DEPARTMENT OF STATISTICS

University of Wisconsin
Madison, Wisconsin 53706

TECHNICAL REPORT NO. 316

November 1972

D-OPTIMAL DESIGNS FOR DYNAMIC MODELS

Part II. Applications

By

Bernard Viort

Typist: Jacquelyn R. Forer

This research was supported by the Air Force Office of Scientific Research under Grant No. AFOSR-72-2363 and by the Wisconsin Alumni Research Foundation.

D-OPTIMAL DESIGNS FOR DYNAMIC MODELS

Part II. Applications

In a preceding report (Viort [4]) some results on the theory of C-constrained D-optimal designs were presented. The present report is intended to be an illustration of the methods proposed: the consideration of different examples raises some interesting questions and remarks concerning both the theoretical situation considered in [4] and the possible applications. The numerical methods developed in relation with the notion of D-optimality allow for a greater flexibility in judging the overall quality of a design, suggesting the need for a more general theory.

Throughout this report the notations are those of [4].

I. On the Example of Box and Jenkins

Looking into a new problem it is priceless to have at hand the solution in a simple (but not trivial) situation: this important help was provided here by the example presented in Box and Jenkins ([1] pp. 416-420).

They considered the model

$$(1 - \delta B)y_t = \omega x_{t-1} + a_t \tag{1}$$

where a_t is a white noise process with variance σ_a^2, and found the different D-optimal inputs:

- For C_1, $(1 - \delta B)x_t$ = white noise
- For C_2, $(1 + \delta B)x_t$ = white noise
- For C_3, x_t = white noise

(in all three cases the variance of the white noise process is easy to determine).

I.1. Analytical results.

The small number of unknown parameters makes it possible to give a complete illustration of the method.

With the notation of [4], the model (1) is to be considered as

$$y_t = \frac{\omega}{1 - \delta B} x_t + \frac{1}{1 - \delta B} a_t \tag{2}$$

leading to:

$$|\omega(\theta)|^2 = \omega^2 \tag{3}$$

$$|\delta(\theta)|^2 = (1-\delta e^{-i\theta})(1-\delta e^{i\theta}) = 1 + \delta^2 - 2\delta \cos\theta \tag{4}$$

$$G(\theta) = \frac{\omega}{1 - \delta e^{-i\theta}} \tag{5}$$

$$\frac{\partial G}{\partial \omega}(\theta) = \frac{1}{1 - \delta e^{-i\theta}} \tag{6}$$

$$\frac{\partial G}{\partial \delta}(\theta) = \frac{\omega e^{-i\theta}}{(1 - \delta e^{-i\theta})^2} \tag{7}$$

$$f_e(\theta) = \frac{1 - \delta^2}{2\pi} \frac{1}{|\delta(\theta)|^2} \tag{8}$$

For $|\delta| < 1$ all the functions to be considered below are holomorphic in the neighborhood of the unit circle $|z| = 1$. The integral on this curve will be represented by $\int_{\mathcal{C}}$.

A. Constraint C_1.

In this case the optimal solution of Box and Jenkins is $f_x(\theta) = f_e(\theta)$ and since

$$\frac{1}{2\pi}\int_0^{2\pi} \frac{d}{|\delta(\theta)|^2} = \frac{1}{2i\pi}\int_{\mathcal{C}} \frac{dz}{(1-\delta z)(z-\delta)} = \frac{1}{1-\delta^2} \tag{9}$$

$$\frac{1}{2\pi}\int_0^{2\pi} \frac{\omega e^{i\theta}\, d\theta}{|\delta(\theta)|^2\,(1-\delta e^{i\theta})} = \frac{1}{2i\pi}\int_{\mathcal{C}} \frac{\omega_z dz}{(1-\delta z)^2(z-\delta)} = \frac{\omega\ \delta}{(1-\delta^2)^2} \tag{10}$$

$$\frac{1}{2\pi}\int_0^{2\pi} \frac{\omega^2}{|\delta(\theta)|^4}\, d\theta = \frac{1}{2i\pi}\int_{\mathcal{C}} \frac{z\ dz}{(1-\delta z)^2\ (z-\delta)^2} = \frac{\delta^2(1+\delta^2)}{(1-\delta^2)^3} \tag{11}$$

one has

$$I_{\omega,\delta}(f_x) \propto \begin{bmatrix} \frac{1}{1-\delta^2} & \frac{\omega\ \delta}{(1-\delta^2)^2} \\ \frac{\omega\ \delta}{(1-\delta^2)^2} & \frac{\omega^2(1+\delta^2)}{(1-\delta^2)^3} \end{bmatrix} \tag{12}$$

$$\det\ (I_{\omega,\delta}(f_x)) \propto \frac{\omega^2}{(1-\delta^2)^4} \tag{13}$$

and

$$d_{f_x}(\theta) = \frac{|\delta(\theta)|^2}{1-\delta^2} \,\mathrm{Tr} \left\{ [I_{\omega,\delta}(f_x)]^{-1} \begin{bmatrix} \dfrac{1}{|\delta(\theta)|^2} & \dfrac{\omega\, e^{i\theta}}{|\delta(\theta)|^2(1-\delta e^{i\theta})} \\ \dfrac{\omega\, e^{-i\theta}}{|\delta(\theta)|^2(1-\delta e^{-i\theta})} & \dfrac{\omega^2}{|\delta(\theta)|^4} \end{bmatrix} \right\} \tag{14}$$

$$d_{f_x}(\theta) = (1+\delta^2) + \frac{(1-\delta^2)^2}{|\delta(\theta)|^2} - \frac{\delta}{\omega}(1-\delta^2)\left[\frac{\omega e^{i\theta}}{1-\delta e^{i\theta}} + \frac{\omega e^{-i\theta}}{1-\delta e^{-i\theta}}\right] \tag{15}$$

$$= 1 + \delta^2 + \frac{1-\delta^2}{|\delta(\theta)|^2}\left[(1-\delta)^2 - 2\delta \cos\theta + 2\delta^2\right] \tag{16}$$

$$= 2 \tag{17}$$

proving that, at least in this case, the proposed method works. Note that the condition i) of Theorem VII of [4] is verified in this case, since $|\delta(\theta)|^4 f_e(\theta) \propto (1 + \delta^2 - 2\delta \cos\theta)$ a polynomial of degree 1 in $\cos\theta$.

B. Constraint C_2.

The optimal solution is now:

$$f_x(\theta) = \frac{1-\delta^2}{2\pi} \frac{1}{(1+\delta e^{i\theta})(1+\delta e^{-i\theta})} \tag{18}$$

and one has:

$$\frac{1}{2\pi}\int_0^{2\pi} \frac{d\theta}{(1+\delta e^{-i\theta})(1+\delta e^{i\theta})} = \frac{1}{1-\delta^2} \tag{19}$$

$$\frac{1}{2\pi}\int_0^{2\pi}\frac{\omega\, e^{i\theta}\, d\theta}{(1-\delta e^{i\theta})(1+\delta e^{i\theta})(1+\delta e^{-i\theta})} = \frac{1}{2i\pi}\int_{\mathcal{C}}\frac{z\, dz}{(1-\delta z)(1+\delta z)(z+\delta)} = -\frac{\omega\,\delta}{(1+\delta^2)(1-\delta^2)} \tag{20}$$

$$\frac{1}{2\pi}\int_0^{2\pi}\frac{\omega^2\, d\theta}{(1-\delta e^{i\theta})(1-\delta e^{-i\theta})(1+\delta e^{i\theta})(1+\delta e^{-i\theta})} = \frac{1}{2i\pi}\int_{\mathcal{C}}\frac{\omega^2\, z\, dz}{(1-\delta z)(1+\delta z)(z-\delta)(z+\delta)} = \frac{\omega^2}{(1+\delta^2)(1-\delta^2)} \tag{21}$$

and

$$I_{\omega,\delta}(f_x) \propto \begin{bmatrix} \dfrac{1}{1-\delta^2} & -\dfrac{\omega\,\delta}{(1+\delta^2)(1-\delta^2)} \\ \\ -\dfrac{\omega\,\delta}{(1+\delta^2)(1-\delta^2)} & \dfrac{\omega^2}{(1+\delta^2)(1-\delta^2)} \end{bmatrix} \tag{22}$$

$$\det\,(I_{\omega,\delta}(f_x)) \propto \frac{\omega^2}{(1+\delta^2)^2(1-\delta^2)^2} \tag{23}$$

leading to

$$d_{f_x}(\theta) = \frac{|\delta(\theta)|^2}{1-\delta^2} \operatorname{Tr} \left\{ \left[I_{\omega,\delta}(f_x) \right]^{-1} \begin{bmatrix} \dfrac{1}{|\delta(\theta)|^2} & \dfrac{\omega\, e^{i\theta}}{|\delta(\theta)|^2 (1-\delta e^{i\theta})} \\ \dfrac{\omega\, e^{-i\theta}}{|\delta(\theta)|^2 (1-\delta e^{-i\theta})} & \dfrac{\omega^2}{|\delta(\theta)|^4} \end{bmatrix} \right\} \tag{24}$$

i.e., after some computations:

$$d_{f_x}(\theta) = 2\, \frac{(1+\delta^2)}{|\delta(\theta)|^2} \tag{25}$$

and since, because of (21)

$$\int_0^{2\pi} |G(\theta)|^2\, f_x(\theta)\, d\theta = \frac{\omega^2}{1+\delta^2} \tag{26}$$

$$d_{f_x}(\theta) = 2\, \frac{|G(\theta)|^2}{\int_0^{2\pi} |G(\theta)|^2\, f_x(\theta) d\theta} \tag{27}$$

or

$$\Delta^2_{f_x}(\theta) = 0 \tag{28}$$

which is the condition for C_2-constrained D-optimality.

