stepsize ($\alpha_{start} = 0.001$ in the clipping-off-version and $\alpha_{start} = 0.01$ in the transformation-version of the general program), the second one consisting of using an initial stepsize which was equal to half the optimal stepsize $\alpha^{(i-1)}$ in the preceding iteration. The result of this experiment is given in Table 4.6 which will be discussed in more detail below. As *convergence criterion* for terminating the iterative process use was made of the criterion that in two successive steps the performance criterion should not change in absolute value more than $\varepsilon_{conv} = 0.0001$. Whenever this criterion is satisfied one extra line minimization is performed with as search direction the negative of the local gradient. Only in the case that the convergence criterion is satisfied again the iterative process is terminated, otherwise the process is continued.

The results of the application of the different methods to the *simplified controlled World 2 model* are given in the Tables 4.3 to 4.6, in which the number of iterations, the number of function (= performance criterion) evaluations, the value of the performance criterion, the total computer time (on a Burroughs B 6700 multiprocessing system), the average number of calls per interation and the average amount of computer time per call are listed. The computer times given should not be taken as hard figures but only as an indication for the relative performance. The computer used being a multiprocessing machine, the actual process time may differ from case to case up to 30% depending on what other programs are processed simultaneously.

The numbers in the individual tables apply to iteration processes with the following initial controls:

In case of Table 4.3 and 4.5

$$u_1^{(o)}(t) \equiv 0 \qquad u_2^{(o)}(t) \equiv 0.04777 \qquad u_3(t) \equiv 0.05 \qquad u_4(t) \equiv 0 \tag{4.3}$$

and in case of Table 4.4 and 4.6

$$v_1^{(o)}(t) \equiv 0.1 \qquad v_2^{(o)}(t) \equiv 0.05 \qquad v_3^{(o)}(t) \equiv 0.1 \qquad v_4^{(o)}(t) \equiv 0.1 \tag{4.4}$$

which, with the actual transformations used

$$\begin{aligned} u_1(t) &= v_1^2(t) \qquad u_2(t) = 0.04777 - 0.0075 \cos(\pi.10.v_2(t)) \\ u_3(t) &= v_3^2(t) \qquad u_4(t) = v_4^2(t) \end{aligned} \tag{4.5}$$

are equivalent to initial controls in terms of u equal to

$$u_1^{(o)}(t) \equiv 0.01 \qquad u_2^{(o)}(t) \equiv 0.04777 \qquad u_3(t) \equiv 0.01 \qquad u_4(t) \equiv 0.01 \qquad (4.6)$$

Table 4.3 shows the results of the tests with the different methods in combination with the use of the clipping-off technique. As known in the literature (cf. Pierson & Rajtora (1970)) it is advantageous to periodically restart the iteration process. To determine the best number after which to restart as well as to get more data on the same method all methods were tried with periodic restarts after respectively 6, 12 and 18 iterations (In the PARTAN method application periodic restarts were made after respectively 6, 12 and 18 PARTAN directions of search, i.e. after respectively 13, 25 and 37 line searches following (3.26)). From the results listed in the table it is immediately clear that the most efficient method in terms of number of iterations, number of function evaluations as well as computer time is the CGI method. The second best method in terms of number of iterations is the CGIIA method. Unfortunately, however, this method also requires the most computer time per iteration, which makes it into the most time consuming method. The third best method in number of iterations and at the same time the second best in terms of computer time is the DFP method, which makes this method a good second choice. Of interest in Table 4.3 is furthermore the relative poor performance of the CGIIB method in comparison with the CGIIA method mentioned above and the similarly poor performance of the PARTAN method in comparison even with the SD method. It should be remarked in this context that the number of iterations of the PARTAN method in the present case is defined as the number of search directions, a definition which is different from the one used by Wong, Dressler and Luenberger (1971). In addition to the results for the different methods of Section 3.2, Table 4.3 also lists the results for an experimental method, in which the search direction is calculated in the same way as in the CGI method (following (3.13)) but with a fixed value of $\beta^{(i)} = 1.0$. The results show clearly that such a simple-minded method is much inferior to the hardly more complicated CGI method and also inferior to the other methods of Section 3.2.

Table 4.4 shows results similar to Table 4.3 for the case that the transformation technique is used instead of the clipping-off technique. Again the CGI method is the most efficient method in terms of the amount of computer time. On the average the CGIIA method requires less iterations, however, with the highest amount of computer time per call, the method is at the same time one of the most time consuming methods.The second best method in terms of computer time is in this case the PARTAN method with the DFP-method being third. Again, the poorer performance of the CGIIB method relative to the CGIIA method in terms of number of iterations and number of function evaluations is evident.

