MECH577 Optimum Design

Lecture Notes

Jorge Angeles

Department of Mechanical Engineering &
Centre for Intelligent Machines
McGill University, Montreal (Quebec), Canada

September 2002

Chapter 1

Preliminaries and Single-Variable Optimization

1.1 Introduction

The English word *design* derives from the Latin word *designare*, which means "to mark out"—as found, for example, in the *Random College Dictionary*. The word thus implies a goal, an objective. As such, the meaning of the word is extremely broad, encompassing the general activity of producing concepts aimed at a given goal, be this pure intellectual pleasure, in the realm of art, or pragmatic, in the realm of engineering.

The product of the design activity is a good, whether tangible, e.g., a fountain pen, in the realm of *industrial design*, or intangible, e.g., a business plan, in the realm of *management*. We focus here on *engineering design*, but this does not mean that we exclude intangible goods. An important branch of engineering is *production systems*, whereby the design good is many a time intangible, such as the *organization* of a healthcare system.

In mechanical engineering, more specifically, engineering design is aimed at the production of mechanical systems or components thereof. Optimization finds its place in the design process within the dashed box of Fig. 1.1. From this figure, it is apparent that optimization falls in the realm of *synthesis*, i.e., of production of the embodiment of the *preliminary design*. However, synthesis is not a one-way street, and hence, a feedback loop is formed as the designer resorts to *analysis* to evaluate the design performance.

Engineering design problems have increasingly become model-based, in that their

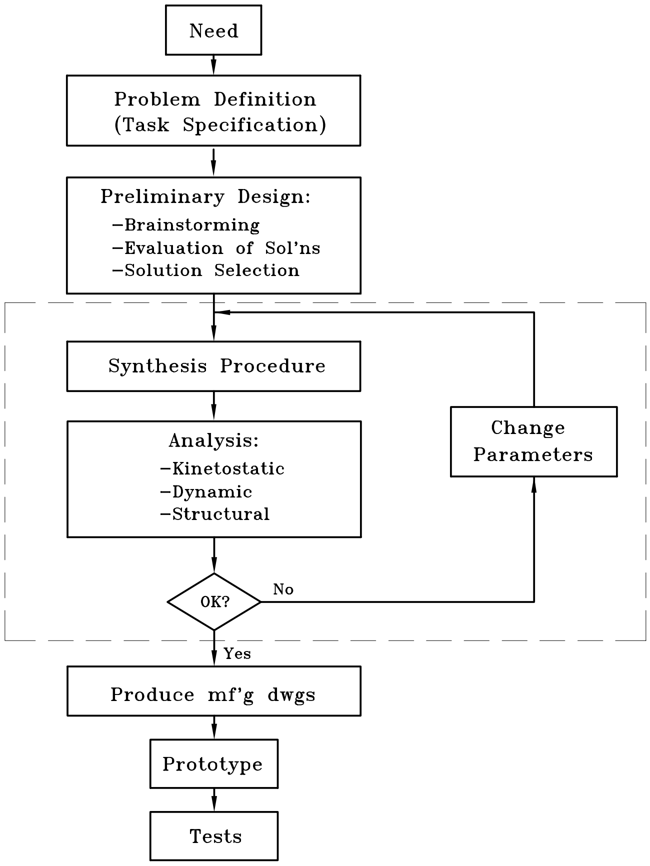

Figure 1.1: The role of optimization within the design process

complexity calls for mathematical models involving many quantities, some of which are to be decided on by the designer with the purpose of meeting performance specifications—e.g., the thrust that an aircraft engine must deliver at a given rpm—under given environment conditions—engine must operate at a specified ambient temperature and at a given ambient pressure. We thus classify the various quantities occurring in the model into:

- *Design Variables:* Those quantities that the designer has to find so as to produce the specified performance under the given conditions;
- *Design-Environment Parameters:* Those quantities over which the designer has no control, and that define the conditions under which the designed object must operate; and
- *Performance Functions:* Relations representing the performance of the design in terms of design variables and design-environment parameters.

Henceforth we shall denote by $\mathbf{x}$ the n-dimensional vector of design variables; we shall refer to this vector, consequently, as the *design-variable vector (DVV)*. Likewise, we shall denote by $\mathbf{p}$ the ν-dimensional vector of design-environment parameters, which will be termed the *design-environment-parameter vector (DEPV)*, while the μ performance functions, $f_i = f_i(\mathbf{x};\, \mathbf{p})$, for $i = 1, 2, \ldots, \mu$ are grouped in the *design-performance vector (DPV)* $\mathbf{f}$. We thus have

$$\mathbf{x} \equiv \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{p} \equiv \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_\nu \end{bmatrix}, \quad \mathbf{f} \equiv \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_\mu \end{bmatrix}, \quad \mathbf{f} = \mathbf{f}(\mathbf{x};\, \mathbf{p}) \tag{1.1}$$

Needless to say, the DVV and the DEPV being independent from each other, their dimensions are also independent, and hence, $n \neq \nu$, in general. By the same token, the number μ of performance functions is independent from n and ν.

1.1.1 Example 1: The Design of an Automatic Screw Driver

Shown in Fig. 1.2 is a model of an innovative automatic screw driver (Becker, 1996). The purpose of this *electromechanical system* is to *automatically* insert screws either in wood or in masonry, as fed by a mechanism and presented to the bit. As the axis of one screw is aligned with that of the bit, the bit is advanced by the motor driving the rack-and-pinion until it engages the head of the screw. With bit tip and screw head engaged, the second motor drives the bit-screw ensemble towards the surface of the object into which the screw will be inserted. The use of two motors, whose motion is coordinated and regulated by a chip fed with signals coming from mechanical and electronic sensors, is the result of *invention*.

The geometry of the cross section of the bit is hexagonal with a height of $2d$. Moreover, the cross section is made hollow with a circular hole of radius r, to reduce weight, as shown in Fig. 1.3. The reason behind the hexagonal shape of the bit is the need to both transmit torque to it and allow for the relative sliding of the bit with respect to the gear driving it. The discussion process leading to the forgoing geometry belongs to the realm of *preliminary design*. The material of the bit is steel, with Young modulus $E = 200$ GPa, yield stress $\sigma_Y = 500$ MPa, and density $\rho = 7\ 850$ kg/m^3.

During the screw-insertion process, the bit is subjected to an axial load F of 4 760 N and a torque T of 4 Nm. The problem is to find the optimum dimensions of the bit for minimum weight, subject to constraints on buckling, yield, and geometry

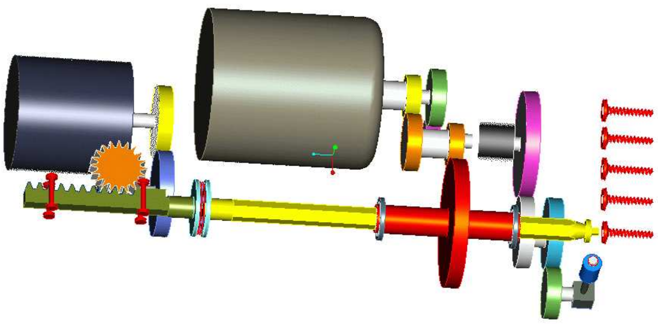

Figure 1.2: The transmission of an automatic screw driver

requirements. We designate by L the bit length, its cross-section area by A, which is given by

$$A = 2\sqrt{3}d^2 - \pi r^2$$

The weight W is given, in turn, by

$$W = \rho g A L = \rho g(2\sqrt{3}d^2 - \pi r^2)L \qquad (1.2)$$

where g is the gravity-acceleration constant. Furthermore, we notice that by virtue of the axial force F, the bit risks failure under buckling, for which reason we recall Euler's formula for the critical load P_{cr} under which a clamped-free column will buckle, namely (Gere and Timoshenko, 1991),

$$P_{cr} = \frac{2.046\pi^2 EI}{L^2} \qquad (1.3)$$

where I is the area moment of inertia of the cross section with respect to the z-axis[1]. From Fig. 1.3, I turns out to be

$$I = \frac{5\sqrt{3}}{9}d^4 - \frac{\pi}{4}r^4$$

Moreover, the combined load of force and torque is bound to produce a general state of stress in the bit, which can produce failure if the "stress level" becomes

[1] As a matter of fact, the cross section being axially symmetric, its two principal moments of inertia are identical, which means that I is also the area moment of inertia with respect to the x axis.

