DEVELOPMENT AND APPLICATION OF NEW EVOLUTIONARY ALGORITHMS AND COMPUTER GRAPHICS TOOLS FOR THE DESIGN OF MULTIBODY SYSTEMS

Except where reference is made to the work of others, the work described in this thesis is my own or was done in collaboration with my advisory committee. This dissertation does not include proprietary or classified information.

P. A. Simionescu

Petru-Aurelian Simionescu

Certificate of Approval:

Amnon J. Meir
Professor
Mathematics

David G. Beale, Chair
Professor
Mechanical Engineering

Dan B. Marghitu
Associate Professor
Mechanical Engineering

Gerry V. Dozier
Associate Professor
Computer Science and Software Engineering

Stephen L. McFarland
Acting Dean
Graduate School

DEVELOPMENT AND APPLICATION OF NEW EVOLUTIONARY ALGORITHMS AND COMPUTER GRAPHICS TOOLS FOR THE DESIGN OF MULTIBODY SYSTEMS

Petru-Aurelian Simionescu

A Dissertation

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements of the

Degree of

Doctor of Philosophy

Auburn, Alabama

December 17, 2004

DISSERTATION ABSTRACT

DEVELOPMENT AND APPLICATION OF NEW EVOLUTIONARY ALGORITHMS AND COMPUTER GRAPHICS TOOLS FOR THE DESIGN OF MULTIBODY SYSTEMS

Petru-Aurelian Simionescu

Doctor of Philosophy, December 17, 2004
(Doctor of Philosophy, Politehnica University, Bucharest 1999)
(Bachelor of Science, Politehnica University, Bucharest 1992)

151 Typed pages

Directed by David G. Beale

In this dissertation new optimum design tools are proposed and tested for solving several difficult multibody design problems drawn from engineering practice.

Estimation of Distribution Algorithms (EDA) have been considered as optimization tools for their robustness and global optimum finding capabilities. EDA's are the latest newcomers to the growing family of Evolutionary Algorithms, and their potential for solving constrained optimization problems has been investigated in this dissertation for the first time.

Secondly a new method of visualizing multivariate functions is proposed where the dimension reduction is done by repeated partial global minimizations and maximizations. For objective functions in particular it allows studying constraint activity, performing parametric design studies, *what-if* scenario analysis, etc.

These tools were applied successfully to the solution of several multibody design problems:

▪ Synthesis of the slider-crank and oscillating slide actuators for imposed output member displacement and maximum motion transmission efficiency. By systematically investigating their

properties, design and performance charts were generated and new properties and possible applications identified (patent pending).

- Determination of the optimum number of gear teeth of a multispeed automatic transmission that best fulfill imposed gear ratios, under the conditions of avoiding gear-teeth undercut, noninterference of neighboring gears and assembling equally spaced planets.

- Synthesis of a five-link rear wheel suspension used in automobiles from the condition of simultaneously ensuring that the wheel base, wheel track, toe and camber angles have minimum variations while the wheel oscillates. Displacement, velocity, acceleration and roll-center height analyses were also performed in a new, simplified manner and improvements of the solutions obtained through synthesis over an existing design revealed. The analysis results were found to agree with those obtained using a commercially available multibody simulation software.

The above problems were solved with the goal of revealing general properties of the respective multibody systems and whenever possible, parameter charts and design recommendations have been advanced to aid the practicing engineer.

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. David Beale for his guidance and support during my studies and research at Auburn University. I am also grateful to the members of my advisory committee: Dr. Gerry Dozier, Dr. A. J. Meir and Dr. Dan Marghitu for their willingness to serve on my committee, involvement with oral examination and for reviewing this dissertation. Thanks are also extended to the outside reader, Dr. Ted S. Kornecki for his thorough review of this dissertation.

My deepest appreciation goes to my wife Florentina, my parents and my sister for their constant love, encouragement and moral support that made this work possible.