In order to make a comparison possible of the application of the transformation technique versus the application of the clipping-off technique,Table 4.4 also lists the results for the CGI method with the clipping-off technique applied to a case with initial

controls (4.6) equivalent to the initial controls (4.5) used to generate the other results in the table. Comparison shows that the clipping-off technique requires less iterations, less function evaluations and less computer time. Also the clipping-off technique leads in general to higher values of the performance criterion than the transformation technique. From detailed results on the convergence behavior not given here,it appeared that the initial convergence using the transformation technique was faster than using the clipping-off technique, whilst the final convergence on the other hand was much slower. Reasons for this phenomenon may be on one hand the simplification of the optimization problem in case of the clipping-off technique caused by the elimination of all control variable components on their bounds and on the other hand the distortion of the equi-cost surfaces by the transformation from the u-variables to the v-variables.

Table 4.5 shows the results of some more experiments to determine the best reset or restart value for the two most efficient methods, the CGI method and the DFP method,both with the clipping-off technique. In addition results are presented for a modification of the CGI method (cf..Section 3.3), in which the previous search direction $d^{(i-1)}(t)$ in (3.27) is replaced by $s^{(i-1)}(t)/\alpha^{(i-1)}$. It follows that the best reset value for both versions of the CGI method is 18, whereas for the DFP method a reset value of 30 or higher is best. Both these reset values are higher than commonly suggested in the literature (cf. Pierson & Rajtora (1970, Keller & Sengupta (1973)). It also follows that the CGI method with $d^{(i-1)}(t)$ is superior to the same method with $s^{i-1)}(t)/\alpha^{i-1)}$ replacing $d^{(i-1)}(t)$. This result is of interest since it contradicts the suggestion of Quintana and Davison (1974). It may be remarked in this context that in the CGIIA method as well as in the CGIIB method the use of $s^{(i-1)}(t)$ instead of $d^{(i-1)}(t)$ as prescribed by the algorithm (3.28)-(3.31) turned out to be almost imperative: In a number of, though not all, tests with the CGIIA and CGIIB methods with $d^{(i-1)}(t)$ instead of $s^{(i-1)}(t)$, the iterative process did not converge at all.

Table 4.6 lists the results of some extra experiments with a different stepsize strategy in the line search procedure. In particular, for three cases listed in Table 4.4 and repeated here, i.e. the CGI method and the DFP method with the transformation technique and the CGI method with the clipping-off technique the results are presented which were generated while using as initial stepsize in the line search procedure $\alpha_{start} := 0.5\alpha^{(i-1)}$ instead of a constant fixed value. The table shows that while on the average the number of iterations does not differ too much, the total number of function evaluations as well as the average number of function evaluations per iteration are considerably less. The result clearly indicates the superiority of the strategy to let the initial stepsize α_{start} depend on the preceding optimal stepsize $\alpha^{(i-1)}$.Unfortunately, however there is one important proviso and that is that in no iteration such large steps are generated that computer overflow results. In fact, in a great number of trials this happened, for which reason the strategy was not used for the comparison runs presented in the preceding tables.

4.3 Some numerical experiments with the complete controlled World 2 model

After the numerical experiments described in the preceding section had indicated the superiority of the CGI algorithm for solving optimal control problems of the type of the controlled World 2 model, only a limited number of comparison runs (with the same initial controls and the same overall conditions) were tried out with the complete World 2 model. (The computer time for one function (= performance criterion) evaluation was roughly 2.5 times as long as in case of the simplified model). One set of comparison runs which was tried was concerned with four runs with respectively the SD method, the PARTAN method, the CGI method and the DFP method, all four in combination with the transformation technique, restarting the process after every 6 iterations. The convergence histories of these runs are presented (up to the 40th iteration) in *Figure 4.7*. From this figure it follows that the CGI method is again the fastest converging method followed by the PARTAN method, the SD method and the DFP method, which order is reasonably well in agreement with the results presented in Table 4.4. The dotted line segments in the figure show the convergence behavior of the PARTAN method for the case that the iteration definition of Wong, Dressler and Luenberger (1971) is followed. (One iteration is then defined to consist of one search along the negative gradient followed by one search along the PARTAN direction). It is of interest to note the little difference between the convergence histories of the CGI method and the thus defined PARTAN method (in which per iteration roughly twice as much work has to be done).