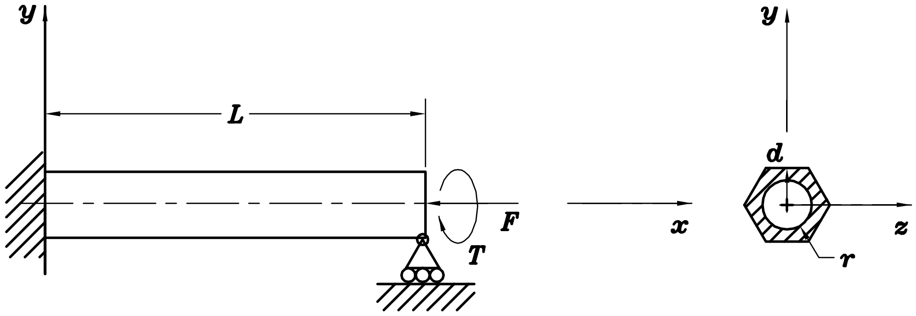

Figure 1.3: Loading and support conditions on the bit

higher than the yield stress. The stress level, moreover, can be defined in several ways, but is based on the model provided by the *theory of failure* adopted. Although several theories of failure are available, the one based on the von Mises stress σ_{vM} is the most appropriate, for it relies on the *volumetric strain energy* (Norton, 2000), and is given by the second invariant of the *stress deviator*, obtained upon subtracting the hydrostatic component from the stress tensor (Chou and Pagano, 1992). In order to derive a simple expression for the von Mises stress, we introduce a simplifying assumption: the shear stress produced by the torque T is the same as that occurring in a hollow shaft of circular cross section[2], the normal and shear stresses thus being

$$\sigma_{xx} = \frac{F}{A}, \quad \sigma_{yy} = \sigma_{zz} = 0$$
$$\tau = \sqrt{\tau_{xy}^2 + \tau_{zx}^2} = \frac{Td}{I_p}, \quad \tau_{yz} = 0$$

and I_p defined as the area polar moment of inertia of the cross section, which, by virtue of its symmetry—the two principal moments of inertia of the cross section are identical—yields

$$I_p = 2I = \frac{10\sqrt{3}}{9}d^4 - \frac{\pi}{2}r^4$$

Then, σ_{vM} is given, under the above conditions, as

$$\sigma_{vM} = \sqrt{\frac{F^2}{A^2} + \frac{3T^2d^2}{I_p^2}} \tag{1.4}$$

The design-performance functions are, apparently, the weight W, the critical buckling load P_{cr} and the the von Mises stress σ_{vM}, given in eqs.(1.2), (1.3) and (1.4), respectively.

[2]To account for this assumption, we must introduce a safety factor.

We have thus described a design problem with DVV $\mathbf{x} = [d,\ r,\ L]^T$, and design performance functions W, P_{cr} and σ_{vM}. In this example, then, the DEP are the applied force F and the applied torque T. Hence, $n = 3$, $\nu = 2$ and $\mu = 3$.

1.1.2 Example 2: The Design of a Low-Pass Filter

We consider here the design of the RL circuit shown in Fig. 1.4, first used by Taguchi (1988) to illustrate the concept of *robust design*, and then by Wilde (1992). The design variables are the resistance R and the inductance L, to be determined by the designer, Moreover, the excitation voltage $v(t)$ is given by $v(t) = V_o \cos \omega t$. The voltage amplitude V_o and its frequency ω undergo variations beyond the control of the designer.

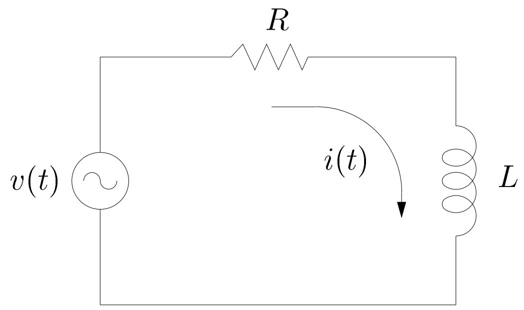

Figure 1.4: A low-pass filter

For this filter, the steady-state current $i(t)$ is harmonic: $i(t) = I_o \cos(\omega t + \phi)$, where I_o and ϕ are the magnitude and the phase of $i(t)$. These are given by

$$I_o = \frac{V_o}{\sqrt{R^2 + \omega^2 L^2}} \tag{1.5}$$

$$\phi = \tan^{-1}\left(\frac{\omega L}{R}\right) \tag{1.6}$$

Furthermore, V_0 is 110 volt, while ω is 60 Hz, as provided by a power utility, the filter being designed for an amplitude I_0 of the current of 10 ampere. The vectors of DV, DEP and DPF are, thus,

$$\mathbf{x} = \begin{bmatrix} R \\ L \end{bmatrix}, \quad \mathbf{p} = \begin{bmatrix} V_o \\ \omega \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} I_o \\ \phi \end{bmatrix}$$

Apparently, then, in this design problem we have $n = 2$, $\nu = 2$ and $\mu = 2$.

1.2 The Structure of Optimum Design Problems

In optimum design problems, the DEP are given by the client either explicitly or implicitly. The designer then assumes that these bear *ideal* values that are *representative* of the operation conditions, and do not change. The fact of the matter is that the DEP entail values that are random and hence, are not known but through their statistics, such as mean values and standard deviations; when the parameters obey a Gaussian distribution, mean value and standard derivation are sufficient to describe the variation of the DEP. In this case, then, the designer uses the mean values of the DEP as the *nominal values* of these parameters, on which the design is based.

The foregoing approach is classical, and will be followed here. A modern approach, due to Taguchi (1988), consists in admitting that the DEP are not fixed, but varying in a random manner, beyond the control of the designer, the purpose of the design task, then, being to select the design variables in such a way that, under arbitrary variations of the DEP within a certain range, the DPF exhibit "small" variations. This approach is known as *robust design*. For an introductory course, we will not dwell on this second approach, but will illustrate its main underlying ideas with some examples of optimum design.

In an optimum design problem, then, the designer chooses either one or several DPF and formulates an *objective function* to be either minimized, when this function represents a cost, or maximized, when the same represents a profit. As a matter of fact, profit-maximization can be readily turned into cost-minimization if the profit is redefined as a cost by, for example, reversing its sign or taking its reciprocal. Moreover, a large class of optimum design problems lends itself to a *least-square* formulation, which inherently aims at *minimizing* a sum of squares. For these reasons, and without loss of generality, we will aim in this course at the *minimization* of an objective function $f(\mathbf{x})$. Defining the objective function is thus a simple matter when only one of the DPF is of interest. In defining objective functions, however, as in many design tasks, it will prove convenient to use dimensionless quantities. In this vein, we replace d, r and L in Example 1 by the nondimensional variables x_1, x_2, and x_3, defined as

$$x_1 = \frac{d}{L_0}, \quad x_2 = \frac{r}{L_0}, \quad x_3 = \frac{L}{L_0} \tag{1.7}$$

where L_0 can be defined, for example, as a bound for L. Now, if all we want to achieve in Example 1 is to minimize weight, then the objective function is simply

$$f(\mathbf{x}) = W = \rho g(2\sqrt{3}\,x_1^2 - \pi x_2^2)x_3 \tag{1.8}$$

in which the constant factor can be dispensed with, the objective function then simplifying to

$$f = (2\sqrt{3}\,x_1^2 - \pi x_2^2)x_3 \tag{1.9}$$

If we want both to minimize weight and maximize the critical load of the same example, then we must first rewrite each of these two DPF in nondimensional form. Let us redefine $f(\mathbf{x})$, as defined in eq.(1.9), as $f_1(\mathbf{x})$, a second nondimensional objective function being derived from the reciprocal of P_{cr}—upon minimizing $1/P_{cr}$ we will maximize P_{cr} itself—which, apparently, is proportional to L^2/I. Upon introducing the foregoing dimensionless DV into this quotient, we notice that this quotient is proportional to $f_2(\mathbf{x})$, defined as

$$f_2(\mathbf{x}) = \frac{1}{(5\sqrt{3}/9)x_1^4 - (\pi/4)x_2^4} \tag{1.10}$$

Now we must decide what *relative importance* to assign to each of the two foregoing functions, and assign, correspondingly, weighting factors w_1 and w_2 to them, so that we obtain now a combined objective function

$$f(\mathbf{x}) = w_1 f_1(\mathbf{x}) + w_2 f_2(\mathbf{x}) \tag{1.11}$$

However, the formulation of the optimum design problem does not end with defining the objective function. We still must incorporate into the formulation the *constraints* under which every designer must work. To be true, sometimes an optimum design task lends itself to an unconstrained formulation, but this is rather the exception than the rule. In general, a designer works under physical and budgetary constraints: the laws of Newton must be obeyed in mechanical design; the laws of Kirchhoff must be obeyed in electrical design; a fixed budget ($) has been allocated to the design project; the design project must be finished by a certain date; etc.

1.2.1 Example 3: The Optimum Design of an Automatic Screw Driver

In Example 1, the bit

1. Should be able to withstand the axial force without buckling;

2. should be able to withstand the combined effect of the axial force and the torque without failing;

3. should have a minimum thickness of 20.0 mm to prevent high stress concentrations that can lead to failure;

4. should allow for space to accommodate its mechanical transmission and the motor, without interference, which means that its length should be at least of 76.2 mm; and

5. should be machinable at *a reasonable cost*, which means that its bore should have a radius not smaller than 6.0 mm.

Of the above constraints, items 4 and 5 were supplied by the client, while the first three items were introduced by the designer, based on both knowledge of strength of materials and experience.

The constraints in every design are formulated either as equations or as inequalities. Accordingly, the constraints are termed either *equality constraints* or *inequality constraints*. As to the latter, they can be expressed in one of two forms, namely,

$$g(\mathbf{x}) \leq 0 \quad \text{or} \quad g(\mathbf{x}) \geq 0 \tag{1.12}$$

In this course we will consistently use the first form. Now, in formulating the five foregoing constriants as "≤ 0" inequalities, we need to introduce *safety factors* S_b and S_y, associated with buckling and yield, respectively, the constraints now taking the form

$$\text{Buckling}: \; F - \frac{P_{cr}}{S_b} \leq 0 \tag{1.13a}$$

$$\text{Yield}: \; S_y \sigma_{vM} - \sigma_Y \leq 0 \tag{1.13b}$$

$$\text{Cross sectionthickness}: \; r - d + 0.002 \leq 0 \tag{1.13c}$$

$$\text{Length of bit}: \; 0.0762 - L \leq 0 \tag{1.13d}$$

$$\text{Radius of the circular bore}: \; 0.006 - r \leq 0 \tag{1.13e}$$

whence $L_0 = 0.0762$ m.