Style manual or journal used Journal of Mechanical Design

Computer software used Microsoft Word 2002

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

1.1. Optimum Seeking, Visualization and the Design Process

Any engineering design problem is an optimization problem, where certain functional performances are sought to be maximized while the manufacturing and maintenance expenses of the product are sought to be minimized [2] [10]. In the design of multibody systems there are well defined measures that are desired to be maximums or minimums. For example the efficiency or mechanical advantage should be maximum while the departure between the imposed and the actual motion of the output member of a linkage mechanism should be minumum. Likewise the error between the imposed and actual gear ratios of a multispeed gear transmission should also be minimum [5] [7]. In many cases additional conflicting requirements can be prescribed and the designer must seek a superior tradeoff among these [4]. These translate into multiobjective optimization problems, and although not the main topic of this dissertation, Chapter 6 presents the example of synthesizing a rear wheel suspension system used in automobiles where the problem of satisfying several functional requirements was formulated in a simplified manner as a single objective optimization problem.

Optimal design is the process of finding the values of certain modifiable parameters called *design variables* that minimize or maximize a function called *cost* or *objective function*. In order for the solution to the optimization problem to be acceptable, additional requirements must be satisfied, known as *constraints*. Formally, a general nonlinear programming problem requires finding the minimum or maximum of a function of *n* real variables [10]:

$$f(\mathrm{x}_1 ... \mathrm{x}_n) \qquad (1.1)$$

subject to side constraints:

$$\mathrm{x}_{j\min} \le \mathrm{x}_j \le \mathrm{x}_{j\max} \quad (1 \le j \le n), \tag{1.2}$$

inequality constraints:

$$g_j(\mathrm{x}_1 ... \mathrm{x}_n) \le 0 \quad (1 \le j \le n_1) \tag{1.3}$$

and equality constraints:

$$h_j(\mathrm{x}_1 ... \mathrm{x}_n) = 0 \quad (1 \le j \le n_2). \tag{1.4}$$

Often case in engineering problems the design variables are imposed integer or discrete values. Imposing some or all of the design variables to be integer can be accounted for by further formulating equality constraints of the form:

$$round(\mathrm{x}_j) - \mathrm{x}_j = 0 \quad (1 \le j \le n_3). \tag{1.5}$$

where *round* is the round-off function and $n_3 \le n$. Alternatively a continuous optimization problem can be solved first and then a search performed in the neighborhood of the optima thus found, until a set of integers that satisfy all constraints is identified (the cases of the design variables being imposed discrete values can be handled in a similar manner) [2] [10].

When investigating optimization problems the understanding of the relations between design variables is greatly enhanced by visualizing the *n*-dimensional objective function surface [6] [16]. This would allow establishing dependences between design variables as they aggregate, together with the constraints, within the objective function. Moreover, according to [1], “the knowledge of a family of good solutions is far more important than obtaining an isolated optimum.” This emphasizes the need of identifying not only an improvement to the existing solution - as some advanced engineering analysis software are capable of [16] - or of the global optimum point, but also of the most promising local optimum points which may have additional good properties the designer can exploit (like better stability of the optimum or less sensitivity to constraint violation) [10]. Unfortunately, visualization of complex design spaces yields difficulty due to our limitations of perceiving dimensions higher than three, although there are a number of methods available of visualizing hypersurfaces of functions of more than two variables, as discussed in Chapter 3.

Because of their robustness, ease of implementation and global optimum finding capabilities, Evolutionary Algorithms (EA) have drawn considerable attention as optimization methods for engineering [9]. EAs are stochastic optimization methods that use populations of individuals (design solutions) rather than a single solution at a time. They employ the simulated genetic operators of *mutation* and *crossover* and the principle of *survival of the fittest* in natural evolution to evolve improved populations of individuals [3]. Starting with a randomly initialized population, new individuals are generated, their probability of survival depending on their fitness (the best are kept with higher probability, the worst are rapidly discarded).

Three main algorithmic trends have been developed over the years that employ the aforementioned evolutionary schemes: Genetic Algorithms (GA), Evolutionary Strategies (ES) and Evolutionary Programming (EP). In addition, there is a new class of algorithms that emerged relatively recently into the field of Evolutionary Computation, called Estimation of Distribution Algorithms (EDA) [8]. As compared to the more traditional Evolutionary Algorithms, EDAs do not employ mutation and crossover operations. Instead, the new population of individuals is sampled from a probability distribution, which is estimated from a database of selected individuals from the previous generation. This reduces the number of parameters the user must set at the beginning of the run (i.e. population size and survival rate), making EDAs easier to implement; moreover, their population dynamics is easier to comprehend, which facilitates tuning these parameters in order to achieve improved searching performance.