A second set of comparison runs which was tried was concerned with four runs with the CGI method, with restarts after every 6 iterations, in combination with four different strategies for taking care of the bounds on the values of the control variables:the use of the clipping-off technique, the use of the transformation technique, the use of a mixture of these techniques (first 15 iterations with the clipping-off technique, thereafter the transformation technique) and finally the use of the clipping-off technique after a reformulation or rescaling of the control variables as discussed in Section 3.4. The convergence histories of these runs are presented in *Figure 4.8*. It follows that the best convergence behavior is obtained through the use of rescaling or reformulation in combination with the clipping-off technique. The second best strategy is to alternate between the clipping-off technique and the transformation technique.The pure strategies, i.e. using the transformation technique of the clipping-off technique for all interations produced a less good convergence behavior.

5. CONCLUSIONS

Numerical experiments have been carried out with six different gradient methods for the determination of the optimal control of a simplified version of the controlled World 2 model of Forrester. The main conclusion of these experiments was that the most efficient method in terms of computer time and generally also in terms of number of iterations and number of function evaluations was the CGI method (i.e. the infinite

dimensional equivalent to the Conjugate Gradient method of Fletcher and Reeves, first suggested by Lasdon, Mitter & Waren (1967)) in combination with a clipping-off technique (as described by Pagurek and Woodside (1968)) to take care of bounds on the values of the control variable components and periodically restarted every 18 iterations. A good second choice proved to be the DFP method (i.e. the infinite dimensional equivalent of the Davidon-Fletcher-Powell method following the algorithm of Tripathi-Narendra (1968)) in combination with the clipping-off technique, which in general turned out to be a more efficient method to take care of bounded controls than the transformation of variables technique.

The results of numerical experiments with the determination of the optimal control of the complete controlled World 2 model of Forrester showed in general good agreement with the results obtained for the simplified model. Again the CGI method in combination with the clipping-off technique turned out to be the most efficient method when the problem first had been rescaled by means of a reformulation of the control variables. Scaling proved in this case to be one of the most important factors for convergence.

6. ACKNOWLEDGEMENTS

The authors should like to acknowledge the contributions of the former students J.G. van der Velden, E.J. Mendieta, J.H. Kessels, P.F.G. Vereijken and H. Paulissen to the research reported here. They are very much indebted to Mr. R. Kool of the Department of Mathematics for his numerous improvements and suggestions while actually transforming written-out algorithms into working computer programs.

7. REFERENCES

BRYSON, Jr, A.E. and HO, Y.C. (1969): Applied Optimal Control, Blaisdell Publ. Cy, Waltham, Mass.

BRYSON, Jr. A.E. and DENHAM, W.F. (1962): A Steepest Ascent Method for Solving Optimum Programming Problems. Trans ASME, J.Appl.Mech. 29, pp. 247-259.

BURNS, J.R. and MALONE, D.W. (1974): Optimization Techniques Applied to the Forrester Model of the World, IEEE Trans.Syst.Man.Cybern., SMC-4, pp. 164-172.

CUYPERS, J.G.M. (1973): Two simplified versions of Forrester's model. Automatica, 9, pp. 399-401.

DEKKER, L. and KERCKHOFFS, E.H.J. (1974): Hybrid simulation of a World model, AICA J. 16, nr.4 pp. 10-14.

FORRESTER, J.W. (1971): World Dynamics, Wright-Allen Press Inc., Cambridge, Mass.

JACOBY, S.L.S., KOWALIK, J.S. and PIZZO, J.T. (1972): Iterative methods for nonlinear optimization problems, Prentice Hall, Inc., Englewood Cliffs, N.J.

KELLER, Jr., E.A. and SENGUPTA, J.K. (1973): Relative efficiency of computing optimal growth by conjugate gradient and Davidon methods, Int.J. Systems Sci, 4, pp. 97-120.

KELLEY, J.H. (1962): Methods of Gradients, Ch. 6 of Optimization Techniques. G. Leitmann (Ed.), Academic Press New York, pp. 206-252.

LASDON, L.S., MITTER, S.K. and WAREN, A.D. (1967) The Conjugate Gradient Method for Optimal Control Problems,IEEE Trans. Aut. Contr., AC-12, pp. 132-138.

MEADOWS, D.H. et al. (1972): The Limits to Growth, Universe Books, New York.

MURRAY, W. (ed) (1972): Numerical methods for unconstrained minimization problems, Academic Press, London.