By substituting the expressions for P_{cr} and σ_{vM} into relations (1.13a & b), these constraints become, more explicitly,

$$L^2 - \frac{2.046\pi^2}{S_b F} EI \leq 0 \tag{1.14a}$$

$$I_p^2 + \frac{3T^2 d^2 A^2}{F^2} - \frac{1}{S_y^2 F^2} \sigma_Y^2 I_p^2 A^2 \leq 0 \tag{1.14b}$$

The five constraints thus can be expressed as

$$g_1(\mathbf{x}) \equiv L_0^2\left[x_3^2 - \frac{2.046\pi^2 L_0^2 E}{S_b F}\left(\frac{5\sqrt{3}}{9}x_1^4 - \frac{\pi}{4}x_2^4\right)\right] \leq 0 \tag{1.15a}$$

$$g_2(\mathbf{x}) \equiv L_0^8\left\{\left(\frac{10\sqrt{3}}{9}x_1^4 - \frac{\pi}{2}x_2^4\right)^2\left[1 - \frac{\sigma_Y^2 L_0^4}{S_y^2 F^2}(2\sqrt{3}x_1^2 - \pi x_2^2)^2\right] + \frac{3T^2}{F^2 L_0^2}(2\sqrt{3}x_1^2 - \pi x_2^2)^2 x_1^2\right\} \leq 0 \tag{1.15b}$$

$$g_3(\mathbf{x}) \equiv L_0\left(x_2 - x_1 + \frac{0.002}{L_0}\right) \leq 0 \tag{1.15c}$$

$$g_4(\mathbf{x}) \equiv L_0(1 - x_3) \leq 0 \tag{1.15d}$$

$$g_5(\mathbf{x}) \equiv L_0\left(\frac{0.006}{L_0} - x_2\right) \leq 0 \tag{1.15e}$$

$$\tag{1.15f}$$

In summary, the optimum design problem is formulated as

$$f(\mathbf{x}) \equiv (2\sqrt{3}\,x_1^2 - \pi x_2^2)x_3 \quad \rightarrow \min_{\mathbf{x}} \tag{1.16a}$$

subject to

$$g_1(\mathbf{x}) \equiv x_3^2 - \frac{2.046\pi^2 L_0^2 E}{s_b F}\left(\frac{5\sqrt{3}}{9}x_1^4 - \frac{\pi}{4}x_2^4\right) \leq 0 \tag{1.16b}$$

$$g_2(\mathbf{x}) \equiv \left(\frac{10\sqrt{3}}{9}x_1^4 - \frac{\pi}{2}x_2^4\right)^2\left[1 - \frac{\sigma_Y^2 L_0^4}{S_y^2 F^2}(2\sqrt{3}x_1^2 - \pi x_2^2)^2\right] + \frac{3T^2}{F^2 L_0^2}(2\sqrt{3}x_1^2 - \pi x_2^2)^2 x_1^2 \leq 0 \tag{1.16c}$$

$$g_3(\mathbf{x}) \equiv x_2 - x_1 + \frac{0.002}{L_0} \leq 0 \tag{1.16d}$$

$$g_4(\mathbf{x}) \equiv 1 - x_3 \leq 0 \tag{1.16e}$$

$$g_5(\mathbf{x}) \equiv \frac{0.006}{L_0} - x_2 \leq 0 \tag{1.16f}$$

In compact form, the inequality constraints are usually expresses as

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \quad \mathbf{g}(\mathbf{x}) \equiv [g_1(\mathbf{x}) \ \ldots \ g_5(\mathbf{x})]^T \tag{1.17}$$

The reader should handle the above vector inequality with care for, properly speaking, vector inequalities are maningless, the reason being that *arrays do not form ordered sets*!

1.3 Methods of Single-Variable Optimization

While real-life design problems involve multiple variables, some techniques developed to find the optimum of these problems rely on a search along each of the variables at a time. Moreover, the designer in many instances is interested in the role played by one single variable, in which case the search for the optimum value can be conducted with techniques specific to this case. For this reason, it is convenient to study techniques applicable to the solution of single-variable optimization problems, which is the subject of the balance of this chapter. We start by introducing a **definition:**

> *A function $f(x)$ is* unimodal *in the interval* $[\,0,\,1\,]$ *if it attains one single extremum—a minimum or a maximum—within this interval.*

Remarks:

- We will deal only with function minimization—function maximization can be handled as a minimization problem by either simply reversing the sign of the objective function or using the reciprocal of the original function as the objective function to minimize.
- A unimodal function need neither be continuous nor smooth.
- Defining the interval of interest as $[\,0,\,1\,]$ is not restrictive. If this interval is $[\,a,\,b\,]$, where a and b are any real numbers, then a simple linear transformation of the variable in question can lead to the above interval.

We introduce, moreover, the **basic assumption:** Function $f(x)$, to be minimized, is *unimodal* in the interval $[\,0,\,1\,]$, which means that $f(x)$ attains exactly one minimum (or one maximum) in the given interval.

As a consequence of the above definition, we have

Lemma 1.3.1 *Let $f(x)$ be unimodal in $[\,0,\,1\,]$ and attain a minimum within this interval. Then, its maximum lies necessarily at the* extremes *of the interval, i.e., either at $x = 0$ or at $x = 1$.*

The proof of this lemma is left to the reader as an exercise. Moreover, note that:

- The objective function can be evaluated *only at a discrete, finite set of sample values* of its argument x, $\{\,x_i\,\}_1^n$: Each function evaluation, $f(x_i) \equiv f_i$, is termed an *experiment*. The name is quite appropriate because in some instances it may happen that the evaluation of function $f(x)$ can be done only by physical experiments, e.g., when this function is the steady-state temperature of an engine, that is known to change as the proportion of a mixture of fuel and air varies;

- We assume that the interval in which the minimum lies is *known*, and termed the *interval of uncertainty* of the problem at hand. Upon a suitable transformation of the design variable, this interval is mapped into the *normal interval* $[\,0,\,1\,]$, which is of unit length. The length of the interval of uncertainty when the series of experiments is initiated is thus 1, the purpose of the minimization exercise being to bring down the interval of uncertainty to an acceptable low, which is dictated mostly by the cost of each experiment;

- If the computational cost of each experiment is not an issue, then the function can be evaluated in a rich sample of argument values within the interval $[\,0,\,1\,]$ and plot the corresponding values; the optimum can then be located by inspection, possibly at the click of a mouse. This is termed an *exhaustive search*.

- If the computational cost is high, then proceed *iteratively*: At each iteration, the interval of uncertainty is cut by a certain factor using a suitable *strategy*, i.e., a search method.

- Any strategy exploits the unimodality assumption. We can cite four strategies that are the most commonly employed:

  - Dichotomous search
  - Interval-halving
  - Fibonacci numbers
  - Golden search

We will study here only three strategies: dichotomous search; Fibonacci numbers; and golden search.

1.4 Dichotomous Search

The qualifier "dichotomous" derives from Greek, meaning *to cut into two parts*. The strategy to follow thus consists in splitting the interval into two subintervals, not necessarily of the same length, with one not containing the minimum, and is hence, rejected; the other subinterval then is bound to contain the minimum sought.

The search strategy of this method is described below:

- Assume that, at iteration i, the current search interval is $\mathcal{I}_i = [\,l,\, r\,]$, of length $L_i = r - l < 1$;

- locate **two** points of abscissae x_1 and x_2 around the centre of the interval: For a "small" $\delta > 0$,

$$x_1 \equiv \frac{1-\delta}{2}, \quad x_2 \equiv \frac{1+\delta}{2}; \quad f_i \equiv f(x_i); \quad f_1 \neq f_2$$

- if $f_2 > f_1$, then eliminate the interval segment to the right of x_2, the new search interval being $[\,l,\, x_2\,]$. If, on the contrary, $f_1 > f_2$, then eliminate the interval segment to the left of x_1, the new search interval being $[\,x_1,\, r\,]$

- New search interval $\mathcal{I}_{i+1}$ is of length $L_{i+1} = (L_i + \delta)/2$, i.e., slightly over one half the length of the previous one.

Now we determine the length L_{2k} of interval $\mathcal{I}_{2k}$ after $2k$ experiments—this number is always an even number! To this end, we notice how the length of the interval of uncertainty (i.o.u.) evolves as the search progresses:

$$\begin{aligned}
L_2 &= \frac{1}{2} + \frac{\delta}{2} \\
L_4 &= \frac{L_2}{2} + \frac{\delta}{2} = \frac{1}{4} + \frac{\delta}{4} + \frac{\delta}{2} \\
&= \frac{1}{4} + \frac{3\delta}{4} \\
L_6 &= \frac{L_4}{2} + \frac{\delta}{2} = \frac{1}{8} + \frac{3\delta}{8} + \frac{\delta}{2} \\
&= \frac{1}{8} + \frac{7\delta}{8} \\
&\vdots \\
L_{2k} &= \frac{1}{2^k} + \left(1 - \frac{1}{2^k}\right)\delta
\end{aligned}$$

The length of the interval after $2k$ experiments is thus

$$L_{2k} = \frac{1}{2^k} + \delta\left(1 - \frac{1}{2^k}\right) \tag{1.18}$$

Usually, L_{2k} is prescribed, but k is not. Computing k from L_{2k} is, nevertheless, straightforward, as described below: Solving eq.(1.18) for 2^k yields

$$2^k = \frac{1-\delta}{L_{2k}-\delta} \tag{1.19}$$

and hence,

$$k = \lceil \frac{\ln[(1-\delta)/(L_{2k}-\delta)]}{\ln(2)} \rceil \tag{1.20}$$

where $\lceil(\cdot)\rceil$ is the *ceiling function*, defining the smallest natural number that is greater than the real argument $(\cdot)$.