C. Constraint C_3

The details of the computations will not be derived for the simple reason that the optimal input (white noise) of Box and Jenkins turns out not to be optimal (for the method developed in [4]). In order to understand the reason for this discrepancy, it is necessary to consider the assumption underlying the general method, as will be done now.

I.2. Some remarks on the method of [4].

Consider first the model of Box and Jenkins:

$$y_t = \delta\, y_{t-1} + \omega\, x_{t-1} + a_t \tag{29}$$

$\{a_t\}$ being white noise. It is proved in Minnich [3] that the information matrix per observation of (ω,δ) is for n large

$$I_{\omega,\delta} = \frac{1}{\sigma_a^2}\begin{bmatrix} E[X_t^2] & E[X_t Y_t] \\ E[X_t Y_t] & E[Y_t^2] \end{bmatrix} \tag{30}$$

and, when X_t = white noise with variance σ_x^2, it is easy to show that

$$I_{\omega,\delta} = \begin{bmatrix} \dfrac{\sigma_x^2}{\sigma_a^2} & 0 \\ 0 & \dfrac{\omega^2}{1-\delta^2}\dfrac{\sigma_x^2}{\sigma_a^2} + \dfrac{1}{1-\delta^2} \end{bmatrix} \tag{31}$$

For the same white noise input, the method of [4] gives

$$\tilde{I}_{\omega,\delta} = \begin{bmatrix} \frac{\sigma_x^2}{\sigma_a^2} & 0 \\ 0 & \frac{\omega^2}{1-\delta^2}\frac{\sigma_x^2}{\sigma_a^2} \end{bmatrix} \tag{32}$$

corresponding to the model

$$y_t = \frac{\omega}{(1-\delta B)} x_t + \frac{1}{1-\delta B} a_t \, . \tag{33}$$

The difference between $I_{\omega,\delta}$ and $\tilde{I}_{\omega,\delta}$ comes from the difference between the informations on the estimator of δ:

$$\frac{\omega^2}{1-\delta^2}\frac{\sigma_x^2}{\sigma_a^2} + \frac{1}{1-\delta^2} \quad \text{vs.} \quad \frac{\omega^2}{1-\delta^2}\frac{\sigma_x^2}{\sigma_a^2}$$

and the reason for this is the fact that δ appears both in the dynamic part and the error part of (33), while only the information from the dynamic part is taken into account in the method of [4]. Does this mean that the estimators considered in [4] are not efficient? Not at all since it is assumed there that the error structure (i.e., here the parameter δ) is known: the special input $\{X_t = 0\}$ is suggested as a good input to determine it accurately. The rigourous application of the method suggested in [4] would require that when the dynamic part has some parameters in common with the error part, these parameters should not be considered for the design problem. This implies that the example of Box and Jenkins is, from this point of view, a one parameter problem: it is therefore necessary to explain the results of I.1. (A and B). This can be done by considering the determinant of (31)

$$\det(I_{\omega,\delta}) = \frac{1}{(\sigma_a^2)^2}\left[\sigma_x^2\,\sigma_y^2 - (\sigma_a^2)^2\,E^2[X_tY_t]\right] \tag{34}$$

i.e., writing

$$\sigma_y^2 = \sigma_{y_1}^2 + \sigma_{y_2}^2 \tag{35}$$

(where $\sigma_{y_1}^2$ is a function of the design while $\sigma_{y_2}^2$ is a constant) one wants to maximize

$$\sigma_x^2\,\sigma_{y_1}^2 + \sigma_x^2\,\sigma_{y_2}^2 - (\sigma_a^2)^2\,E[X_tY_t] \tag{36}$$

and for C_1 this is equivalent to

$$\begin{cases} \text{maximize:} & \sigma_x^2\,\sigma_{y_1}^2 - (\sigma_a^2)^2\,E^2[X_tY_t] \\ \text{subject to} & \sigma_x^2 \leq c_1 \end{cases} \tag{37}$$

for C_2 this is equivalent to

$$\begin{cases} \text{maximize:} & \sigma_x^2\,\sigma_{y_1}^2 - (\sigma_a^2)^2\,E^2[X_tY_t] \\ \text{subject to} & \sigma_{y_1}^2 \leq c_2 - \sigma_{y_2}^2 \end{cases} \tag{38}$$

so that the optimal solutions are basically the same in the two methods, but for slight changes in numerical values as will be shown later.

It is also clear that, as soon as $m > 2$, there is no decomposition like (36) and the solutions will differ.

This point being clarified, it is necessary to consider another question: how good is the improper use of the method for all the dynamic parameters? Once again, the answer is provided by (31): when $\frac{\sigma_x^2}{\sigma_a^2}$ (the signal-to-noise ratio) is large, one can expect that there will be little difference between (31) and (32), and therefore that the method can be applied without significant error to all the parameters. In the opposite case, when $\frac{\sigma_x^2}{\sigma_a^2}$ is small, it is clear that one loses a lot of information on δ, and also that one has very little information on ω: all these results are in agreement with one's common sense.

I.3. Numerical results.

The iterative procedures of [4] are particularly well adapted to numerical computations: a program implementing these methods has been developed for the construction of nearly D-optimal constrained designs. It was run for all three examples: in each case the first choice for the input was white noise. The results are summarized in Table 1, together with the results of Box and Jenkins. [The value $\Delta^{-1/2}$ of Box and Jenkins is approximately proportional to the inverse of the "Information" value considered here: this is due to the facts that they omitted the factor σ_a^2, and that the constraints are slightly different (I.2.).]

Constraint	Results of B. & J. Variances	$\Delta^{-\frac{1}{2}}$	Information for white noise	Optimal Design Variances	Optimal Design Information
C_1: $\sigma_x^2 = 1$	$\sigma_y^2 = 2.49$	.70	23.41	$\sigma_y^2 = 2.42$	31.20
C_2: $\sigma_y^2 = 2.49$	$\sigma_x^2 = 2.91$	.42	69.02	$\sigma_x^2 = 2.86$	92.04
C_3: $\sigma_x^2\,\sigma_y^2 = 7.25$	$\sigma_x^2 = 2.29$ $\sigma_y^2 = 3.16$	.37	119.33	$\sigma_x^2 = 2.23$ $\sigma_y^2 = 3.24$	119.37

Table 1.

The remarks of I.2 justify the small discrepancies of the variances. The last line shows clearly that white noise is almost optimal for the C_3-constrained problem (in the set-up of [4]).

It is also very interesting to compare the solutions of Box and Jenkins with those of the numerical computations: for constraints C_1 and C_2 it turns out that one has at hand, for each case, two optimal solutions (one stochastic and one mixed). These are shown in Figures 1a, 1b, 2a, 2b: in Figures 1b, 2b the shaded area represents the contribution of white noise and the heavy lines the deterministic components (discrete masses in the spectral density).

Recalling the results of [4], a discrete mass p at θ corresponds to a deterministic input

$$\pm \sigma_x^2 \, p \cos\theta t \tag{39}$$

where the sign is decided by flipping a fair coin. Considering now Figure 1b, the corresponding "optimal" solution is, since $\theta = 0$,

$$\pm \sigma_x^2 \, p + \text{white noise} \tag{40}$$

This is in opposition with the requirements of the experiment, which was to be performed with a "mean value" (i.e., time average) zero. (This is a classical example of non-ergodic process, i.e., a process where the expectation is not equal to the time average.)

One can think of two ways to overcome this difficulty of discrete masses at 0:

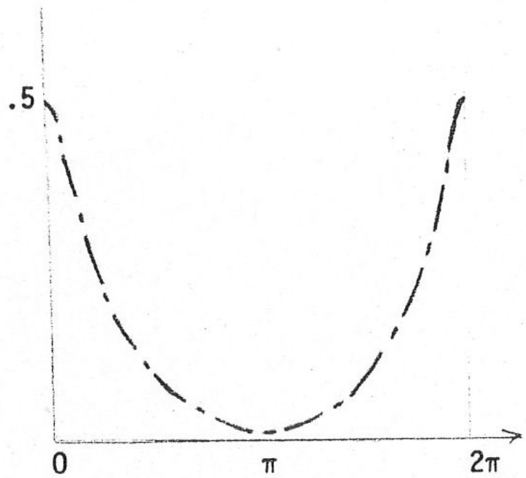

Figure 1a

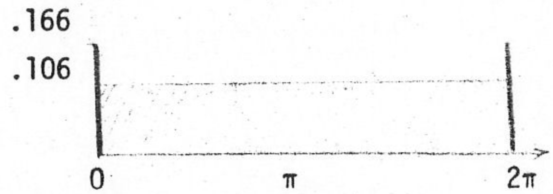

Figure 1b

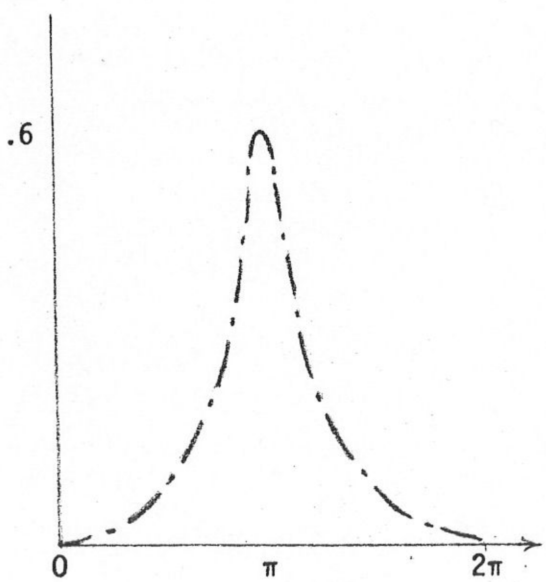

Figure 2a

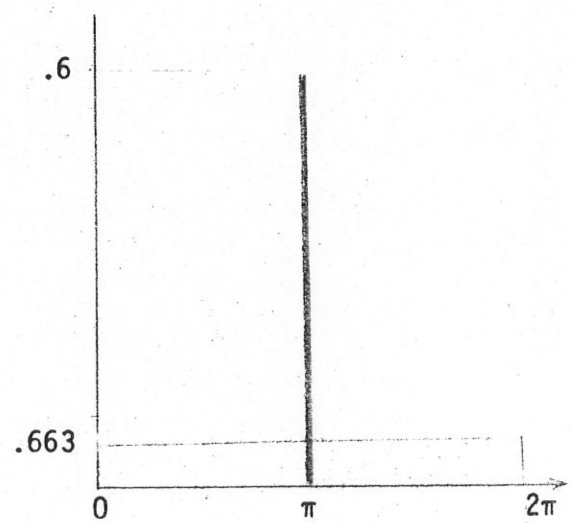

Figure 2b

Optimal solutions for the model $(1-.5B) = x_{t-1} + a_t$

1 - Constraint C_1

2 - Constraint C_2

a - Stochastic solution (B & J)

b - Mixed solution

- Exclude small and large values of θ in the algorithm, i.e., use only

$$0 < \Theta \leq \theta < 2\pi - \Theta < 2\pi \qquad \Theta \text{ small} \tag{41}$$

There is a good chance that this will lead to another almost D-optimal design with discrete masses at Θ, $2\pi - \Theta$, and so Θ has to be chosen according to the experiment.

- Comparing Figures 1a and 1b one can think to the following rule: when there is a discrete mass at 0, replace it by an autoregressive process

$$(1-dB)x_t = \text{white noise} \qquad d > 0 \tag{42}$$

where d and the variance of white noise are suitably chosen. There is still a problem with such a method: it was shown in [4] that it may happen that no stochastic or mixed solution can be optimal--and it is then necessary to consider deterministic solutions with restrictions on the range of θ.

I.4. Transfer of energy.

Considering the transfer of energy from x to y in the model (1) Box and Jenkins give the following interpretation of their results:

- When σ_x^2 is constrained, the optimal solution achieves a maximum transfer of energy from x to y $(f_x(\theta) \propto |G(\theta)|^2)$;

- When σ_y^2 is constrained, the optimal solution achieves a minimum transfer from x to y;
- When $\sigma_x^2 \times \sigma_y^2$ is constrained, white noise should be (and in fact is) a good compromise.

The reason for this behavior is not completely elucidated now, but seems to be related to general considerations on designs of experiments for non-linear models as follows:

- Consider first the case of one parameter α, i.e., the model

$$y_\theta = f(\theta,\alpha) + e_\theta \qquad \theta\varepsilon[0,2\pi] \tag{42}$$

where $f(\theta,\alpha)$ is a non-linear smooth function of α and e_θ are independent r.v. with variance $V(\theta)$.

If it is known that α is close to α_0, one would obviously take measurements at θ_0 such that

$$\frac{\left\{ \frac{\partial f}{\partial \alpha} (\theta_0,\alpha)\Big|_{\alpha=\alpha_0} \right\}^2}{V(\theta_0)} = \max_\theta \frac{\left\{ \frac{\partial f}{\partial \alpha} (\theta,\alpha)\Big|_{\alpha=\alpha_0} \right\}^2}{V(\theta_0)} \tag{43}$$

i.e., at a point θ_0 where, for a given $d\alpha$, the variation of the curve is maximum as compared to the variance of the error (see Figure 3).

- When there is more than one parameter, the same type of reasoning could be carried on, with different possible measures of the variation of the curve for a given variation of the parameters. This is a first difficulty, a second one being the fact that it

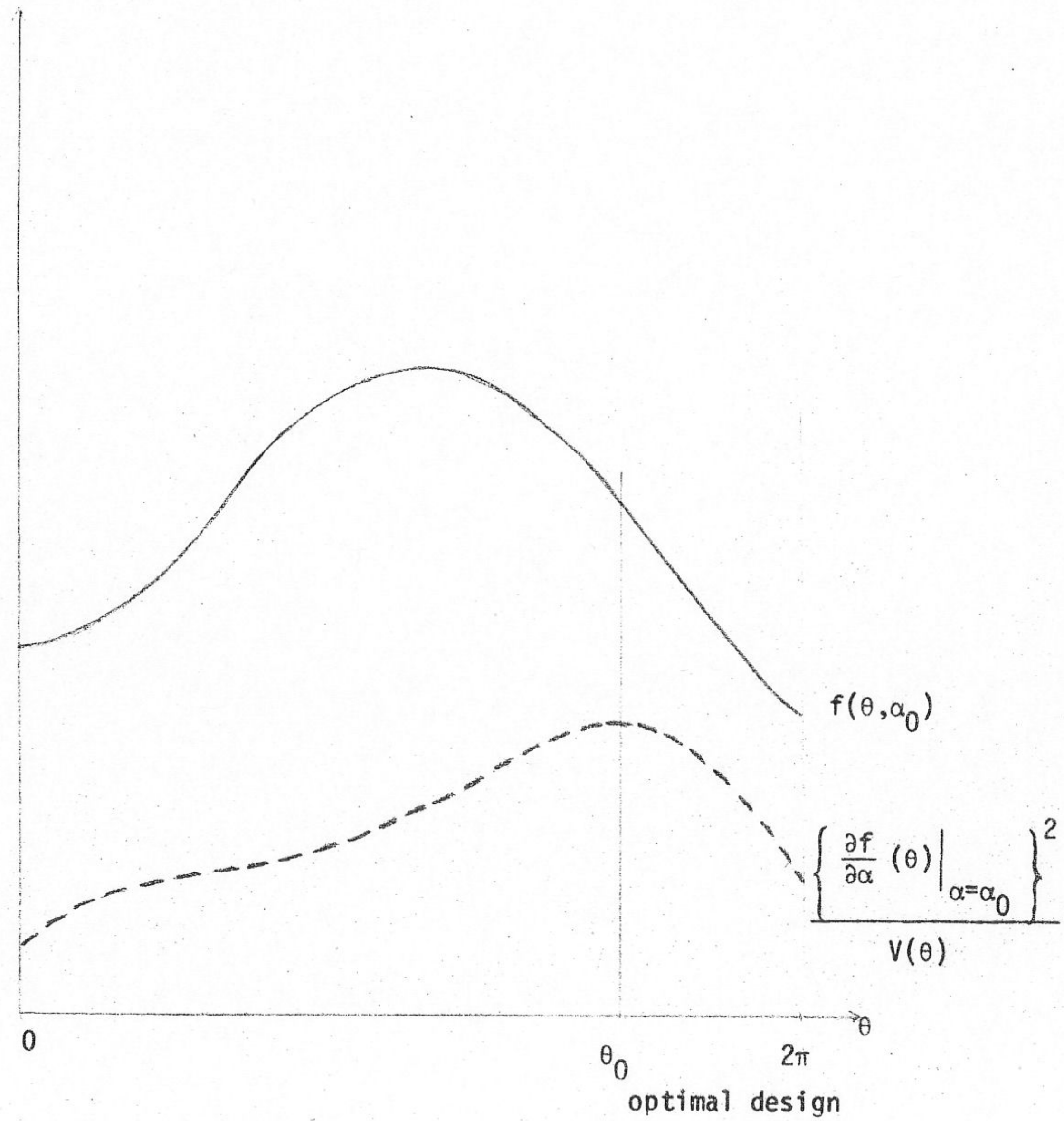

Figure 3. Optimal design for the case of one parameter.

is necessary to consider designs on more than one point: one point gives information only for some linear compilation of the parameters.

- Now the approach to the design problem considered in [4] consists in expressing the model

$$y_t = \frac{\omega(B)}{\delta(B)} x_t + e_t \tag{44}$$

as

$$z_\theta = G_{\omega,\delta}(\theta) + \varepsilon_\theta \tag{45}$$

where

$$G_{\omega,\delta}(\theta) = \frac{\omega(e^{-i\theta})}{\delta(e^{-i\theta})} \tag{46}$$

ε_θ are independent r.v. with variance proportional to $f_e(\theta)$ the spectral density of the process $\{e_t\}$

The solution of the design problem is a function $f_x(\theta)$ (the spectral density of $\{x_t\}$), which represents the "proportion" of the total number of measurements to be taken at the point θ in the model (45).