OLSDER, G.J. and STRIJBOS, R.C.W. (1973). World Dynamics, a dynamic optimization study, Annals of Systems Research, 3, pp. 21-37.

PAGUREK, B. and WOODSIDE, C.M. (1968): The conjugate gradient method for optimal control problems with bounded control variables. Automatica, 4, pp. 337-349.

PIERSON, B.L. and RAJTORA, S.G. (1970): Computational Experience with the Davidon Method Applied to Optimal Control Problems, IEEE Trans Syst.Sci and Cybern., SSC-6, pp. 240-242.

QUINTANA, V.H. and DAVISON, E.J. (1974). Clipping-off gradient algorithms to compute optimal controls with constrained magnitude, Int.J. Control, 20, pp. 245-255.

RADEMAKER, O. (1972): Project Group Global Dynamics, Progress Reports nrs. 1,2 (1972), 3,4 (1974). Available from author, Address: T.H.E., P.O.Box 513, Eindhoven, The Netherlands.

SINNOTT, J.F. and LUENBERGER, D.G. (1967): Solution of Optimal Control Problems by the Method of Conjugate Gradients. JACC 1967 Preprints, pp. 566-574.

TRIPATHI, S.S. and NARENDRA, K.S. (1968): Conjugate direction methods for nonlinear optimization problems, Proc. 1968 NEC (Chicago, Ill.), pp. 125-129.

WONG, P.J., DRESSLER, R.M. and LUENBERGER, D.G. (1971): A combined Parallel-Tangents/ Penalty-Function Approach to Solving Trajectory Optimization Problems, AIAAJ 9, pp. 2443-2448.

F_3(MSL) (BRMM)	F_{12}(POLR) (DRPM)	F_{17}(FR) (BRFM)	F_{28}(POLR) (FRM)	F_{36}(FR) (CFIFR)	F_{40}(FR) (QLF)
1.16452528	9.27142853 -1	1.07788100 -1	1.03523809	1.0009320	-2.56917756 -2
-1.80875430 -1	1.37142858 -2	1.07280700	-1.21587300 -2	-8.7828670 -1	1.13038139
2.07672036 -2	2.07142857 -3	-1.96197200 -1	-5.33333337 -4	6.2762180 -2	-1.11816125 -1
-7.27686482 -4		1.19048700 -2	7.77777778 -6	1.78613100 -1	
	F_{13}(FR) (DRFM)			-6.50349800 -2	F_{41}(POLR) (QLP)
	4.36114600	F_{18}(POLR) (BRPM)		6.66666680 -3	1.04928571
F_6(NRFR) (NREM)	-6.46689100	1.02925871			-1.86865081 -2
7.00241120 -2	4.24544000	-9.48015873 -3	F_{32}(CIR) (POLCM)		-3.54761899 -4
-4.20335073 -1	-1.37198100	-5.63095238 -4	-7.17151600 -2		6.38888884 -6
3.79442180	2.19004600 -1	7.77777778 -6	4.79241100 -1	F_{38}(MSL) (QLM)	
-2.49366758	-1.38490000 -2		1.05333300	2.17053700 -1	F_{42}(MSL) (NRMM)
			-2.80868200 -1	9.36667000 -1	-5.66460900 -3
	F_{14}(CR) (DRCM)	F_{20}(CR) (FCM)	2.90083700 -2	-1.0012200 -1	1.13683700
F_{11}(MSL) (DRMM)	9.00000000 -1	1.57662780	-1.35139100 -3	3.3387600 -3	-1.38493300 -1
2.97868000	5.00000000 -2	-6.60394837 -1	2.36467500 -5		1.06418400 -2
-3.04801000	5.00000000 -2	8.98694540 -2			-2.62043700 -4
1.53463000				F_{39}(CR) (QLC)	
-3.99813000 -1	F_{16}(CR) (BRCM)	F_{21}(CIRA) (FPCI)	F_{34}(POLR) (POLAT)	1.99832182	F_{43}(F_{38}/F_{40})(CLQR)
5.79524000 -2	1.04801587	5.09934300 -1	5.92857148 -1	-1.66214850	5.55944000 -1
-4.69843000 -3	2.81084673 -2	5.43464300 -1	1.6480159 -1	8.99824969 -1	7.69541600 -1
1.98890000 -4	-7.63888896 -2	-5.57696700 -2	2.80952381 -3	-2.86928923 -1	-8.60140000 -2
-3.41888000 -6	1.01851852 -2	1.83614000 -3	-2.77777777 -6	3.73391612 -2	-2.64180300 -3

Table 2.1. Coefficients of the nonnegative powers of the polynomial approximations of the coupling functions of Forrester.