1.4.1 Example: Finding the Maximum Dexterity Posture of a Two-Joint Robotic Manipulator

In optimizing the performance of robotic manipulators one is interested in maximizing their *dexterity*, a performance index that comes into play as explained below.

In robot control, a velocity $\mathbf{v}$ of the *operation point* of the end link is to be produced by a suitable set of joint rates, grouped in vector $\dot{\mathbf{q}}$, the relation between the two vectors being linear: $\mathbf{J}\dot{\mathbf{q}} = \mathbf{v}$. Hence, the *Jacobian matrix* $\mathbf{J}(\mathbf{q})$ must be inverted in order to compute the joint-rate vector, for a given posture of the manipulator, as specified by vector $\mathbf{q}$, and a given desired velocity $\mathbf{v}$. Dexterity measures, essentially, how *invertible* the Jacobian matrix is; dexterity can be quantified by means of the product $\mathbf{J}\mathbf{J}^T = \ell^2\mathbf{K}$, where ℓ is the length of the proximal (the longer) link, and $\mathbf{K}$ is given by

$$\mathbf{K} \equiv \begin{bmatrix} 3+2\sqrt{2}\cos\theta & 1+\sqrt{2}\cos\theta \\ 1+\sqrt{2}\cos\theta & 1 \end{bmatrix}$$

where θ is the angle made by the axis of the distal link with the extension of that of the proximal. It should be apparent that, when $\theta = 0$ or π, matrix $\mathbf{K}$, that we shall term here the *dexterity matrix*, is singular, and hence, not invertible, as is $\mathbf{J}$. Between these two values, 0 and π, there is one specific value θ_o optimum, at which the dexterity matrix is *maximally invertible*. To find θ_o, we start by defining the dexterity as the ratio of the smallest (λ_m) to the largest (λ_M) eigenvalues of $\mathbf{K}$. In this regard, note that $\mathbf{K}$ is symmetric, and hence, its eigenvalues are real. Moreover, verify that $\mathbf{K}$ is positive-definite, and becomes singular only for the two values of θ given above. We thus have the dexterity function $D(\theta)$ defined below:

$$D(\theta) = \frac{\lambda_m}{\lambda_M} \geq 0, \quad 0 \leq D(\theta) \leq 1$$

Now, maximizing $D(\theta)$ is equivalent to minimizing $f(\theta) \equiv 1/D(\theta)$, which will be defined as the *objective function* of the problem at hand. Given the form of the objective function, then, each experiment involves three steps:

1. For a given value of θ, compute the two eigenvalues of $\mathbf{K}$, a task that can be readily implemented using an eigenvalue routine, a quadratic-equation solver, or even the Mohr circle (Norton, 2000).

2. Order the two eigenvalues in ascending order: λ_m, λ_M.

3. Compute $f(\theta)$ as
$$f(\theta) = \frac{\lambda_M}{\lambda_m} \quad 0 \leq f(\theta) \leq 1$$
An interpretation of $f(\theta)$ can be obtained if we define the *loss of dexterity* $L(\theta)$ in the form
$$L(\theta) \equiv [1 - f(\theta)] \times 100\ (\%), \quad 0 \leq \theta < 100\%$$
and hence, when $D(\theta)$ attains its maximum value of unity, $L(\theta)$ attains its minimum of 0. At the other side of the spectrum, when the manipulator is postured at a singularity, $L(\theta)$ becomes infinitely large, indicating that the manipulator has lost all its dexterity.

An expert roboticist claims that the dexterity is maximum—the robot is at the peak of its positioning accuracy—when θ lies "somewhere between 90° and 150°." Find an estimate of θ_o within an interval of uncertainty of 5% of the given interval length of 60°.
Solution: We implemented the dichotomous search in the Maple worksheet described below, which is posted in the course Web page.

```
> restart:withlinalg;
```

$$withlinalg$$

```
> with(linalg):

Warning, the protected names norm and trace have been redefined and
unprotected
```

We start by producing a procedure K that will allow us to evaluate matrix $\mathbf{K}$ for a given value θ:

```
> K:=proc(theta) matrix([[3+2*sqrt(2)*cos(theta),
> 1+sqrt(2)*cos(theta)], [1+sqrt(2)*cos(theta), 1]]) end;
```

$K :=$ **proc**(θ)
  matrix$([[3 + 2 \times \mathrm{sqrt}(2) \times \cos(\theta),\ 1 + \mathrm{sqrt}(2) \times \cos(\theta)],\ [1 + \mathrm{sqrt}(2) \times \cos(\theta),\ 1]])$
end

```
> argu:= 3*Pi/4; K(argu);#testing procedure, which should yield the
> 2 by 2 identity matrix for this value of argument theta
```

$$argu := \frac{3}{4}\pi$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Apparently, procedure is OK.

We introduce now a transformation that maps θ into the normal interval $[\,0,\,1\,]$. Let the associated "normal" variable be x, to be produced by a second procedure x. By the same token, we need a third procedure θ to return the angle in radians, for a given value of x. Thus,

```
> x:=proc(theta) (theta - Pi/2)/(5*Pi/6 - Pi/2) end;
```

$$x := \textbf{proc}(\theta)\, 3 \times (\theta - 1/2 \times \pi)/\pi\ \textbf{end}$$

```
> theta:=proc(x) (Pi/3)*x + Pi/2 end;
```

$$\theta := \textbf{proc}(x)\, 1/3 \times \pi \times x + 1/2 \times \pi\ \textbf{end}$$

Now we determine the number $2k$ of experiments needed to attain the prescribed length of the i.o.u. We recall that the length L_{2k} of this interval is given by

$$L_{2k} = \frac{1}{2^k} + \delta(1 - \frac{1}{2^k}) \tag{1.21}$$

```
> L_2k:= (1/2^k) + delta*(1 - (1/2^k));
```

$$L\_2k := \frac{1}{2^k} + \delta\,(1 - \frac{1}{2^k})$$

Let $2^k = N$. Then,

```
> N:=solve(L_N= (1/N) + delta*(1 - 1/N), N);
```

$$N := \frac{-1+\delta}{-L\_N+\delta}$$

```
> delta:=0.01; L_N:= 0.05;#we want the length
> of the final i.o.u.  to be 5% of original length
```

$$\delta := .01$$

$$L\_N := .05$$

```
> N:=subs((delta=0.01, L_N=0.05), N);
```

$$N := 24.75000000$$

Hence,

```
> k:= solve(2^k = N, k); k:= ceil(k);
```

$$k := 4.629356620$$

$$k := 5$$

where the Maple `ceil(·)` command has been used. We thus need $2k = 10$ experiments. Hence, the two points x_1 and x_2 within I_0 are defined as

```
> x1:= (1-delta)/2; x2:=(1+delta)/2;
```

$$x1 := .4950000000$$

$$x2 := .5050000000$$

```
> th1:=evalf(theta(x1)); K1:=K(th1); th2:=evalf(theta(x2));
> K2:=K(th2);
```

$$th1 := 2.089159115$$

$$K1 := \begin{bmatrix} 3 - .9909173374\sqrt{2} & 1 - .4954586687\sqrt{2} \\ 1 - .4954586687\sqrt{2} & 1 \end{bmatrix}$$

$$th2 := 2.099631090$$

$$K2 := \begin{bmatrix} 3 - 1.009055247\sqrt{2} & 1 - .5045276237\sqrt{2} \\ 1 - .5045276237\sqrt{2} & 1 \end{bmatrix}$$

```
> lambda:=<eigenvals(K1)>; mu:=<eigenvals(K2)>;
```

$$\lambda := \text{RTABLE}(24331224, \begin{bmatrix} .8760194061 \\ 1.722611857 \end{bmatrix})$$

$$\mu := \text{RTABLE}(24331264, \begin{bmatrix} .8813318770 \\ 1.691648508 \end{bmatrix})$$

```
> f[1]:= lambda[2]/lambda[1]; f[2]:= mu[2]/mu[1];
```

$$f_1 := 1.966408330$$

$$f_2 := 1.919422810$$

$f_1 > f_2 \Rightarrow$ delete subinterval $[\,0,\, x_1\,]$. Let l and r denote, respectively, the abscissae of the left and right ends of the new subinterval:

```
> l:=x1; r:= 1; L:= r - l;
```

$$l := .4950000000$$

$$r := 1$$

$$L := .5050000000$$

where L is the length of i.o.u. at the end of the first two experiments. Carry on:

```
> x1:= (l+r-delta)/2; x2:= (l+r+delta)/2;
```

$$x1 := .7425000000$$

$$x2 := .7525000000$$

```
> th1:=evalf(theta(x1)); K1:=K(th1); th2:=evalf(theta(x2));
> K2:=K(th2);
```

$$th1 := 2.348340509$$

$$K1 := \begin{bmatrix} 3 - 1.403062852\sqrt{2} & 1 - .7015314261\sqrt{2} \\ 1 - .7015314261\sqrt{2} & 1 \end{bmatrix}$$