One measure of the variation of the curve z_θ for a given variation of the parameters is the trace of the matrix

$$M(\theta) = \begin{bmatrix} \frac{\partial G}{\partial \omega_0}(\theta) \\ \vdots \\ \frac{\partial G}{\partial \delta_q}(\theta) \end{bmatrix} \begin{bmatrix} \frac{\partial G}{\partial \omega_0}(\theta) \\ \vdots \\ \frac{\partial G}{\partial \delta_q}(\theta) \end{bmatrix}^* \tag{47}$$

which generalizes the above $\frac{\partial f}{\partial \alpha}(\theta,\alpha)$, being the square length of the gradient of $G_{\omega,\delta}(\theta)$.

After some simple computations, $Tr(M(\theta))$ turns out to be

$$Tr(M(\theta)) = \frac{1}{|\delta(\theta)|^2}(p + 1 + q|G(\theta)|^2) \tag{48}$$

and, in the example of Box and Jenkins, $f_e(\theta) \propto \frac{1}{|\delta(\theta)|^2}$ so that, at least in this case

$$\frac{Tr(M(\theta))}{Var(\varepsilon_\theta)} \propto p + 1 + q|G(\theta)|^2 \tag{49}$$

and this is clearly to be considered in relation to

$$\sigma_y^2 = \sigma_x^2 \int_0^{2\pi} |G(\theta)|^2 f_x(\theta)\, d\theta + \sigma_e^2 \tag{50}$$

in order to understand that the optimal design problem should be related to the consideration of transfer of energy. Note that two different measures of dispersion are used here in order to make the point clear: one is the determinant (D-optimality) and one is the trace (length of the gradient). One could expect that using A-optimality (trace of the dispersion matrix) the results would be more coherent.

II. On the Shortcomings of "Optimal"

Dealing with multiparameter problems there are many different notions of "optimality", each one particularizing one aspect of the dispersion matrix. The present work was done with D-optimality, which corresponds to the determinant of this matrix: the theoretical results presented in [4] do not allow for a discussion of the overall quality of the so-called optimal designs but, as soon as one considers the applications, it is almost imperative to adopt a more flexible attitude and be ready to reject a D-optimal solution.

For an m-parameter problem, there are m basic quantities: the eigenvalues of the information matrix. When the computer program for the construction of D-optimal designs was written, it has been very easy to display, at each iteration, the eigenvalues of the information matrix, as well as different functions of these eigenvalues:

- Sum of the eigenvalues (trace of the matrix),
- Ratio of the largest and the smallest (measure of conditioning of the matrix),
- Sum of the inverses (trace of the dispersion matrix)

so that, when the program is run[1], this allows a greater flexibility in the decisions:

- Adopt the D-optimal design if it is a satisfactory solution,
- Adopt any intermediate design which could be considered as the best in the sequence $f_0, f_1, \ldots, f^*$ (see [5]),
- Try to change the choice for the first input f_0, so as to get a better sequence of designs,

[1] For details on the program, see Appendix I.

- Reconsider the identification of the model (see Box and Jenkins [1]).

III. Examples and General Considerations

At this point one might like to see how the D-optimal designs are constructed, and to have some illustrations of the different remarks.

III.1. A general example.

The model considered was

$$(1-.5B)y_t = x_{t-1} + .4x_{t-2} + .2x_{t-3} + a_t \tag{51}$$

with

$$Var(a_t) = 1.0$$

the choice for the first input was white noise, and the constraint was

$$C_3 = \sigma_x^2 \sigma_y^2 \leq 10.0 \tag{52}$$

The complete results are given in Figure 4 (Different values of $\Delta_{f_x}^3(\theta)$), Figure 5 (Half of the symmetric solution), Figure 6 (Successive values of the determinant) and Table 2 (Summary of the results).

Only 10 iterations were considered: it seems to be a general rule that, after five iterations, one has an almost D-optimal design

CRITERION

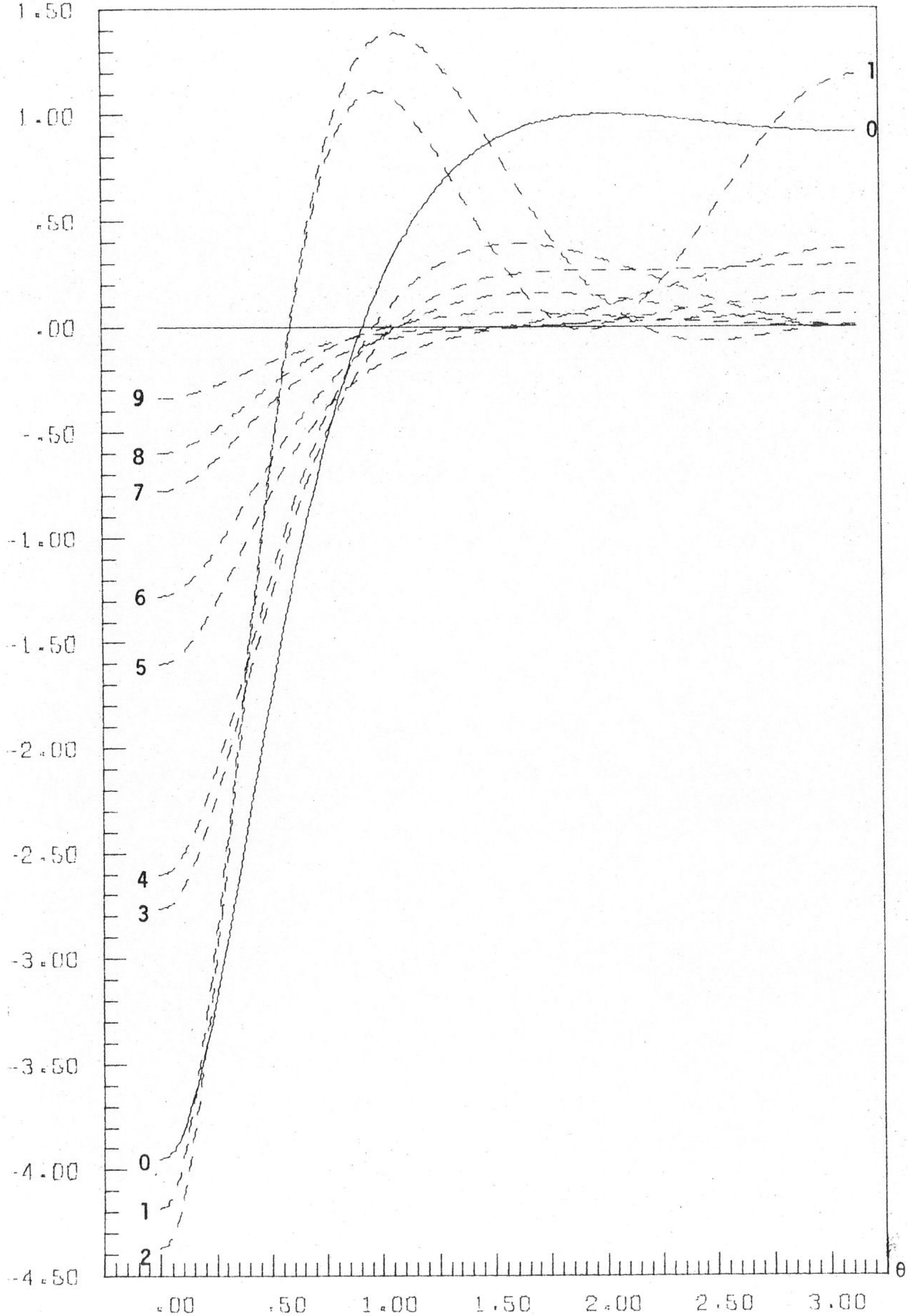

Figure 4. $\Delta^3_{f_x}(\theta)$ for the first 10 iterations.