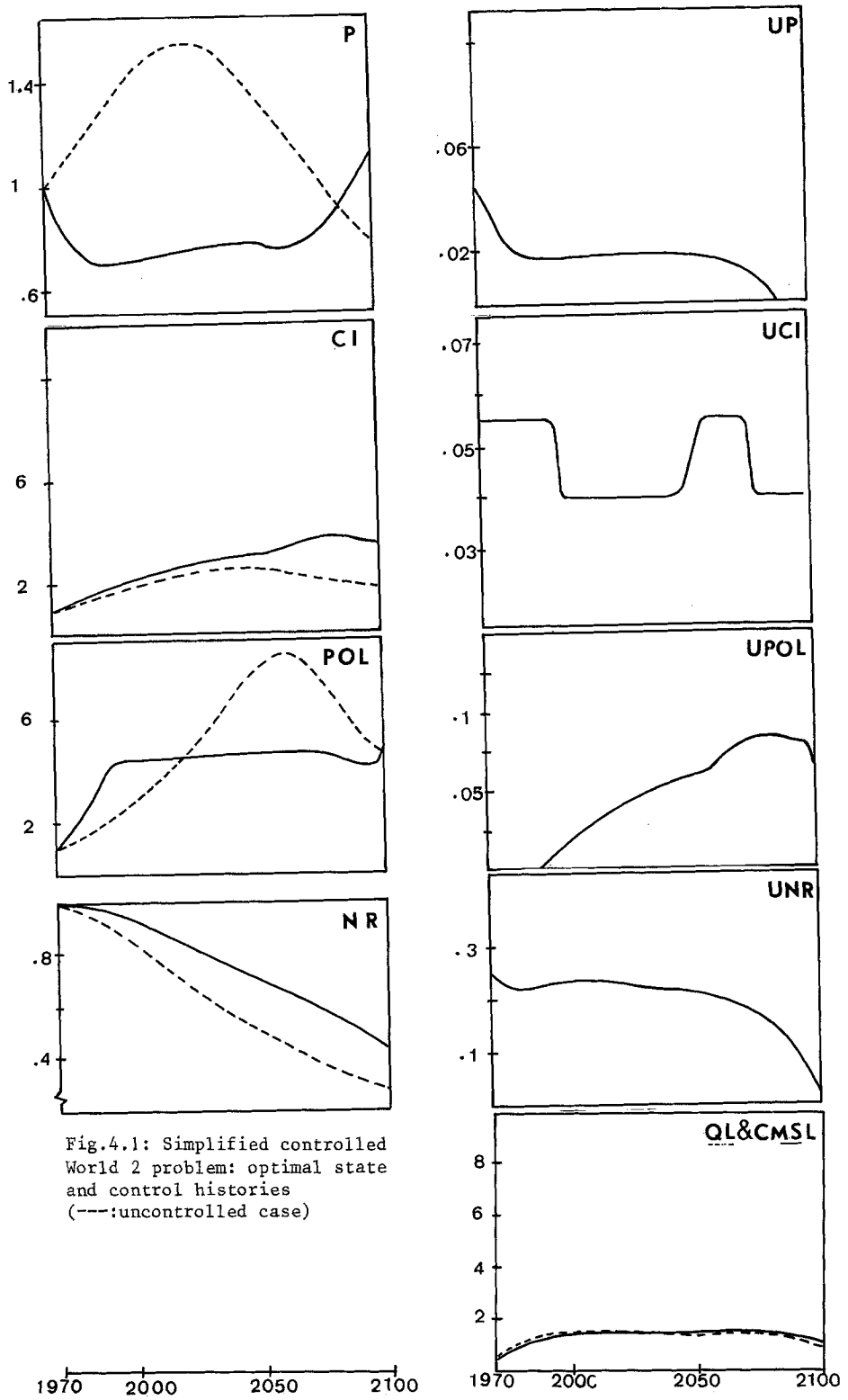

Fig.4.1: Simplified controlled World 2 problem: optimal state and control histories (---:uncontrolled case)