$$th2 := 2.358812484$$

$$K2 := \begin{bmatrix} 3 - 1.417911114\sqrt{2} & 1 - .7089555570\sqrt{2} \\ 1 - .7089555570\sqrt{2} & 1 \end{bmatrix}$$

```
> lambda:=<eigenvals(K1)>; mu:=<eigenvals(K2)>;
```

$$\lambda := \text{RTABLE}(24331304, \begin{bmatrix} .9967340325 \\ 1.019035453 \end{bmatrix})$$

$$\mu := \text{RTABLE}(24331384, \begin{bmatrix} .9936878850 \\ 1.001082988 \end{bmatrix})$$

```
> f[1]:= lambda[2]/lambda[1]; f[2]:= mu[2]/mu[1];
```

$$f_1 := 1.022374495$$
$$f_2 := 1.007442078$$

$f_1 > f_2 \Rightarrow$ delete subinterval $[\, l,\, x_1\,]$. Redefine l and r:

```
> l:=x1; L:= r - l;# r remains unchanged
```

$$l := .7425000000$$
$$L := .2575000000$$

where L is length of i.o.u. at the end of 3rd and 4th experiments. Carry on:

```
> x1:= (l+r-delta)/2; x2:= (l+r+delta)/2;
```

$$x1 := .8662500000$$
$$x2 := .8762500000$$

```
> th1:=evalf(theta(x1)); K1:=K(th1); th2:=evalf(theta(x2));
> K2:=K(th2);
```

$$th1 := 2.477931206$$
$$K1 := \begin{bmatrix} 3 - 1.575484092\,\sqrt{2} & 1 - .7877420459\,\sqrt{2} \\ 1 - .7877420459\,\sqrt{2} & 1 \end{bmatrix}$$
$$th2 := 2.488403181$$
$$K2 := \begin{bmatrix} 3 - 1.588299054\,\sqrt{2} & 1 - .7941495272\,\sqrt{2} \\ 1 - .7941495272\,\sqrt{2} & 1 \end{bmatrix}$$

```
> lambda:=<eigenvals(K1)>; mu:=<eigenvals(K2)>;
```

$$\lambda := \text{RTABLE}(24331464, \begin{bmatrix} .7246939855 \\ 1.047235044 \end{bmatrix})$$
$$\mu := \text{RTABLE}(24331544, \begin{bmatrix} .7028174767 \\ 1.050988460 \end{bmatrix})$$

```
> f[1]:= lambda[2]/lambda[1]; f[2]:= mu[2]/mu[1];
```

$$f_1 := 1.445072079$$
$$f_2 := 1.495393178$$

The ensuing computations follow the same pattern. In the interest of brevity, we record here only the last two experiments: At the end of 7th and 8th experiments, we have

$$r := .8143750000$$
$$L := .0718750000$$

```
> x1:= (l+r-delta)/2; x2:= (l+r+delta)/2;
```

$$x1 := .7734375000$$
$$x2 := .7834375000$$

```
> th1:=evalf(theta(x1)); K1:=K(th1); th2:=evalf(theta(x2));
> K2:=K(th2);
```

$$th1 := 2.380738183$$
$$K1 := \begin{bmatrix} 3 - 1.448494166\sqrt{2} & 1 - .7242470831\sqrt{2} \\ 1 - .7242470831\sqrt{2} & 1 \end{bmatrix}$$
$$th2 := 2.391210159$$
$$K2 := \begin{bmatrix} 3 - 1.462856184\sqrt{2} & 1 - .7314280922\sqrt{2} \\ 1 - .7314280922\sqrt{2} & 1 \end{bmatrix}$$

```
> lambda:=<eigenvals(K1)>; mu:=<eigenvals(K2)>;
```

$$\lambda := \mathrm{RTABLE}(24331704, \begin{bmatrix} .9414793498 \\ 1.010040556 \end{bmatrix})$$
$$\mu := \mathrm{RTABLE}(24331744, \begin{bmatrix} .9169618507 \\ 1.014247094 \end{bmatrix})$$

```
> f[1]:= lambda[2]/lambda[1]; f[2]:= mu[2]/mu[1];
```

$$f_1 := 1.072822847$$
$$f_2 := 1.106095192$$

$f_2 > f_1 \Rightarrow$ delete subinterval $[\,x_2,\, r\,]$. Redefine l and r:

```
> r:= x2; L:= r - l;# l remains unchanged
```

$$r := .7834375000$$
$$L := .0409375000$$

where L is length of i.o.u. at the end of the 9th and 10th experiments. Since L is smaller than 0.05, we're done. The best estimate of θ_{opt} is obviously the mid point of current i.o.u., i.e.,

```
> x_opt:= (l+r)/2; th:= evalf(theta(x_opt));
```

$$x\_opt := .7629687500$$
$$th := 2.369775334$$

```
> K_opt:=K(th);
```

$$K\_opt := \begin{bmatrix} 3 - 1.433288769\sqrt{2} & 1 - .7166443846\sqrt{2} \\ 1 - .7166443846\sqrt{2} & 1 \end{bmatrix}$$

```
> lambda:=<eigenvals(K_opt)>;
```

$$\lambda := \text{RTABLE}(24331824, \begin{bmatrix} .9674365862 \\ 1.005586999 \end{bmatrix})$$

```
> f[o]:= lambda[2]/lambda[1];
```

$$f_o := 1.039434536$$

Note that the exact optimum value of theta is $135°$, i.e., $3\pi/4 = 2.3562$ rad, which yields a value of $f = 1.0$.

1.5 Fibonacci Numbers

Fibonacci numbers are named after the Italian mathematician Leonardo Pisano (1175), son of Gulielmo Bonaccio, and hence, referred to as *filius Bonacci*. These numbers form a *sequence*, defined *recursively* as

$$F_0 = F_1 = 1 \tag{1.22a}$$
$$F_k = F_{k-2} + F_{k-1} \tag{1.22b}$$

Remark: The sequence is *monotonically increasing*, for all numbers are positive integers and the current one equals the sum of the two previous ones. Hence,

$$F_k - F_{k-1} = F_{k-2} \tag{1.23}$$

As well, since the Fibonacci series is monotonically increasing,

$$F_{k-2} < F_{k-1} \quad \text{or} \quad F_{k-1} > F_{k-2} \tag{1.24}$$

Upon addition of eqs.(1.23) and (1.24), we obtain

$$F_k - F_{k-1} + F_{k-1} > 2F_{k-2}$$

i.e.,

$$\frac{F_{k-2}}{F_k} < \frac{1}{2} \tag{1.25}$$

Furthermore, from eq.(1.22b),

$$\frac{F_{k-1}}{F_k} = 1 - \frac{F_{k-2}}{F_k} \tag{1.26}$$

Now we outline the strategy to follow in this method:

- Let $\mathcal{I}_0 \equiv [\,0,\,1\,]$ be the initial interval of uncertainty, of length 1, where the minimum is *known* to lie.

- **Prescribe** the number n of experiments to be conducted

- **Define** a length L_2^* as[3]

$$L_2^* \equiv \frac{F_{n-2}}{F_n} L_0 \equiv \frac{F_{n-2}}{F_n} \tag{1.27}$$

- **Let** $P_1(x_1)$ and $P_2(x_2)$ be two points *equidistant* from the left and the right ends of $\mathcal{I}_0$, respectively, by a distance L_2^*, i.e., with abscissae

$$x_1 = 0 + L_2^* = \frac{F_{n-2}}{F_n} \tag{1.28a}$$

$$x_2 = 1 - L_2^* = 1 - \frac{F_{n-2}}{F_n} \tag{1.28b}$$

Note that

$$x_2 = 1 - \frac{F_{n-2}}{F_n} = \frac{F_n - F_{n-2}}{F_n} = \frac{F_{n-1}}{F_n} \tag{1.28c}$$

Use the unimodality assumption to eliminate the segment of $\mathcal{I}_0$ where the minimum **cannot** lie. Whether the eliminated segment is at the right or at the left end of $\mathcal{I}_0$, the length L_2 of the new, shorter interval, $\mathcal{I}_2$—again, such as we do not define L_1^*, we neither define $\mathcal{I}_1$—is given by

$$L_2 = 1 - L_2^* = 1 - \frac{F_{n-2}}{F_n} = \frac{F_n - F_{n-2}}{F_n} = \frac{F_{n-1}}{F_n} < 1 \tag{1.29}$$

[3]No length L_1^* is defined because we want to make the subscript of L^* match that of the corresponding Fibonacci number; since $F_1 = F_0$, the first two Fibonacci numbers are undistinguishable.

Also, by virtue of relations (1.25) and (1.27),

$$\text{For} \quad n \geq 2, \quad L_n^* < \frac{1}{2} \tag{1.30}$$

Once the first iteration is completed,

- Let l and $r > l$ denote the abscissae of the left and the right ends of the current, smaller interval of uncertainty $\mathcal{I}_2$. The abscissa of one of the ends of $\mathcal{I}_2$, left or right, is either x_1 or x_2;

- if x_1 is the abscissa of one of the ends of $\mathcal{I}_2$, then $x_2 \in \mathcal{I}_2$; else, $x_1 \in \mathcal{I}_2$. Let x_I be the abscissa of point P_I, the interior point of $\mathcal{I}_2$;

- note that P_I lies a distance L_2^* from one of the ends of $\mathcal{I}_2$. Now, define $x_3 \in \mathcal{I}_2$ so that its associated point P_3 also lies a distance L_2^* from the other end;

- the process is continued until the interval $\mathcal{I}_n$, of length L_n, is obtained. L_n is the length of the final interval of uncertainty.