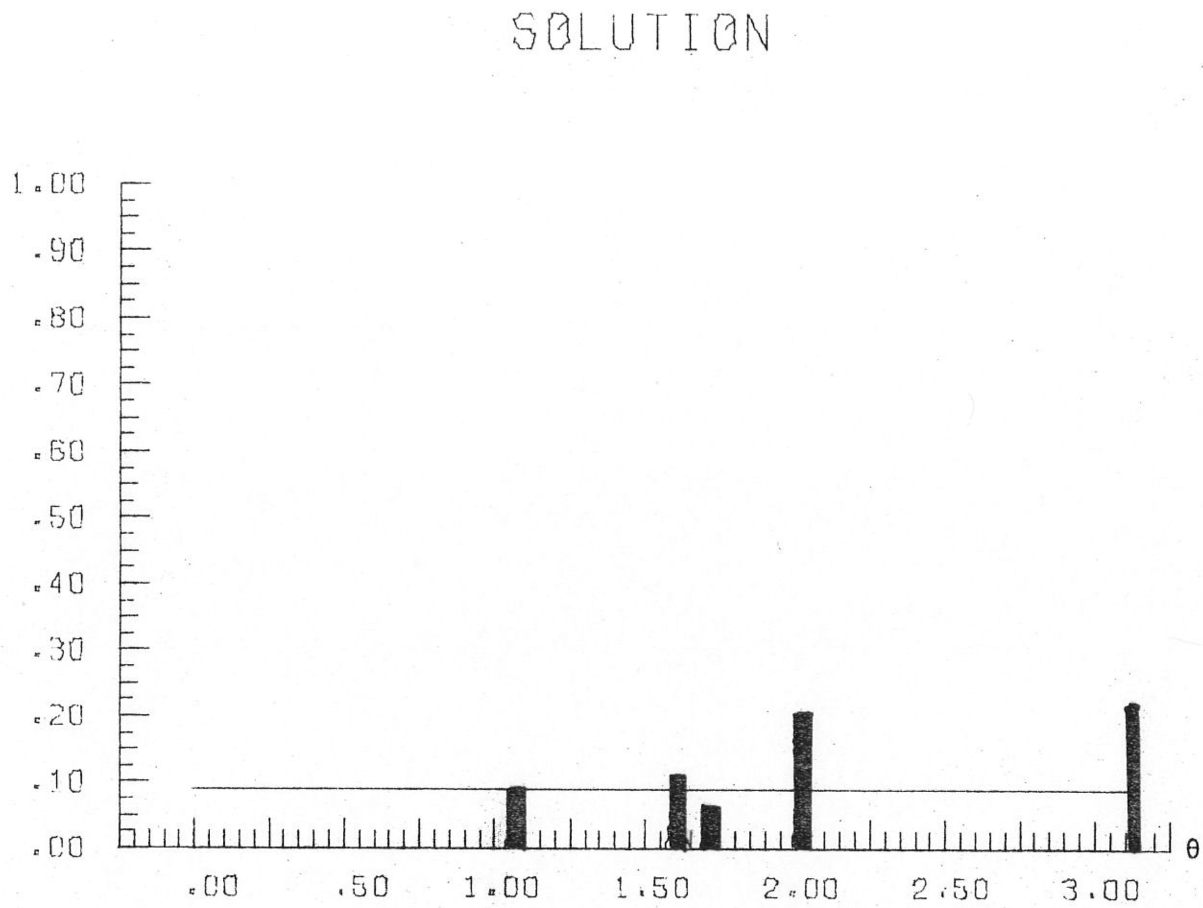

Figure 5. D-optimal (approximate) solution.

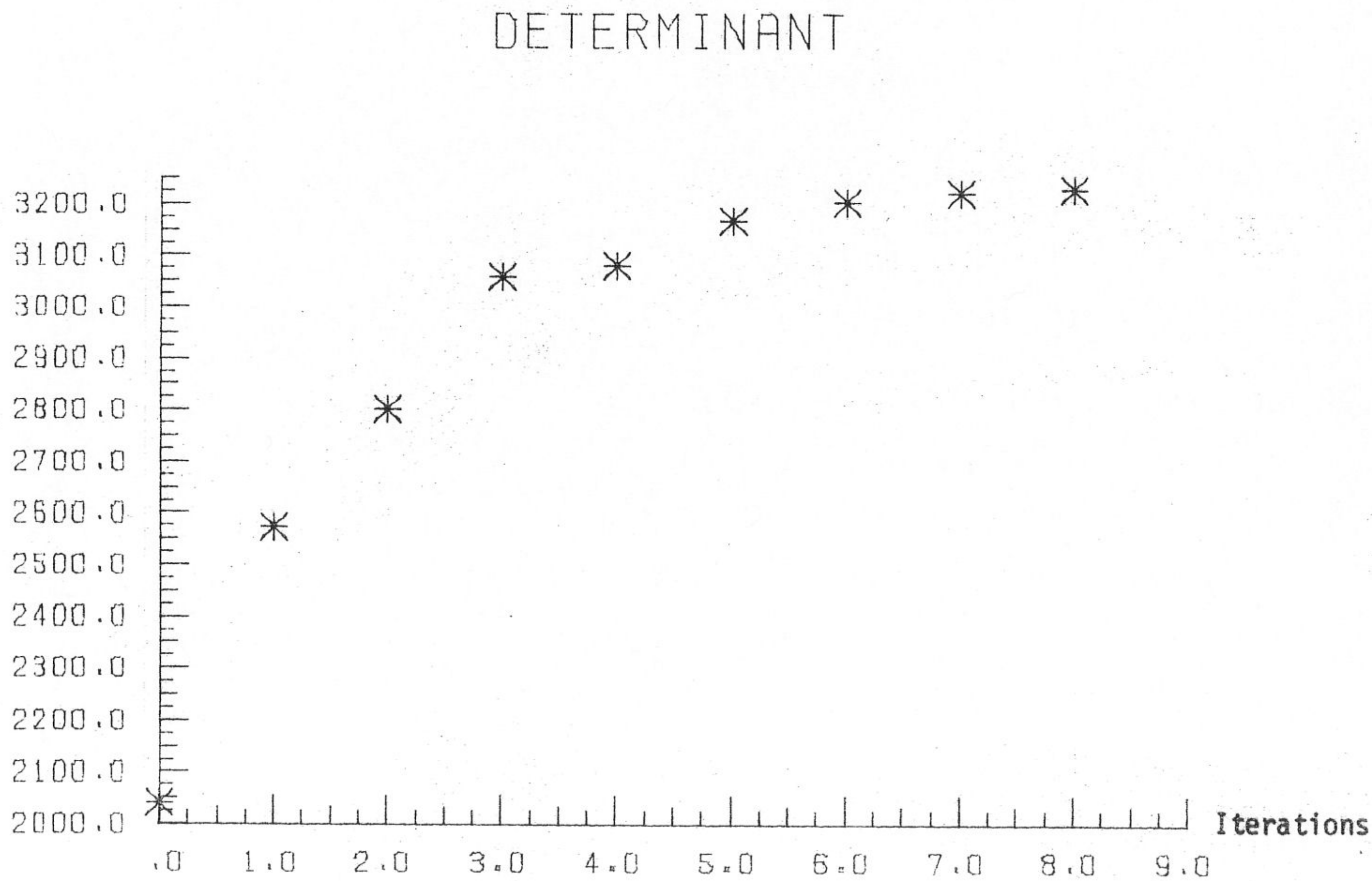

Figure 6. Successive values of the determinant.

Iteration	Information Matrix: Determinant	Largest Eigenvalue	Smallest Eigenvalue	Trace	Variances: White Noise	σ_x^2	σ_y^2
0	2041.	13.40	.74	22.5	.318	1.85	5.40
1	2571.	9.54	.54	18.9	.285	2.26	4.42
2	2799.	8.70	.51	18.1	.157	2.40	4.17
3	3057.	9.19	.61	18.4	.135	2.33	4.28
4	3081.	8.91	.58	18.2	.128	2.38	4.20
5	3168.	8.78	.58	17.7	.110	2.46	4.05
6	3203.	8.51	.55	17.5	.103	2.51	3.97
7	3221.	8.45	.54	17.3	.096	2.56	3.90
8	3227.	8.35	.53	17.2	.094	2.58	3.87
9	3230.	8.34	.53	17.1	.091	2.59	3.85
10	3231.	8.31	.52	17.1	.090	2.60	3.83

Table 2

and, for different practical reasons to be discussed later, there is no need for an exactly optimal design.

Even when θ is continuous it is convenient for the computations to consider it as a discrete variable: 113 equispaced values on $[0,\pi]$ were used here, providing what is believed to be an excellent approximation to the continuous case.

The solution on $[0,\pi]$ is composed of two parts:

- White noise with variance $.09\ \sigma_x^2$,
- Deterministic cosine waves as in Table 3:

Frequency (multiple of 1/224)	38	57	58	61	112
Amplitude (multiple of σ_x^2)	.093	.118	.025	.063	.219

Table 3

It is clear that this design will not be used, but instead the following:

- White noise with variance $.09\ \sigma_x^2$,
- Deterministic cosine waves as in Table 4:

Frequency (multiple of 1/224)	57	112
Amplitude (multiple of σ_x^2)	.30	.22

Table 4

i.e., the input

$$x_t = (.22\ (-1)^t + .30 \cos(\tfrac{57}{112}\ t) + .09\ \alpha_t)\sigma_x^2 \tag{53}$$

where $\{\alpha_t\}$ is white noise with variance 1.

The advantage, measured in terms of information, of (53) as compared to

$$x_t = 1.85\ \alpha_t \tag{54}$$

is approximately:

$$\sqrt{\frac{3231.}{2041.}} = 1.25 \tag{55}$$

III.2. Some applied considerations.

There has not been yet a comprehensive study of the D-optimal designs for dynamic models: the considerations to be presented now are just the results of a limited number of examples, and are therefore to be considered with caution.

As previously noted, it seems that the first iterations improve considerably the design: the number of iterations necessary to get a good approximation is of course a function of the number of parameters.

Very often the discrete masses in the solution have a tendency to cluster around different values: it may be realistic to concentrate them at these points, in order to get simple designs as was done in the preceding example (Fedorov [2]).

In all the examples considered, the information matrix turned out to be very poorly conditioned (λ_{max} very large as compared to λ_{min}), indicating a long thin ellipsoid of concentration. This is an unfavorable situation (see Section II) which requires further attention (see Viort [5], Chapter V).

References

[1] G. E. P. Box and G. M. Jenkins. Time Series Analysis: Forecasting and Control. Holden-Day, San Francisco (1970).