1.2. Objectives of this Research

The first research objective is to investigate the use of Estimation of Distribution Algorithms in solving constrained optimization problems. EDAs are fairly new optimization techniques and their potential of solving constrained optimization problems has not been yet investigated [12]. In this dissertation several constraint handing techniques of the penalty and repair type were experimented with in conjunction with EDAs, and some useful conclusions highlighted.

The second objective is the development of a new technique of visualizing the hypersurface of single-valued functions of more than two variables useful in inspecting the design space of optimization problems, "what-if" scenario analyses etc. This new technique is tightly linked to optimization theory since the dimension reduction is performed through repeated partial global minimization and maximization of the function to be visualized [13].

The third objective is to illustrate the advantage of using EDAs and of the proposed visualization technique in solving several complex engineering optimization problems:

▪ synthesis of the slider-crank and oscillating slide mechanisms for maximum motion transmission efficiency and imposed displacement of the output members [15];

▪ determination of the optimum number of gear teeth of an automatic transmission used in automobiles [14];

▪ synthesis of a five-link suspension system used in automobiles of the condition of minimum camber, wheel track and wheel base variations [11].

1.3. Organization of this Dissertation

The capability of Estimation of Distribution Algorithms (EDAs) of solving constrained optimization problems has been investigated in Chapter 2. Extensive numerical experiments have been performed to test their suitability of solving constrained optimization problems when coupled with several types of constraint handling techniques of the penalty and repair type.

Chapter 3 begins with a review of the known hypersurface and hyperobject visualization techniques. Then a description of a new method of visualizing multivariate functions, in particular objective functions, is given, supported by several numerical examples.

In Chapter 4 the usefulness of the author's visualization technique is illustrated in the motion capability investigation and optimum design of the slider-crank and oscillating slide actuators, from the condition of maximum motion transmission efficiency and imposed displacement of the output member.

In Chapter 5 the problem of optimum synthesis of the gear teeth of an automatic transmission used in automobiles is solved using an EDA of the Population Based Incremental Learning type (PBIL). The visualization method developed in Chapter 3 proved useful in investigating the design space of this optimization problem and selecting the final number of gear teeth based on additional constructive criteria.

A complex optimization problem is solved in Chapter 6, that of synthesizing of a five-link suspension system for automobile from the condition of the camber, wheel-track and wheel-base having minimum variation as the wheel oscillates. Displacement, velocity and acceleration analysis of this suspension mechanism are also performed in a new, simplified manner and improvements over an existing design revealed. The simulation results were found to agree with those obtained using MSC.visualNastran 4D multibody simulation software.

Concluding remarks and directions of further studies are outlined in Chapter 7.

1.4 References

[1] Arabeyre, J., Steiger, J. and Teather, W. (1969) “The Airline Crew Scheduling Problem: A Survey,” Transportation Science, Vol. 3, p. 140-163.

[2] Arora, J.S. (2004) *Introduction to Optimum Design*, Academic Press.

[3] Bäck, T., Fogel, D. and Michalewicz, Z. (2000) *Handbook of Evolutionary Computation*, The Institute of Physics Publishing.

[4] Collette, Y. and Siarry, P. (2003) *Multiobjective Optimization: Principles and Case Studies*, Springer-Verlag.

[5] Datoussaid S., Verlinden O. and Conti C. (2002) “Application of Evolutionary Strategies to Optimal Design of Multibody Systems,” Multibody System Dynamics, Vol. 8, p. 393–408.

[6] Jones C V (1996) *Visualization and Optimization*, Kluwer Academic Publishers.

[7] Kunjur, A. and Krishnamurty S. (1997) “Genetic Algorithms in Mechanism Synthesis,” Journal of Applied Mechanisms and Robotics, Vol. 4, No. 2, p. 18-24.

[8] Larrañaga, P. and Lozano, J. A., Eds., (2002) *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, Kluwer Academic Publishers.

[9] Michalewicz, Z., Dasgupta, D., Le Riche, R.G. and Schoenauer, M. (1996) “Evolutionary Algorithms for Constrained Engineering Problems,” Computers and Industrial Engineering, Vol. 30, No. 4, p. 851-870.

[10] Papalambros P. Y. and Wilde D. J. (2000) *Principles of Optimal Design: Modeling and Computation*, Cambridge University Press.