The abscissa x_j computed at the jth experiment is determined by length L_j^*, so that its associated point, P_j, as well as the interior point P_I of interval $\mathcal{I}_{j-1}$ are a distance L_j^* from the ends of $\mathcal{I}_j$, with L_j^* given by—see eq.(1.27)

$$L_j^* = \frac{F_{n-j}}{F_{n-(j-2)}} L_{j-1} \tag{1.31a}$$

while the length L_j of the jth interval of uncertainty is—see eq.(1.29)

$$L_j = \frac{F_{n-(j-1)}}{F_n} \tag{1.31b}$$

Hence, for $j = n$,

$$L_n = \frac{F_1}{F_n} \tag{1.32}$$

which allows us to find n for a prescribed length L_n.

1.5.1 The Location of the Final Experiment

Let $\mathcal{I}_{n-1} = [\,l,\, r\,]$, of length L_{n-1}, be the one-before-the-last interval. According with eqs.(1.28a & b), the abscissae of the last two experiments, x_{n-1} and x_n, are given as

$$x_{n-1} = l + L_n^* \tag{1.33a}$$

$$x_n = r - L_n^* = l + L_{n-1} - L_n^* \tag{1.33b}$$

where L_n^* is given by eq.(1.31a), with $j = n$:

$$L_n^* = \frac{F_0}{F_2} L_{n-1} \equiv \frac{1}{2} L_{n-1} \tag{1.34}$$

Upon substitution of L_n^*, as given by eq.(1.34), into eq.(1.33b), it is apparent that

$$x_n = l + \frac{1}{2} L_{n-1} = x_{n-1}$$

x_{n-1} and x_n thus coinciding, and hence, the last experiment fails to produce two distinct points in $\mathcal{I}_{n-1}$. To cope with this outcome, we have to define points P_1 and P_2, of abscissae x_1 and x_2, in a suitable manner. For example, we can define them as in the strategy employed by the dichotomous search, with a δ small enough with respect to L_n^*.

Fibonacci numbers are tabulated in many manuals, with short tables available in textbooks (Rao, 1996). Also note that scientific software is provided with Fibonacci numbers. For example, Maple includes the command

```
with(combinat, Fibonacci):
```

that allows the user to invoke the Fibonacci number `F(i)` by typing

```
fibonacci(i)
```

However, note that **not all Fibonacci sequences are identical**. For example, The first two Fibonacci numbers in Maple are defined as

`f(0)` = `0` and `f(1)` = `1`

1.5.2 Example: Finding the Maximum Dexterity Posture of a Two-Joint Robotic Manipulator

We implement the Fibonacci search in a Maple worksheet:

```
> restart:withlinalg;
```

$$withlinalg$$

```
> with(linalg):

Warning, the protected names norm and trace have been redefined and
unprotected
```

```
> with(combinat, fibonacci):
```

```
Warning, the name fibonacci has been redefined
```

We retake the example problem consisting in the finding of the most dexterous posture of a two-axis robot, using exactly **10 experiments.** However, because of the way Maple defines the Fibonacci sequence, we must use $F(n+1)$ when we would normally use $F(n)$.

We shall use K, θ and x exactly as described in Subsection 1.4.1.

We want to have

$$L_2^* = \frac{F_{10-2}}{F_{10}}, \tag{1.35}$$

but must shift the subscript by 1:

```
> Lstar[2]:=evalf(fibonacci(9)/fibonacci(11));
```

$$Lstar_2 := .3820224719$$

```
> l:=0; r:=1; L[0]:=r - l;# Abscissae of extremes of left- & right-hand
> sides of the initial (normal) interval, and length of this interval
```

$$l := 0$$

$$r := 1$$

$$L_0 := 1$$

```
> x1:= l + Lstar[2]; x2:= r - Lstar[2];
```

$$x1 := .3820224719$$

$$x2 := .6179775281$$

```
> th1:=evalf(theta(x1)); K1:=K(th1); th2:=evalf(theta(x2));
> K2:=K(th2);
```

$$th1 := 1.970849324$$

$$K1 := \begin{bmatrix} 3 - .7789343108\sqrt{2} & 1 - .3894671554\sqrt{2} \\ 1 - .3894671554\sqrt{2} & 1 \end{bmatrix}$$

$$th2 := 2.217940881$$

$$K2 := \begin{bmatrix} 3 - 1.205821535\sqrt{2} & 1 - .6029107675\sqrt{2} \\ 1 - .6029107675\sqrt{2} & 1 \end{bmatrix}$$

```
> lambda:=<eigenvals(K1)>; mu:=<eigenvals(K2)>;
```

$$\lambda := \mathrm{RTABLE}(7311224, \begin{bmatrix} .8139310151 \\ 2.084489519 \end{bmatrix})$$

$$\mu := \mathrm{RTABLE}(7647492, \begin{bmatrix} .9389633883 \\ 1.355747444 \end{bmatrix})$$

```
> f[1]:=lambda[2]/lambda[1]; f[2]:=mu[2]/mu[1];
```

$$f_1 := 2.561014976$$

$$f_2 := 1.443876791$$

$f_1 > f_2 \Rightarrow$ drop left end:

```
> l:= x1; L[2]:=L[0] - Lstar[2]; Lstar[3]:=
> (fibonacci(8)/fibonacci(10))*L[2];
```

$$l := .3820224719$$

$$L_2 := .6179775281$$

$$Lstar_3 := .2359550562$$

```
> x1:= l + Lstar[3]; x2:= r - Lstar[3];
```

$$x1 := .6179775281$$

$$x2 := .7640449438$$

```
> th1:=evalf(theta(x1)); K1:=K(th1); th2:=evalf(theta(x2));
> K2:=K(th2);
```

$$th1 := 2.217940881$$

$$K1 := \begin{bmatrix} 3 - 1.205821535\sqrt{2} & 1 - .6029107675\sqrt{2} \\ 1 - .6029107675\sqrt{2} & 1 \end{bmatrix}$$

$$th2 := 2.370902321$$

$$K2 := \begin{bmatrix} 3 - 1.434859867\sqrt{2} & 1 - .7174299337\sqrt{2} \\ 1 - .7174299337\sqrt{2} & 1 \end{bmatrix}$$

```
> lambda:=<eigenvals(K1)>; mu:=<eigenvals(K2)>;
```

$$\lambda := \mathrm{RTABLE}(7521524, \begin{bmatrix} .9389633883 \\ 1.355747444 \end{bmatrix})$$

$$\mu := \mathrm{RTABLE}(7648172, \begin{bmatrix} .9647545542 \\ 1.006047163 \end{bmatrix})$$

```
> f[1]:=lambda[2]/lambda[1]; f[2]:=mu[2]/mu[1];
```

$$f_1 := 1.443876791$$

$$f_2 := 1.042801155$$

$f_1 > f_2 \Rightarrow$ drop left end:

```
> l:= x1; L[3]:= L[2] - Lstar[3]; Lstar[4]:=
> (fibonacci(7)/fibonacci(9))*L[3];
```

$$l := .6179775281$$

$$L_3 := .3820224719$$

$$Lstar_4 := .1460674157$$

```
> x1:= l + Lstar[4]; x2:= r - Lstar[4];
```

$$x1 := .7640449438$$

$$x2 := .8539325843$$

```
> th1:=evalf(theta(x1)); K1:=K(th1); th2:=evalf(theta(x2));
> K2:=K(th2);
```

$$th1 := 2.370902321$$

$$K1 := \begin{bmatrix} 3 - 1.434859867\sqrt{2} & 1 - .7174299337\sqrt{2} \\ 1 - .7174299337\sqrt{2} & 1 \end{bmatrix}$$

$$th2 := 2.465032438$$

$$K2 := \begin{bmatrix} 3 - 1.559462054\sqrt{2} & 1 - .7797310268\sqrt{2} \\ 1 - .7797310268\sqrt{2} & 1 \end{bmatrix}$$

```
> lambda:=<eigenvals(K1)>; mu:=<eigenvals(K2)>;
```

$$\lambda := \mathrm{RTABLE}(7738916, \begin{bmatrix} .9647545542 \\ 1.006047163 \end{bmatrix})$$

$$\mu := \mathrm{RTABLE}(7648852, \begin{bmatrix} .7520453159 \\ 1.042542298 \end{bmatrix})$$

```
> f[1]:=lambda[2]/lambda[1]; f[2]:=mu[2]/mu[1];
```

$$f_1 := 1.042801155$$

$$f_2 := 1.386275901$$

```
> x1:= l + Lstar[8]; x2:= r - Lstar[8];
```

$$x1 := .7303370787$$

$$x2 := .7415730337$$

```
> th1:=evalf(theta(x1)); K1:=K(th1); th2:=evalf(theta(x2));
> K2:=K(th2);
```

$$th1 := 2.335603527$$

$$K1 := \begin{bmatrix} 3 - 1.384795807\sqrt{2} & 1 - .6923979033\sqrt{2} \\ 1 - .6923979033\sqrt{2} & 1 \end{bmatrix}$$

$$th2 := 2.347369792$$

$$K2 := \begin{bmatrix} 3 - 1.401678651\sqrt{2} & 1 - .7008393253\sqrt{2} \\ 1 - .7008393253\sqrt{2} & 1 \end{bmatrix}$$

```
> lambda:=<eigenvals(K1)>; mu:=<eigenvals(K2)>;
```