[2] V. V. Fedorov. Theory of Optimal Experiments. (Translated and edited by E. M. Klimko and W. J. Studden), (1969).

[3] G. M. Minnich. Some Considerations in the Choice of Experimental Designs for Dynamic Models. Ph.D. Dissertation, Department of Statistics, University of Wisconsin (1972).

[4] B. Viort. D-Optimal Designs for Dynamic Models. Part I. Theory. Technical Report Number 314, Department of Statistics, University of Wisconsin (1972).

[5] B. Viort. Design of Experiments and Dynamic Models. Ph.D. Dissertation, Department of Statistics, University of Wisconsin (1972).

Appendix I. A Computer Program for Construction and Discussion of D-Optimal Designs.

The program listed below was written for the UNIVAC 1108 of the University of Wisconsin MACC, but could be easily used on another computer. The external subroutines are:

- Numerical integration (NIROMB);
- Matrix operations: Scalar multiplication (MTSCP)
  Matrix addition (MTADD)
  Matrix Multiplication (MTMPY)
  Matrix inversion (MTINV)
  Trace (MITRCE)
  Eigenvalues (MTVLM)
  Determinant (MTDET);
- Plotting capabilities (GRAPH, GRAPHM).

The program allows for:

- Iterative construction of the almost D-optimal solution in a given situation, with or without plotting;
- Comparison of different situations (models or first choices for f_x), with or without plotting;
- Comparison of optimal solutions in different situations, without plotting.

The information to be provided is:

- On the first card (Format 6I2, I3), the parameters

  NIT = maximum number of iterations (≤ 20)

  NDIV = for printed output, to be explained below

NCURV = for plotted output: set NCURV = 1 if one wants all the curves plotted, set NCURV if one wants only one out of NCURV curves plotted (NCURV = NIT - 1 gives only the first and the last curve).

IGRAPH = 0 if no plotting is requested, 1 otherwise

ICOMP = 0 if no comparisons,

1 if comparison of first choices for different f_x or models (NIT = 1 in this case)

2 if comparison of optimal solutions (IGRAPH = 0 in this case)

NCOMP = number of comparisons ($1 \leq$ NCOMP ≤ 20).

NPTS = number of points in which the segment $[0,\pi]$ will be subdivided for the search of the maximum and the plotting (NPTS $\leq$ 131, suggested value 113).

- On the second card (Format 2F6.2, I2), the error variance S2A, the value of the constraint CONT, and the type of the constraint ICONT = 1, 2, or 3.
- The model, the error and the first choice for f_x are three modules on the same format:
  * Degree of the polynomials, Format 2I2 (IP, IQ)(IPE, IQE) or (IPX, IQX). IP + IQ should be less than 8.
  * Coefficients of the numerator, Format 9F6.3 (CP, CPE, or CPX) corresponding to CP(1) + CP(2)*B + CP(3)*B^2 + ...
  * Coefficients of the denominator, Format 9F6.3 (CQ, CQE, or CQX) corresponding to CQ(1) + CQ(2)*B + CQ(3)*B^2 + ... [Note the signs in this last case.]

In the program the position of the MODULE MODEL, MODULE ERROR, MODULE INPUT can be modified to allow different comparisons: all the information prior to

C END OF CONSTANTS

will be the same throughout the program, while the information just after

C DATA FOR COMPARISONS

will be changed according to the data cards.

The output consists in:

- Printed output:
  * Summary of the model,
  * Information matrix for the first input,
  * For each iteration:
    ** Variances σ_x^2, σ_y^2
    ** Information on $\Delta_f(\theta)$:
      Maximum,
      Values (one out of NDIV).
    ** Proportion of the first input kept in the solution
    ** Information on the iterated information matrix:
      Determinant
      Eigenvalues
      Trace
      Trace of the dispersion matrix
      Conditioning
  * At the end: detail of the discrete part of the solution.

- Plotted output.
    * $\Delta_f(\theta)$ (one out of NCURV)
    * The solution
    * Successive values of the determinant.