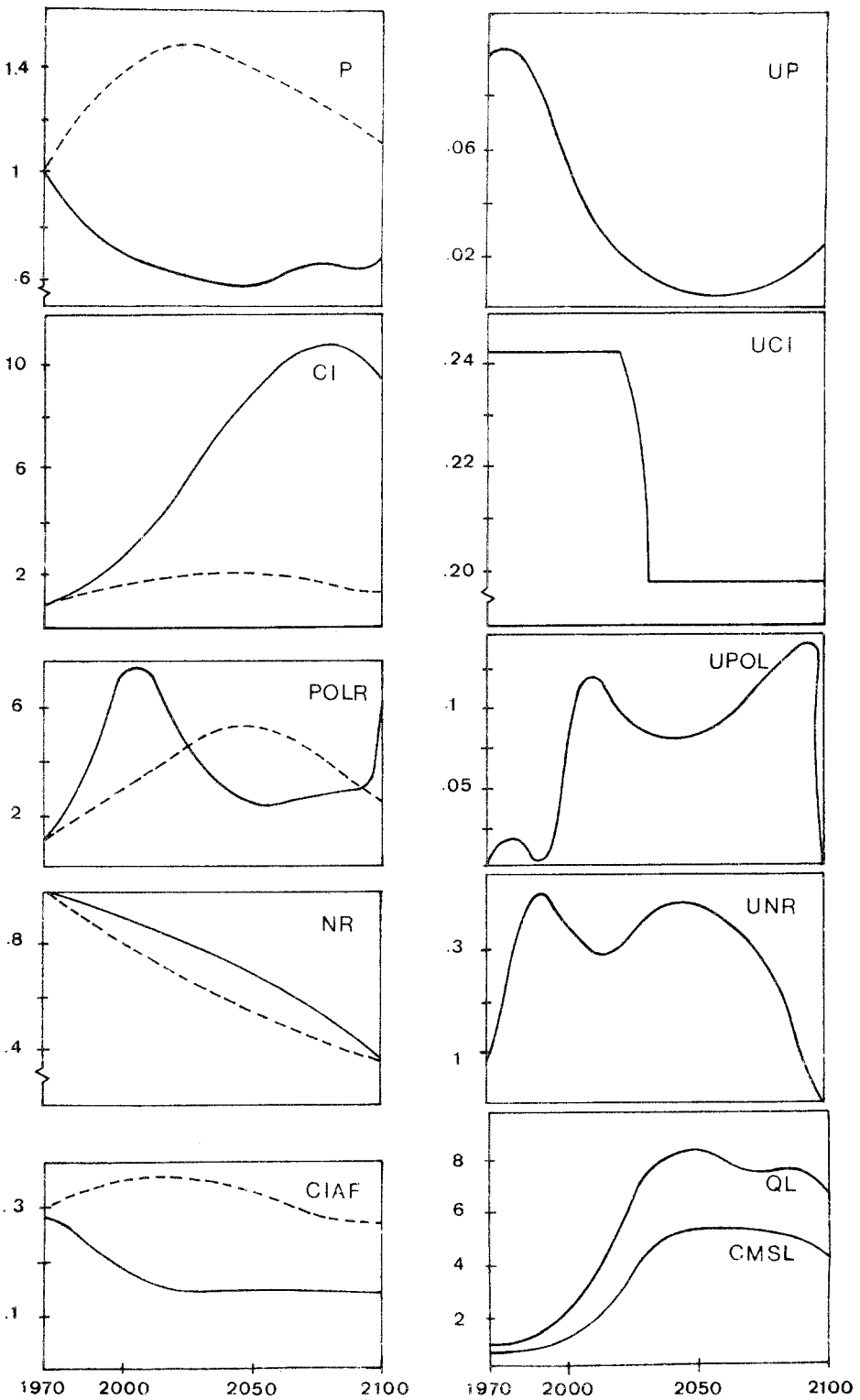

Fig.4.2: Complete controlled World 2 problem: optimal state and control histories
(----- uncontrolled case)

METHOD	RESET	ITER-S	CALLS	CRITERION	TIME	C/I	T/C
SD	1	62	324	178.910560	618	5.2	1.9
PARTAN	6	82	431	.910651	804	5.3	1.9
	12	89	434	.910814	838	4.9	1.9
	18	> 100	496	.909531	862	5.0	1.7
CGI	6	42	226	.910866	376	5.4	1.7
	12	38	204	.910802	338	5.4	1.7
	18	33	188	.910624	263	5.7	1.4
CGIIA	6	50	260	.910909	1493	5.2	5.7
	12	44	233	.911205	1262	5.3	5.4
	18	48	244	.910993	1358	5.1	5.6
CGIIB	6	61	308	.910996	1464	5.1	4.8
	12	56	271	.910523	1128	4.9	4.1
	18	64	300	.910754	1182	4.7	3.9
DFP	6	51	317	.910923	459	6.2	1.4
	12	51	339	.910856	668	6.6	2.0
	18	48	312	.910846	627	6.5	2.0
$\beta^{(i)}=1.0$	6	56	268	.910480	480	4.8	1.8
	12	77	310	.910704	704	4.0	2.3
	18	92	360	.910848	848	4.0	2.4