$$\lambda := \mathrm{RTABLE}(7549132, \begin{bmatrix} .9913837386 \\ 1.050219250 \end{bmatrix})$$

$$\mu := \mathrm{RTABLE}(7809316, \begin{bmatrix} .9963286091 \\ 1.021398433 \end{bmatrix})$$

```
> f[1]:=lambda[2]/lambda[1]; f[2]:=mu[2]/mu[1];
```

$$f_1 := 1.059346859$$

$$f_2 := 1.025162204$$

In the interest of brevity, we skip the intermediate results, and display only the last two experiments:

```
> l:= x1; L[8]:= L[7] - Lstar[8]; Lstar[9]:=
> (fibonacci(2)/fibonacci(4))*L[8];
```

$$l := .7303370787$$

$$L_8 := .03370786516$$

$$Lstar_9 := .01123595505$$

Note that the length of the i.o.u. at the end of the 8th experiment is 3.4% of original length, i.e., smaller than at the end of 10 experiments with the dichotomous search!

```
> x1:= l + Lstar[9]; x2 := r - Lstar[9];
```

$$x1 := .7415730338$$

$$x2 := .7528089888$$

```
> th1:=evalf(theta(x1)); K1:=K(th1); th2:=evalf(theta(x2));
> K2:=K(th2);
```

$$th1 := 2.347369792$$

$$K1 := \begin{bmatrix} 3 - 1.401678651\sqrt{2} & 1 - .7008393253\sqrt{2} \\ 1 - .7008393253\sqrt{2} & 1 \end{bmatrix}$$

$$th2 := 2.359136057$$

$$K2 := \begin{bmatrix} 3 - 1.418367442\sqrt{2} & 1 - .7091837208\sqrt{2} \\ 1 - .7091837208\sqrt{2} & 1 \end{bmatrix}$$

```
> lambda:=<eigenvals(K1)>; mu:=<eigenvals(K2)>;
```

$$\lambda := \text{RTABLE}(24765056, \begin{bmatrix} .9963286091 \\ 1.021398433 \end{bmatrix})$$

$$\mu := \text{RTABLE}(7680224, \begin{bmatrix} .9929088850 \\ 1.001216643 \end{bmatrix})$$

```
> f[1]:=lambda[2]/lambda[1]; f[2]:=mu[2]/mu[1];
```

$$f_1 := 1.025162204$$

$$f_2 := 1.008367090$$

$f_1 > f_2 \Rightarrow$ drop left end:

```
> l:= x1; L[9]:= L[8] - Lstar[9]; Lstar[10]:=
> (fibonacci(1)/fibonacci(3))*L[9];
```

$$l := .7415730338$$

$$L_9 := .02247191011$$

$$Lstar_{10} := .01123595506$$

```
> x1:= l + Lstar[10]; x2 := r - Lstar[10];
```

$$x1 := .7528089889$$

$$x2 := .7528089887$$

As expected, $x_1 = x_2$. Let us estimate the optimum by dichotomous search over the last i.o.u.: Let $\delta = L^*_{10}/10$

```
> delta:= Lstar[10]/10;
```

$$\delta := .001123595506$$

```
> x1:= (l + r - delta)/2; x2:= (l + r + delta)/2;
```

$$x1 := .7522471910$$

$$x2 := .7533707866$$

```
> th1:=evalf(theta(x1)); K1:=K(th1); th2:=evalf(theta(x2)); K2:=K(th2);
```

$$th1 := 2.358547744$$

$$K1 := \begin{bmatrix} 3 - 1.417537647\sqrt{2} & 1 - .7087688235\sqrt{2} \\ 1 - .7087688235\sqrt{2} & 1 \end{bmatrix}$$

$$th2 := 2.359724370$$

$$K2 := \begin{bmatrix} 3 - 1.419196745\sqrt{2} & 1 - .7095983727\sqrt{2} \\ 1 - .7095983727\sqrt{2} & 1 \end{bmatrix}$$

```
> lambda:=<eigenvals(K1)>; mu:=<eigenvals(K2)>;
```

$$\lambda := \text{RTABLE}(7680584, \begin{bmatrix} .9943254329 \\ 1.000973602 \end{bmatrix})$$

$$\mu := \text{RTABLE}(7816444, \begin{bmatrix} .9914931757 \\ 1.001459540 \end{bmatrix})$$

```
> f[1]:=lambda[2]/lambda[1]; f[2]:=mu[2]/mu[1];
```

$$f_1 := 1.006686110$$

$$f_2 := 1.010051874$$

$f_2 > f_1 \Rightarrow$ delete $[\,x_2,\, r\,]$ and take as most likely estimate of the optimum the midpoint of remaining interval $[\,l,\, x_1\,]$:

```
> x[o]:=(l+x1)/2; thopt:=evalf(theta(x[o])); Kopt:= K(thopt);
```

$$x_o := .7469101124$$

$$thopt := 2.352958768$$

$$Kopt := \begin{bmatrix} 3 - 1.409630165\sqrt{2} & 1 - .7048150824\sqrt{2} \\ 1 - .7048150824\sqrt{2} & 1 \end{bmatrix}$$

```
> lambda:=<eigenvals(Kopt)>;
```

$$\lambda := \text{RTABLE}(7262512, \begin{bmatrix} .9986575537 \\ 1.007824349 \end{bmatrix})$$

```
> f[o]:=lambda[2]/lambda[1];
```

$$f_o := 1.009179118$$

1.6 Golden-Section Search

This method is similar to the method based on Fibonacci numbers, but its implementation is much simpler. The outcome is that its convergence is a bit slower than that of the former. A major difference with the Fibonacci search is that the number of elimination stages is not prescribed.

The basis of the golden-search method is the Fibonacci sequence. Indeed, the golden-search strategy is derived from the Fibonacci search under the assumption that n in the Fibonacci search is "large," and hence, we denote it by N. The length of the interval of uncertainty is shrunken at every iteration by the same proportion, as opposed to the Fibonacci search.

In order to find the length L_k of the interval $\mathcal{I}_k$ at the kth iteration of the golden search, we compute the corresponding lengths of the Fibonacci search for $n = N \to \infty$, namely,

$$L_2 = \lim_{N\to\infty} \frac{F_{N-1}}{F_N} \tag{1.36a}$$

$$\begin{aligned} L_3 &= \lim_{N\to\infty} \frac{F_{N-2}}{F_N} = \lim_{N\to\infty} \frac{F_{N-2}}{F_{N-1}} \frac{F_{N-1}}{F_N} \\ &= \lim_{N\to\infty} \left(\frac{F_{N-1}}{F_N}\right)^2 \end{aligned} \tag{1.36b}$$

In general,

$$L_k = \lim_{N\to\infty} \left(\frac{F_{N-1}}{F_N}\right)^{k-1} \tag{1.36c}$$

Hence, all we need to implement this method is the above limit, which is computed below: Recall eq.(1.22b), for $k = N$:

$$F_N = F_{N-1} + F_{N-2} \tag{1.37a}$$

Therefore,

$$\frac{F_N}{F_{N-1}} = 1 + \frac{F_{N-2}}{F_{N-1}} \tag{1.37b}$$

Now let

$$\gamma \equiv \lim_{N\to\infty} \frac{F_N}{F_{N-1}} \tag{1.37c}$$

Upon taking limits, eq.(1.37b) can be rewritten as

$$\gamma = 1 + \frac{1}{\gamma}$$

or

$$\gamma^2 - \gamma - 1 = 0, \quad \gamma > 0 \tag{1.37d}$$

whence,

$$\gamma = 1.618 \quad \text{or} \quad \gamma = -0.618 \tag{1.37e}$$

Obviously, we need only the positive root, and hence, L_k becomes

$$L_k = \left(\frac{1}{\gamma}\right)^{k-1} = (0.618)^{k-1} \tag{1.38}$$

Greeks in the classical period, around the fifth century B.C.E., coined the expression *golden section* to refer to a rectangle of *divine proportions*, whose base b and height h observe the relation

$$\frac{b+h}{b} = \frac{b}{h} \tag{1.39}$$

The foregoing equation readily leads to one on the ratio b/h identical to eq.(1.37d), namely,

$$\left(\frac{b}{h}\right)^2 - \frac{b}{h} - 1 = 0$$

thereby showing that the real solution to eq.(1.39) is, indeed, γ, the golden section.

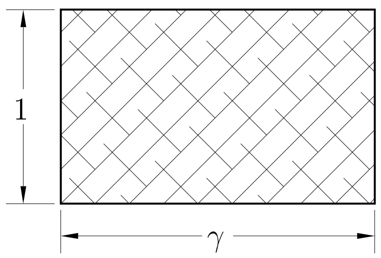

Figure 1.5: A rectangle with sides obeying the divine proportion

This relation is seen in the facade of the Parthenon, besides being present in nature and in many artifacts[4]. Shown in Fig. 1.5 is a rectangle with sides obeying the divine proportion.

To implement the search, we need the quantity L_2^*, which is defined below:

$$L_2^* = \frac{F_{N-2}}{F_N} = \frac{F_{N-2}}{F_{N-1}}\frac{F_{N-1}}{F_N} = \frac{1}{\gamma^2} = 0.382 \tag{1.40}$$

1.6.1 Example: Finding the Maximum Dexterity Posture of a Two-Joint Robotic Manipulator

We implement below the golden-section search strategy by means of a Maple worksheet. We shall resort the K, θ and x procedures introduced earlier.