[11] Simionescu, P. A. and Beale D. G. (2002) "Synthesis and Analysis of the Five-Link Rear Suspension System Used in Automobiles," Mechanism and Machine Theory, Vol. 37, p. 815-832.

[12] Simionescu, P. A., Beale D. G. and Dozier G. V. (2004) "Constrained Optimization Problem Solving Using Estimation of Distribution Algorithms," IEEE Congress on Evolutionary Computation, Portland - OR, June 20-23, 2004, Vol. 1, p. 296-302.

[13] Simionescu P. A., Beale D. (2004) “Visualization of Multivariable (Objective) Functions by Partial Global Optimization,” The Visual Computer Journal, Vol. 20(10), p. 665-681.

[14] Simionescu, P. A., Beale D. G. and Dozier G.V. (2004) "Teeth-Number Synthesis of a Multispeed Planetary Transmission Using an Estimation of Distribution Algorithm," ASME Journal of Mechanical Design (in review).

[15] Simionescu, P. A. and Beale D. G. (2002) "General Solutions for the Optimum Design of Slider-Rocker and Oscillating-Slide Actuators,” Mechanism and Machine Theory (in review).

[16] Winer E. H. and Bloebaum C. L. (2001) “Visual Design Steering for Optimization Solution Improvement,” Structural and Multidisciplinary Optimization, Vol. 22, No 3, p. 219–229.

CHAPTER 2. INVESTIGATION OF THE CONSTRAINED OPTIMIZATION PROBLEM SOLVING CAPABILITIES OF ESTIMATION OF DISTRIBUTION ALGORITHMS

Two variants of Estimation of Distribution Algorithm (EDA) were tested against solving several continuous optimization problems with constraints. Numerical experiments were conducted and comparison was made between constraint handling using several types of penalty and repair operators in case of both elitist and non-elitist implementations of the EDAs. Graphical display and animations of representative runs of the best and worst performers proved useful in enhancing the understanding of how such algorithms work and deciding which type of constraint is more appropriate for certain type of practical problems.

2.1. Introduction

Estimation of Distribution Algorithms are relatively newcomers to the field of Evolutionary Computation [1] [2]. Their appealing features over other evolutionary algorithms are a simple structure and an intuitive dynamics of the population which facilitate choosing the values of the control parameters. In standard EDAs there are no crossover and mutation operations, the new population being generated by sampling the probability distribution of a number of superior individuals selected from the current population. As highlighted in [3], the known EDA implementations differ by the probability distributions and by the survival selection schemes employed.

Several authors have reported solving combinatorial, discrete and continuous optimization problems using EDAs [2] [4] [5] [6]. There is however no report available on the capabilities of EDAs of solving constrained optimization problems. In this chapter the Univariate Marginal Distribution Algorithm (UMDA) and a variant of the Population Based Incremental Learning Algorithm (PBIL) were tested on solving three continuous objective functions with constraints. Comparison was made between constraint handling using penalty and repair techniques through numerical experimentation and some useful conclusions highlighted.

2.2 Estimation of Distribution Algorithms Tested

Two Estimation of Distribution Algorithms have been implemented in both elitist and non-elitist variants as described below.

▪ The first algorithm considered, a Univariate Marginal Distribution Algorithm (UMDA) [3] [5] was coded in the following structure:

Step 1: Generate M uniform random points within the imposed boundaries of the design variables [$x_{i\,\min} \ldots x_{i\,\max}$] ($i$=1...$n$) or until at least one feasible individual has been generated. The population size, M, is a constant specified by the user.

repeat

Step 2: Select the best N individuals in the population and evaluate the average and standard deviation vectors:

$$\{\mu_i\} = \left\{ \frac{1}{\mathrm{N}} \sum_{\mathrm{k}=1}^{\mathrm{N}} (x_i)_{\mathrm{k}} \right\} \quad (i = 1 ... n) \tag{2.1}$$

$$\{\sigma_i\} = \left\{ \sqrt{\frac{1}{\mathrm{N}} \sum_{\mathrm{k}=1}^{\mathrm{N}} \left[(x_i)_{\mathrm{k}} - \mu_i \right]^2} \right\} \quad (i = 1 ... n) \tag{2.2}$$

In the above formulae N is a specified integer restricted to 1 < N < M.