If no plotted output is requested, it is cheaper to compile the program without the Plotting Module.

```
'RUN ...
'MSG,   PLOT TAPE ON 27,11 INCH PAPER
'ASG,TL  27,T,PLOT

'FOR,SI MAIN
      DIMENSION ABZ(131),ORD1(131),ORD2(131),ORD3(131,20),ORD(131)
      DIMENSION ORDX1(131),ORDX2(131),XX(2),YY(2),DE(20),EVL(9)
      DIMENSION RM(9,9),RMP(9,9),RMIT(9,9),RA(9,9),RAP(9,9),RIN(9,9)
      DIMENSION CP(9),CQ(9),CPX(9),CQX(9),CPE(9),CQE(9)
      COMMON/E2/IPX,IQX,CPX,CQX,RKX/E3/IPE,IQE,CPE,CQE,RKE
      COMMON/E4/S2A,CONT,ICONT,SGXP/E5/K/E6/R/E7/IP,IQ,CP,CQ
      COMMON/E8/IP1,IQ1
      CALL MTADEF(RM,9,9,'S')
      CALL MTADEF(RA,9,9,'S')
      CALL MTADEF(RMP,9,9,'S')
      CALL MTADEF(RAP,9,9,'S')
      CALL MTADEF(RIN,9,9,'S')
      CALL MTADEF(RMIT,9,9,'S')
      PI=3.1416
      READ 1,NIT,NDIV,NCURV,IGRAPH,ICOMP,NCOMP,NPTS
      PRINT 6
      IF (ICOMP .EQ. 1) NIT=1
      IF (ICOMP .EQ. 2)IGRAPH=0
      RNPTS=NPTS-1
      PAS=PI/RNPTS
      CALL MTADEF(EVL,9,1,'S')
      IF (ICOMP .EQ. 0)NT=NIT
      IF (ICOMP .EQ. 1)NT=NCOMP
      IF (ICOMP .EQ. 2)NT=NIT
      READ 3,S2A,CONT,ICONT
      PRINT 3,S2A,CONT,ICONT
      PRINT 113
C MODULE MODEL
      DO 150 I=1,9
      CP(I)=.0
  150 CQ(I)=.0
      READ 1,IP,IQ
      IP1=IP+1
      IQ1=IQ+1
      M=IP+IQ+1
      M1=M-1
      RSM=M
      CALL MTMDEF(RM,M,M,'GEN')
      CALL MTMDEF(RA,M,M,'GEN')
      CALL MTMDEF(RMP,M,M,'GEN')
      CALL MTMDEF(RAP,M,M,'GEN')
      CALL MTMDEF(RIN,M,M,'GEN')
      CALL MTMDEF(RMIT,M,M,'GEN')
      READ 2,(CP(I),I=1,IP1)
      READ 2,(CQ(I),I=1,IQ1)
      PRINT 116
      PRINT 1,IP,IQ
      PRINT 2,(CP(I),I=1,IP1)
      PRINT 2,(CQ(I),I=1,IQ1)
C END MODULE MODEL
```

```
C MODULE ERROR
      DO 151 I=1,9
      CPE(I)=.0
  151 CQE(I)=.0
      READ 1,IPE,IQE
      IPE1=IPE+1
      IQE1=IQE+1
      READ 2,(CPE(I),I=1,IPE1)
      READ 2,(CQE(I),I=1,IQE1)
      PRINT 117
      PRINT 1,IPE,IQE
      PRINT 2,(CPE(I),I=1,IPE1)
      PRINT 2,(CQE(I),I=1,IQE1)
      EXTERNAL FE
      RKE=1.0
      CALL NIROMB(FE,.0,PI,.001 ,9,4,ROMB,SIMP,2,IERROR,DUM)
      RKE=1.0/ROMB
C END MODULE ERROR
C MODULE INPUT
      DO 152 I=1,9
      CPX(I)=.0
  152 CQX(I)=.0
      READ 1,IPX,IQX
      IPX1=IPX+1
      IQX1=IQX+1
      READ 2,(CPX(I),I=1,IPX1)
      READ 2,(CQX(I),I=1,IQX1)
      PRINT 118
      PRINT 1,IPX,IQX
      PRINT 2,(CPX(I),I=1,IPX1)
      PRINT 2,(CQX(I),I=1,IQX1)
      RKX=1.0
      EXTERNAL FX
      CALL NIROMB(FX,.0,PI,.001 ,9,4,ROMB,SIMP,2,IERROR,DUM)
      RKX=1.0/ROMB
C END MODULE INPUT
C END OF CONSTANTS
      NC=0
   19 NC=NC+1
      IF(NC .GT. NCOMP)GO TO 50
C DATA FOR COMPARAISONS
      EXTERNAL SUMMAT
      CALL SUMMAT(RM)
      CALL  MTINV(RM,RIN,1.E-6,$101)
      CALL MTVLM(RM,EVL)
      EXTERNAL GX
      CALL NIROMB(GX,.0,PI,.001 ,9,4,ROMB,SIMP,2,IERROR,DUM)
      SGX=ROMB
      SGXP=SGX
      CALL CONST(S2X,S2Y)
      CALL  MTDET(RM,DETMAX,DUM)
      DETMAX=DETMAX*(S2X**RSM)
```

```
      DO 12 I=1,NPTS
      RI=I-1
      T=RI*PAS
      ABZ(I)=T
      ORDX1(I)=FX(T)
   12 ORDX2(I)=.0
      RPA=1.0
      NI=0
   20 RMAX=.0
      NI=NI+1
      IF(ICOMP .EQ. 0)N=NI
      IF(ICOMP .EQ. 1)N=NC
      IF(ICOMP .EQ. 2)N=NI
      IMAX=0
      TMAX=.0
      DO 21 I=1,NPTS
      T=ABZ(I)
      EXTERNAL MAT
      CALL MAT(RA,T)
      CALL MTMPY(RIN,RA,RAP)
      CALL MTTRCE(RAP,TR)
      EXTERNAL G
      IF(ICONT .EQ. 1)SM=RSM
      IF(ICONT .EQ. 2)SM=RSM*G(T)/SGX
      IF(ICONT .EQ. 3)SM=RSM*(S2X*S2X*G(T)+CONT)/(S2X*S2X*SGX+CONT)
      ORD1(I)=TR
      ORD2(I)=SM
      ORD3(I,N)=ORD1(I)-ORD2(I)
      IF(ORD3(I,N) .LE. RMAX)GO TO 25
      RMAX=ORD3(I,N)
      TMAX=T
      IMAX=I
   25 CONTINUE
   21 CONTINUE
      N1=N-1
      PRINT 104,N1
      PRINT 119
      IF(ICOMP .EQ. 0)PRINT 105,RMAX,TMAX,IMAX
      PRINT 119
      PRINT 106,S2X,S2Y
      PRINT 119
      PRINT 4,(ORD3(I,N),I=1,NPTS,NDIV)
      PRINT 119
      PRINT 110,RPA
      PRINT 119
      PRINT 107,DETMAX
      PRINT 119
      PRINT 109
      PRINT 111,(EVL(I),I=1,M)
      TR=.0
      DO 30 I=1,M
   30 TR=TR+EVL(I)
      PRINT 112,TR
      TR=.0
      DO 31 I=1,M
   31 TR=TR+1.0/EVL(I)
```

```
      PRINT 115,TR
      TR=EVL(1)/EVL(M)
      PRINT 114,TR
      DE(N)=DETMAX
      IF(ICOMP .EQ. 1)GO TO 19
      IF(RMAX .LT. .01)GO TO 40
      IF(NI .GT. NIT)GO TO 40
      CALL MAT(RA,TMAX)
      A=.0
   32 A=A+.05
      IF(A .GE. 1.0)GO TO 102
      A1=1.0-A
      CALL MTSCP(RM,A1,'S',RMP)
      CALL MTSCP(RA,A,'S',RAP)
      CALL MTADD(RMP,RAP,RMIT)
      SGXP=SGX*A1+A*G(TMAX)
      CALL MTDET(RMIT,DET,DUM)
      CALL CONST(S2X,S2Y)
      DET=DET*(S2X**RSM)
      IF(DET .LE. DETMAX)GO TO 33
      DETMAX=DET
      GO TO 32
   33 DETMAX=DET
   34 A=A-.001
      IF(A .LE. .0)GO TO 103
      A1=1.0-A
      CALL MTSCP(RM,A1,'S',RMP)
      CALL MTSCP(RA,A,'S',RAP)
      CALL MTADD(RMP,RAP,RMIT)
      SGXP=SGX*A1+A*G(TMAX)
      CALL MTDET(RMIT,DET,DUM)
      CALL CONST(S2X,S2Y)
      DET=DET*(S2X**RSM)
      IF(DET .LE. DETMAX)GO TO 35
      DETMAX=DET
      GO TO 34
   35 A=A+.001
      A1=1.0-A
      RPA=RPA*A1
      CALL MTSCP(RM,A1,'S',RMP)
      CALL MTSCP(RA,A,'S',RAP)
      CALL MTADD(RMP,RAP,RM)
      CALL MTINV(RM,RII,1.E-6,$101)
      CALL MTVLM(RM,EVL)
      SGX=A1*SGX+A*G(TMAX)
      SGXP=SGX
      CALL CONST(S2X,S2Y)
      CALL MTDET(RM,DETMAX,DUM)
      DETMAX=DETMAX*(S2X**RSM)
      DO 36 I=1,NPTS
      ORDX1(I)=A1*ORDX1(I)
   36 ORDX2(I)=A1*ORDX2(I)
      ORDX2(IMAX)=ORDX2(IMAX)+A
      GO TO 20
```

```
  101 PRINT 9
      GO TO 19
  102 PRINT 8
      GO TO 19
  103 PRINT 7
      GO TO 19
   40 CONTINUE
      PRINT 108
      PRINT 4,(ORDX2(I),I=1,NPTS)
      PRINT 5,(DE(I),I=1,NT)
      GO TO 19
   50 CONTINUE
C POSITION MODULE PLOTTING
C MODULE PLOTTING
      IF (IGRAPH .EQ. 0) GO TO 51
      RMAX=.0
      RMIN=.0
      DO 41 I=1,NPTS
      DO 42 N=1,NT
      IF(ORD3(I,N) .GT. RMAX)RMAX=ORD3(I,N)
      IF(ORD3(I,N) .LT. RMIN)RMIN=ORD3(I,N)
   42 CONTINUE
   41 CONTINUE
      XX(1)=-.25
      XX(2)=PI
      YY(1)=RMIN
      YY(2)=RMAX
      CALL INITPL(27,10.8)
      CALL GRAPH(XX,6HLINEAR,YY,6HLINEAR,2,4HNONE,5HBLANK,'$$','$$',+1,
     1'CRITERION$$',8.5,11.)
      DO 46 I=1,NPTS
   46 ORD(I)=.0
      CALL GRAPHM(ABZ,5HSCAL1,ORD,5HSCAL1,NPTS,4HNONE,5HSOLID)
      DO 48 N=1,NT,NIT
      DO 43 I=1,NPTS
   43 ORD(I)=ORD3(I,N)
      CALL GRAPHM(ABZ,5HSCAL1,ORD,5HSCAL1,NPTS,4HNONE,5HSOLID)
   48 CONTINUE
      IF (ICOMP .EQ. 1)GO TO 49
      IF (NCURV .GE. NIT) GO TO 52
      J=NCURV+1
      DO 44 N=J,NIT,NCURV
      DO 45 I=1,NPTS
   45 ORD(I)=ORD3(I,N)
      CALL GRAPHM(ABZ,5HSCAL1,ORD,5HSCAL1,NPTS,4HNONE,1)
   44 CONTINUE
   52 CONTINUE
      YY(1)=.0
      YY(2)=1.0
      CALL GRAPH(XX,6HLINEAR,YY,6HLINEAR,2,4HNONE,5HBLANK,'$$',
     1'$$',0,'SOLUTION$$',5HTHALF,6HNORMAL)
      CALL GRAPHM(ABZ,5HSCAL1,ORDX2,5HSCAL1,NPTS,4HNONE,5HSOLID)
      CALL GRAPHM(ABZ,5HSCAL1,ORDX1,5HSCAL1,NPTS,4HNONE,5HSOLID)
   49 CONTINUE
```

```
      ABZ(1)=.0
      DO 47 N=2,NT
   47 ABZ(N)=ABZ(N-1)+1.