Let us calculate γ:

```
> eq:=(x^2 - x - 1);
```

$$eq := x^2 - x - 1$$

[4]For example, Microsoft Word uses the golden-section ratio to proportion its margins.

```
> g:=<solve(eq, x)>;
```

$$g := \text{RTABLE}\left(25969576, \left[\begin{array}{c} \frac{1}{2}+\frac{1}{2}\sqrt{5} \\ \frac{1}{2}-\frac{1}{2}\sqrt{5} \end{array}\right]\right)$$

```
> gama:=evalf(g[1]);#Maple reserves the use of "gamma" for the "gamma
> function"!
```

$$gama := 1.618033989$$

```
> amag:=1.0/gama; #we'll also need the
> reciprocal of gamma
```

$$amag := .6180339887$$

```
> Lstar[2]:=amag^2;
```

$$Lstar_2 := .3819660112$$

Now let us find n from the problem specification: $L_n = 0.05$, which leads to

$$\frac{1}{\gamma^{n-1}} = 0.05 \tag{1.41}$$

```
> eq:= (n-1)*ln(amag) - ln(0.05)=0;
```

$$eq := -.4812118251\, n + 3.476944099 = 0$$

```
> n:=ceil(solve(eq, n));
```

$$n := 8$$

```
> l:=0; r:=1;#extremes of initial normal interval
```

$$l := 0$$

$$r := 1$$

```
> L[0]:= r - l;#length of initial interval
```

$$L_0 := 1$$

```
> x1:=l+Lstar[2]; x2:=r-Lstar[2];
```

$$x1 := .3819660112$$

$$x2 := .6180339888$$

```
> th1:=evalf(theta(x1)); K1:=K(th1); th2:=evalf(theta(x2));
> K2:=K(th2);lambda:=<eigenvals(K1)>; mu:=<eigenvals(K2)>;
```

$$th1 := 1.970790199$$

$$K1 := \left[\begin{array}{cc} 3-.7788253964\sqrt{2} & 1-.3894126982\sqrt{2} \\ 1-.3894126982\sqrt{2} & 1 \end{array}\right]$$

$$th2 := 2.218000007$$

$$K2 := \begin{bmatrix} 3 - 1.205915875\sqrt{2} & 1 - .6029579377\sqrt{2} \\ 1 - .6029579377\sqrt{2} & 1 \end{bmatrix}$$

$$\lambda := \mathrm{RTABLE}(25969616, \begin{bmatrix} .8138991148 \\ 2.084675447 \end{bmatrix})$$

$$\mu := \mathrm{RTABLE}(25969656, \begin{bmatrix} .9389910200 \\ 1.355586395 \end{bmatrix})$$

```
> f[1]:=lambda[2]/lambda[1]; f[2]:=mu[2]/mu[1];
```

$$f_1 := 2.561343794$$

$$f_2 := 1.443662789$$

$f_1 > f_2 \Rightarrow$ drop the left end:

```
> L[2]:=L[0] - Lstar[2]; Lstar[3]:= x2 - x1;#You should be able to
> prove that amag^3 = x2 - x1
```

$$L_2 := .6180339888$$

$$Lstar_3 := .2360679776$$

```
> x3:= r - Lstar[3];#x2 is now a distance Lstar_3 from new left end,
> x1, to the left of x3
```

$$x3 := .7639320224$$

```
> th3:=evalf(theta(x3)); K3:=K(th3); lambda:=<eigenvals(K3)>;
```

$$th3 := 2.370784070$$

$$K3 := \begin{bmatrix} 3 - 1.434695103\sqrt{2} & 1 - .7173475515\sqrt{2} \\ 1 - .7173475515\sqrt{2} & 1 \end{bmatrix}$$

$$\lambda := \mathrm{RTABLE}(25969696, \begin{bmatrix} .9650358237 \\ 1.005998904 \end{bmatrix})$$

```
> f[3]:=lambda[2]/lambda[1];f[2];#evaluate f[3] & recall f[2]
```

$$f_3 := 1.042447212$$

$$1.443662789$$

$f_2 > f_3 \Rightarrow$ drop left end:

```
> L[3]:=L[2] - Lstar[3]; Lstar[4]:= x3 - x2;
```

$$L_3 := .3819660112$$

$$Lstar_4 := .1458980336$$

```
> x4:= r - Lstar[4];
```

$$x4 := .8541019664$$

```
> th4:=evalf(theta(x4)); K4:=K(th4); lambda:=<eigenvals(K4)>;
```

$$th4 := 2.465209815$$

$$K4 := \begin{bmatrix} 3 - 1.559684146\sqrt{2} & 1 - .7798420728\sqrt{2} \\ 1 - .7798420728\sqrt{2} & 1 \end{bmatrix}$$

$$\lambda := \mathrm{RTABLE}(25969736, \begin{bmatrix} .7516661806 \\ 1.042607347 \end{bmatrix})$$

```
> f[4]:=lambda[2]/lambda[1];f[3];#evaluate f[4] & recall f[3]
```

$$f_4 := 1.387061669$$

$$1.042447212$$

Again, for brevity we introduce only the last two experiments. We have the interval $[x_5, x_6]$ and hence,

```
> L[6]:=L[5] - Lstar[6]; Lstar[7]:= x6 - x3;
```

$$L_6 := .0901699440$$

$$Lstar_7 := .0344418544$$

```
> x7:= x5 + Lstar[7];
```

$$x7 := .7426457872$$

```
> th7:=evalf(theta(x7)); K7:=K(th7);lambda:=<eigenvals(K7)>;
```

$$th7 := 2.348493177$$

$$K7 := \begin{bmatrix} 3 - 1.403280430\sqrt{2} & 1 - .7016402150\sqrt{2} \\ 1 - .7016402150\sqrt{2} & 1 \end{bmatrix}$$

$$\lambda := \mathrm{RTABLE}(25969856, \begin{bmatrix} .9967977596 \\ 1.018664025 \end{bmatrix})$$

```
> f[7]:=lambda[2]/lambda[1];f[3];#evaluate f[7] & recall f[3]:
```

$$f_7 := 1.021936511$$

$$1.042447212$$

$f_7 < f_3 \Rightarrow$ drop right end and perform last experiment:

```
> L[7]:=L[6] - Lstar[7]; Lstar[8]:= x3 - x7;
```

$$L_7 := .0557280896$$

$$Lstar_8 := .0212862352$$

```
> x8:= x5 + Lstar[8];
```

$$x8 := .7294901680$$

```
> th8:=evalf(theta(x8)); K8:=K(th8);lambda:=<eigenvals(K8)>;
```

$$th8 := 2.334716645$$

$$K8 := \begin{bmatrix} 3 - 1.383515463\sqrt{2} & 1 - .6917577317\sqrt{2} \\ 1 - .6917577317\sqrt{2} & 1 \end{bmatrix}$$

$$\lambda := \mathrm{RTABLE}(25969896, \begin{bmatrix} .9910087350 \\ 1.052404934 \end{bmatrix})$$

```
> f[8]:=lambda[2]/lambda[1];f[7];#evaluate f[8] & recall f[7]:
```

$$f_8 := 1.061953237$$

$$1.021936511$$

$f_8 < f_7 \Rightarrow$ delete left end and accept midpoint, of abscissax_o, as best estimate of optimum:

```
> x[o]:= (x8 + x3)/2;
```

$$x_o := .7467110952$$

Notice that length of final i.o.u. is 3.4% the length of original i.o.u. Evaluate $f_o = f(x_o)$:

```
> th[o]:=evalf(theta(x[o])); Ko:=K(th[o]);lambda:=<eigenvals(Ko)>;
```

$$th_o := 2.352750357$$

$$Ko := \begin{bmatrix} 3 - 1.409334444\sqrt{2} & 1 - .7046672222\sqrt{2} \\ 1 - .7046672222\sqrt{2} & 1 \end{bmatrix}$$

```
> L[8]:= x3 - x8;#Length of final i.o.u.
```

$$L_8 := .0344418544$$

$$\lambda := \mathrm{RTABLE}(25969936, \begin{bmatrix} .9985709394 \\ 1.008329177 \end{bmatrix})$$

```
> f[o]:=lambda[2]/lambda[1];
```

$$f_o := 1.009772203$$

```
>  tho:= evalf(th[o]*180/Pi);# theta_optimum in degrees
```

$$tho := 134.8026657$$

1.7 Bibliography

Becker, B., 1996, *Automated Screw Driving Device*, U.S. Patent Application, September, Swabey, Ogilvy, Renault, Patent Attorneys, Montreal.

Chou, P. C. and Pagano, N. J., 1992, *Elasticity: Tensor, Dyadic and Engineering Approaches*, Dover Publications, Inc., New York.

Gere, J. M., and Timoshenko, S. P., 1991, *Mechanics of Materials*, Chapman and Hall, Boston, Massachusetts.

Rao, S. S., 1996, *Engineering Optimization*, 3rd. Edition, John Wiley and Sons, Inc., New York.

Norton, R. L., 2000, *Machine Design: An Integrated Approach*, Prentice-Hall, Englewood Cliffs, N.J..

Taguchi, G., 1993, *Taguchi on Robust Technology Development: Bringing Quality Engineering Upstream*, ASME Press, New York.

Wilde, D., 1992, "Monotonicity analysis of Taguchi's robust circuit design problem", *ASME Journal of Mechanical Design*, Vol. 114, pp. 616–619.