Table 4.3: Simplified controlled World 2 model: Numerical results of the application of different methods in combination with the clipping-off technique. (">": convergence conditions not yet satisfied).

METHOD	RESET	ITER-S	CALLS	CRITERION	TIME	C/I	T/C
SD	1	92	523	178.910867	901	5.7	1.7
PARTAN	6	81	486	.907703	787	6.0	1.6
	12	72	444	.909202	671	6.2	1.5
	18	71	435	.909373	717	6.1	1.6
CGI	6	38	228	.908535	350	6.0	1.5
	12	69	413	.910531	686	6.0	1.7
	18	64	340	.910400	624	5.3	1.8
CGIIA	6	46	294	.907522	1539	6.4	5.2
	12	> 46	300	.910353	2000	6.5	6.7
	18	49	315	.910747	1783	6.4	5.6
CGIIB	6	62	373	.908684	1393	6.0	3.7
	12	57	333	.906983	1882	5.8	5.7
	18	> 65	343	.905315	1989	5.3	5.8
DFP	6	59	319	.910806	611	5.4	1.9
	12	> 94	459	.908187	907	4.9	2.0
	18	> 89	413	.904944	905	4.6	2.2
CGI	6	40	214	.910855	430	5.3	2.0
(clip)	12	37	196	.910946	394	5.3	2.0
	18	35	191	.911046	371	5.5	1.9

Table 4.4: Simplified controlled World 2 model: Numerical results of the application of different methods in combination with the transformation technique. (">": convergence conditions not yet satisfied).

METHOD	RESET	ITER-S	CALLS	CRITERION	TIME	C/I	T/C
CGI	3	64	316	178.910590	556	4.9	1.8
	6	42	226	.910866	376	5.4	1.7
	12	38	204	.910802	338	5.4	1.7
	18	33	188	.910624	263	5.7	1.4
	24	37	207	.910701	268	5.6	1.3
	30	40	223	.910836	357	5.6	1.6
	100	45	237	.910747	480	5.3	2.0
DFP	3	64	372	.910694	706	5.8	1.9
	6	51	317	.910923	459	6.2	1.4
	12	51	339	.910856	668	6.6	2.0
	18	48	312	.910846	627	6.5	2.0
	24	49	323	.910921	476	6.6	1.5
	30	45	295	.910964	594	6.6	2.0
	100	45	302	.910965	637	6.7	2.1
CGI	3	61	304	.910519	604	5.0	2.0
(modif.)	6	62	293	.910976	550	4.7	1.9
	12	51	253	.910996	400	5.0	1.6
	18	50	252	.910926	476	5.0	1.9
	24	51	341	.910782	444	4.7	1.8
	30	57	368	.910942	515	4.7	1.9
	100	98	371	.910984	814	3.8	2.2

Table 4.5: Simplified controlled World 2 model: Comparison of different reset values for three methods in combination with the clipping-off technique.

	METHOD	RESET	ITER-S	CALLS	CRITERION	TIME	C/I	T/C
A	CGI	6	43	185	178.909905	364	4.3	2.0
	(trsf)	12	47	203	.910616	512	4.3	2.5
		18	59	251	.911060	516	4.2	2.1
	DFP	6	52	279	.908612	670	4.7	2.5
	(trsf)	12	90	438	.909496	997	5.4	2.4
		18	> 74	352	.903749	905	4.9	2.3
	CGI	6	44	201	.911001	373	4.6	1.9
	(clipp.-	12	40	181	910982	409	4.5	2.3
	off)	18	36	167	911382	365	4.6	2.2
B	CGI	6	38	228	.908535	350	6.0	1.5
	(trsf)	12	69	413	.910531	686	6.0	1.7
		18	64	340	.910400	624	5.3	1.8
	DFP	6	59	319	.910806	611	5.4	1.9
	(trsf)	12	> 94	459	.908187	907	4.9	2.0
		18	> 89	413	.904944	905	4.6	2.2
	CGI	6	40	214	.910855	430	5.3	2.0
		12	37	196	.910946	394	5.3	2.0
		18	35	191	.911046	371	5.3	1.9

Table 4.6: Simplified controlled World 2 model:Comparison of different initial stepsize strategies in linesearch: (A):$\alpha_{Start}=0.5\alpha^{(i-1)}$; (B):α_{start} = fixed. (">" : convergence conditions not yet satisfied.)