      CALL GRAPH(ABZ,6HLINEAR,DE,6HLINEAR, NT,11,5HBLANK,'$$','$$',0,
     1'DETERMINANT$$',5HBHALF,6HNORMAL)
      CALL ENDPLT
   51 CONTINUE
C END MODULE PLOTTING


    1 FORMAT(6I2,I3)
    2 FORMAT(9F6.3)
    3 FORMAT(2F6.2,I2)
    4 FORMAT(17F7.3)
    5 FORMAT(F15.1)
    6 FORMAT(1H1)
    7 FORMAT(11H ERROR A=0.)
    8 FORMAT(21H DESIGN ON ONE  POINT)
    9 FORMAT(21H ERROR IN INVERSION. )
  104 FORMAT(///,14H ITERATION NO.,I3)
  105 FORMAT(10H MAXIMUM =,F12.3,4H AT ,F8.3,I4)
  106 FORMAT(16H VARIANCES  S2X=,F8.3,6H  S2Y=,F8.3)
  107 FORMAT(21H VALUE OF DETERMINANT,F14.2)
  108 FORMAT(9H SOLUTION)
  109 FORMAT(19H INFORMATION MATRIX)
  110 FORMAT(30H PROPORTION OF ORIGINAL INPUT=,F10.3)
  111 FORMAT(13H EIGENVALUES=,9F10.3)
  112 FORMAT(7H TRACE=,F10.3)
  113 FORMAT(21H SUMMARY OF THE MODEL)
  114 FORMAT(14H CONDITIONING=,F10.3)
  115 FORMAT(13H TRACE D.MAT=,F10.3)
  116 FORMAT(14H MODULE MODEL )
  117 FORMAT(14H MODULE ERROR )
  119 FORMAT(1H )
  118 FORMAT(14H MODULE INPUT )
      END
```

```
'FOR,SI SUB1
      FUNCTION FX(X)
      DIMENSION CPX(9),CQX(9)
      COMMON/E2/IPX,IQX,CPX,CQX,RKX
      F11=CPX(1)
      F12=.0
      DO 1 I=1,IPX
      RI=I
      F11=F11+CPX(I+1)*COS(RI*X)
    1 F12=F12+CPX(I+1)*SIN(RI*X)
      F21=CQX(1)
      F22=.0
      DO 2 I=1,IQX
      RI=I
      F21=F21+CQX(I+1)*COS(RI*X)
    2 F22=F22+CQX(I+1)*SIN(RI*X)
      FX=RKX*(F11*F11+F12*F12)/(F21*F21+F22*F22)
      RETURN
      END

'FOR,SI SUB2
      FUNCTION FE(X)
      DIMENSION CPE(9),CQE(9)
      COMMON/E3/IPE,IQE,CPE,CQE,RKE
      F11=CPE(1)
      F12=.0
      DO 1 I=1,IPE
      RI=I
      F11=F11+CPE(I+1)*COS(RI*X)
    1 F12=F12+CPE(I+1)*SIN(RI*X)
      F21=CQE(1)
      F22=.0
      DO 2 I=1,IQE
      RI=I
      F21=F21+CQE(I+1)*COS(RI*X)
    2 F22=F22+CQE(I+1)*SIN(RI*X)
      FE=RKE*(F11*F11+F12*F12)/(F21*F21+F22*F22)
      RETURN
      END

'FOR,SI SUB3
      FUNCTION G(X)
      EXTERNAL COM
      CALL COM(O1,O2,OSQ,D1,D2,DSQ,X)
      G=OSQ/DSQ
      RETURN
      END
```

```
'FOR,SI SUB4
      SUBROUTINE CONST(S2X,S2Y)
      COMMON/E4/S2A,CONT,ICONT,SGXP
      IF(ICONT .EQ. 1)GO TO 1
      IF(ICONT .EQ. 2)GO TO 2
      IF(ICONT .EQ. 3)GO TO 3
    1 S2X=CONT
      S2Y=S2X*SGXP+S2A
      GO TO 4
    2 S2X=(CONT-S2A)/SGXP
      S2Y=CONT
      GO TO 4
    3 S2X=(-S2A+SQRT(S2A*S2A+4.0*SGXP*CONT))/(2.0*SGXP)
      S2Y=CONT/S2X
    4 CONTINUE
      RETURN
      END

'FOR,SI SUB5
      FUNCTION GX(X)
      EXTERNAL G
      EXTERNAL FX
      GX=G(X)*FX(X)
      RETURN
      END

'FOR,SI SUB6
      SUBROUTINE COM(O1,O2,OSQ,D1,D2,DSQ,T)
      DIMENSION CP(9),CQ(9)
      COMMON/E7/IP,IQ,CP,CQ
      O1=CP(1)
      O2=.0
      DO 1 I=1,IP
      RI=I
      O1=O1+CP(I+1)*COS(RI*T)
    1 O2=O2+CP(I+1)*SIN(RI*T)
      OSQ=O1*O1+O2*O2
      D1=CQ(1)
      D2=.0
      IF(IQ .EQ. 0)GO TO 3
      DO 2 I=1,IQ
      RI=I
      D1=D1+CQ(I+1)*COS(RI*T)
    2 D2=D2+CQ(I+1)*SIN(RI*T)
    3 CONTINUE
      DSQ=D1*D1+D2*D2
      RETURN
      END
```

```
'FOR,SI SUB7
      FUNCTION F1(X)
      COMMON/E6/R
      EXTERNAL COM
      CALL COM(O1,O2,OSQ,D1,D2,DSQ,X)
      F1    =COS(R*X)/DSQ
      RETURN
      END

'FOR,SI SUB8
      FUNCTION F2(X)
      COMMON/E6/R
      EXTERNAL COM
      CALL COM(O1,O2,OSQ,D1,D2,DSQ,X)
      F2    =COS(R*X)*OSQ/(DSQ*DSQ)
      RETURN
      END

'FOR,SI SUB9
      FUNCTION F3(X)
      COMMON/E6/R
      EXTERNAL COM
      CALL COM(O1,O2,OSQ,D1,D2,DSQ,X)
      F3    =-((O1*D1+O2*D2)*COS(R*X)+(O1*D2-O2*D1)*SIN(R*X))/(DSQ*DSQ)
      RETURN
      END

'FOR,SI SUB10
      FUNCTION FXE1(X)
      EXTERNAL F1
      EXTERNAL FX
      EXTERNAL FE
      FXE1=F1(X)*FX(X)/FE(X)
      RETURN
      END

'FOR,SI SUB11
      FUNCTION FXE2(X)
      EXTERNAL FE
      EXTERNAL FX
      EXTERNAL F2
      FXE2=F2(X)*FX(X)/FE(X)
      RETURN
      END
```

```
'FOR,SI SUB12
      FUNCTION FXE3(X)
      EXTERNAL FE
      EXTERNAL FX
      EXTERNAL F3
      FXE3=F3(X)*FX(X)/FE(X)
      RETURN
      END

'FOR,SI SUB13
      SUBROUTINE MAT(RA,T)
      DIMENSION RA(9,9)
      COMMON/E6/R/E8/IP1,IQ1
      IQ=IQ1-1
      M=IP1+IQ
      EXTERNAL FE
      DO 1 I=1,IP1
      R=I-1
      EXTERNAL F1
      FE1=F1(T)/FE(T)
      J1=IP1+1-I
      DO 2 J=1,J1
      RA(J,I+J-1)=FE1
    2 RA(I+J-1,J)=FE1
    1 CONTINUE
      IF(IQ .EQ. 0)GO TO 8
      DO 3 I=1,IP1
      DO 4 J=1,IQ
      R=J+1-I
      EXTERNAL F3
      FE3=F3(T)/FE(T)
      RA(I,IP1+J)=FE3
    4 RA(IP1+J,I)=FE3
    3 CONTINUE
      DO 5 I=1,IQ
      R=I-1
      EXTERNAL F2
      J1=IQ+1-I
      FE2=F2(T)/FE(T)
      DO 6 J=1,J1
      RA(IP1+J,IP1+J+I-1)=FE2
    6 RA(IP1+J+I-1,IP1+J)=FE2
    5 CONTINUE
    8 CONTINUE
      RETURN
      END
```

```
'FOR,SI SUB14
      SUBROUTINE SUMMAT(RM)
      DIMENSION RM(9,9)
      COMMON/E6/R/E8/IP1,IQ1
      IQ=IQ1-1
      M=IP1+IQ
      PI=3.1416
      DO 1 I=1,IP1
      R=I-1
      EXTERNAL FXE1
      CALL NIROMB(FXE1,.0,1.,.001,9,4,ROM1,SIMP,2,IERROR,DUM)
      CALL NIROMB(FXE1,1.,PI,.001,9,4,ROM2,SIMP,2,IERROR,DUM)
      ROMB=ROM1+ROM2
      J1=IP1+1-I
      DO 2 J=1,J1
      RM(J,I+J-1)=ROMB
    2 RM(I+J-1,J)=ROMB
    1 CONTINUE
      IF(IQ .EQ. 0)GO TO 8
      DO 3 I=1,IP1
      DO 4 J=1,IQ
      R=J+1-I
      EXTERNAL FXE3
      CALL NIROMB(FXE3,.0,1.,.001,9,4,ROM1,SIMP,2,IERROR,DUM)
      CALL NIROMB(FXE3,1.,PI,.001,9,4,ROM2,SIMP,2,IERROR,DUM)
      ROMB=ROM1+ROM2
      RM(I,IP1+J)=ROMB
    4 RM(IP1+J,I)=ROMB
    3 CONTINUE
      DO 5 I=1,IQ
      R=I-1
      EXTERNAL FXE2
      CALL NIROMB(FXE2,1.,PI,.001,9,4,ROM1,SIMP,2,IERROR,DUM)
      CALL NIROMB(FXE2,.0,1.,.001,9,4,ROM2,SIMP,2,IERROR,DUM)
      ROMB=ROM1+ROM2
      J1=IQ+1-I
      DO 6 J=1,J1
      RM(IP1+J,IP1+J+I-1)=ROMB
    6 RM(IP1+J+I-1,IP1+J)=ROMB
    5 CONTINUE
    8 CONTINUE
      PRINT 11
      DO 7 I=1,M
    7 PRINT 10,(RM(I,J),J=1,M)
   10 FORMAT(9F10.5)
   11 FORMAT(19H INFORMATION MATRIX)
      RETURN
      END

'XQT

  DATA CARDS.

'FIN
```

Unclassified

Security Classification

DOCUMENT CONTROL DATA – R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) University of Wisconsin Department of Statistics Madison, Wisconsin	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

D-Optimal Designs for Dynamic Models. Part II: Applications

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (First name, middle initial, last name)

Bernard Viort

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
November 1972	42	5

8a. CONTRACT OR GRANT NO. AFOSR-72-2363 b. PROJECT NO.	9a. ORIGINATOR'S REPORT NUMBER(S) Technical Report Number 316
c. d.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Air Force Office of Scientific Research 1400 Wilson Boulevard Arlington, Virginia

13. ABSTRACT

Different examples of constrained D-optimal designs for dynamic models are presented and discussed.

A computer program for the construction of such designs is listed.

DD FORM 1 NOV 65 1473

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
D-optimality						
Dynamic models						
Design of experiment						
Numerical methods						
Examples						