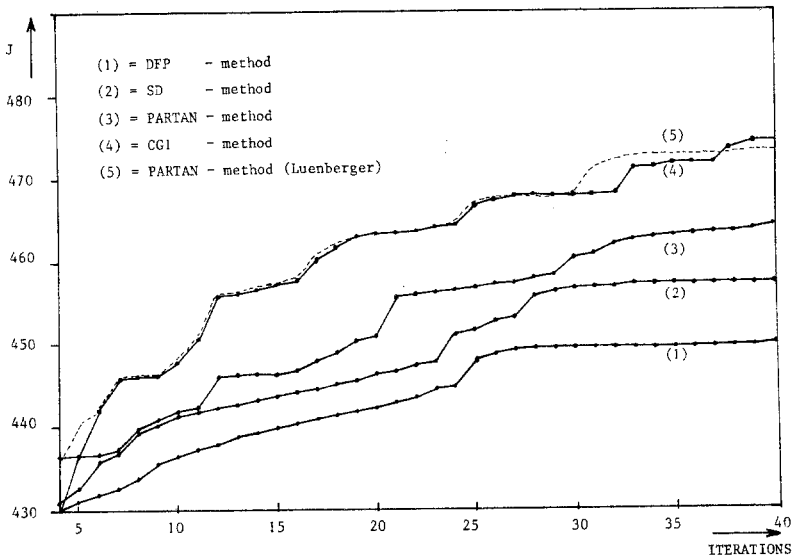

Fig.4.7 Complete controlled World 2 model: Convergence histories of 4 gradient methods in combination with the transformation technique with (k_1=k_2=k_3=k_4=1.0)

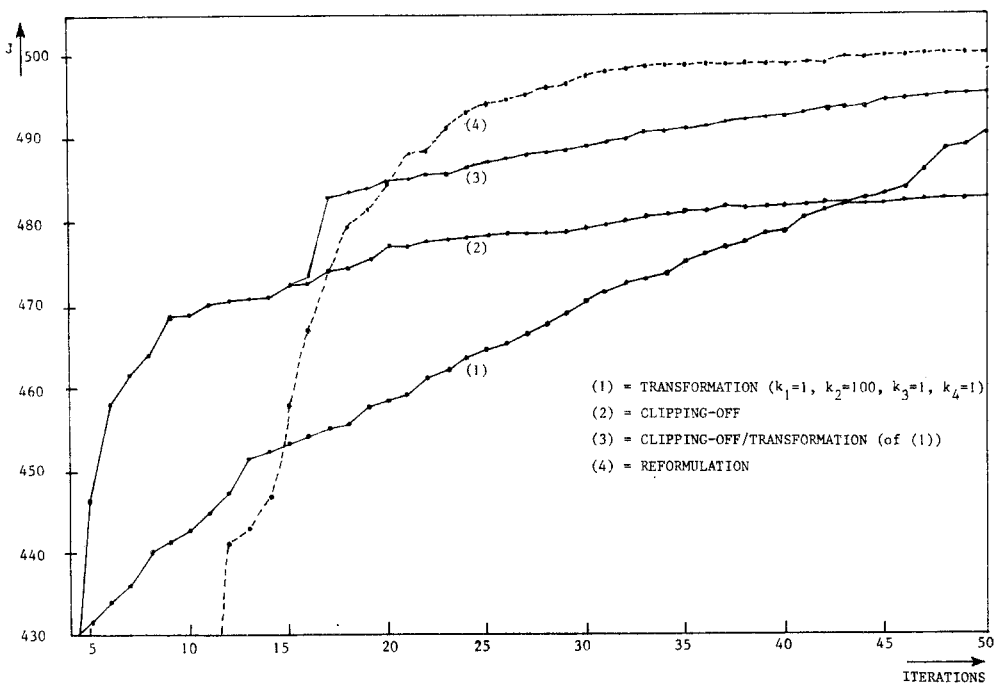

Fig.4.8 Complete controlled World 2 model: Convergence histories of the CGI-method in combination with different techniques for taking into account bounds on the values of the control variables.