Step 3: Replace the whole current population by generating M normally distributed random points $\{x_i\}$, (i=1...n) with the averages and standard deviations given by equations (2.1) and

(2.2) respectively. In order to ensure that the newly generated individuals satisfy the imposed side constraints, the following corrections were performed:

$$\begin{aligned}&\text{If } x_i < x_{i\min} \text{ then } x_i = x_{i\min}\\&\text{If } x_i > x_{i\max} \text{ then } x_i = x_{i\max}\end{aligned} \tag{2.3}$$

Additionally, a record of the best-fit individual generated so far is kept to be provided as solution of the search.

until a certain stopping criteria is met.

The stopping criteria can be either attaining an imposed maximum number of generations $G_{\max}$ or exceeding a prescribed maximum number of function evaluations NF.

▪ The second Estimation of Distribution Algorithm tested was a variant of the Population Based Incremental Learning Algorithm (PBIL) [6]. The algorithm employs the same steps 1 and 2 and stopping criteria as UMDA, but uses a different population-generation scheme i.e.:

Step 3: Generate M new points $\{x_i\}$, (i=1...n, r=1...M) to replace the current population, using the standard deviations (2.2) and the following vector of corrected average values:

$$\left\{\mu_i^*\right\} = \left\{(1-\alpha)\cdot\mu_i + \alpha\cdot(x_i)_{\text{best}}\right\} \tag{2.4}$$

where μ_i are given by the same formula (2.1) and α is a variable parameter:

$$\alpha = w\cdot\left(G_c / G_{\max}\right)^n \tag{2.5}$$

with G_c current generation number and w a chosen constant between 0 and 1. It is to be noticed that for w=0 the algorithm becomes a UMDA algorithm. In order to ensure that the imposed side constraints are satisfied, the same tests in equation (2.3) are applied to the newly generated points. Similarly to UMDA, the best fit individual encountered so far is recorded to be provided as solution of the search.

In case of elitist implementations of the above two algorithms, further referred to as E-UMDA and E-PBIL, *Step 3* must be modified so that only M-1 new individuals are generated and the best fit individual in the population is not destroyed - evidently, there will no longer be necessary to keep a record of the best fit individual generated so far.

2.3. Constraint Handling Techniques

There are numerous constraint-handling techniques used in evolutionary computation as follows [7] [8] [9]:

▪ various implementations of the penalty method,

▪ specialized representations and operators,

▪ repair algorithms,

▪ separation of objectives and constraints (behavioral memory, superiority of feasible points, multiobjective optimization techniques)

▪ hybrid algorithms etc.

Of the known constraints handling techniques, penalty and repair methods will be numerically tested in association with UMDA and PBIL algorithms described in the previous paragraph.

2.3.1. Penalty methods

Three penalty methods have been numerically experimented with; all of them operate by providing some fitness value to the infeasible individuals in the population that will further help with their ranking. Two of the considered methods are step-type penalties while a third method employs the Euclidean-distance from the considered infeasible point to the closest feasible point as a measure of its infeasibility.

1) The first penalty method tested, of the step type, will be further referred to as *1K*-Penalty and has the form:

$$\text{fitness}(x_1 ... x_n) = \begin{cases} \mathrm{F}(x_1 ... x_n) & \text{if feasible} \\ K & \text{if infeasible} \end{cases} \tag{2.6}$$

where *K* a constant about one order of magnitude greater than the expected global maxima of the constrained function. Such a penalty is very easy to implement but has the main drawback that the search is difficult to initiate in case of highly constrained problems with their landscape resembling flat plateaus with scattered crevasses (or only one such crevasse).

2) A slightly more elaborate penalty method tested resembling the *K*-method in [10], further called *νK*-Penalty was:

$$\text{fitness}(x_1...x_n) = \begin{cases} \text{F}(x_1...x_n) & \text{if feasible} \\ \nu \cdot K & \text{if infeasible} \end{cases} \qquad (2.7)$$

with ν is the number of constraints violated at point ($x_1 \ldots x_n$). In this form some rough information about the degree of constraint violation at a certain point can be acquired, which can help directing the search toward the feasible domain. However, as will be seen in case of the first test problem below, the method is less effective when the global optima is bounded by more than one active constraint.

3) A third penalty method tested named *DK*-Penalty:

$$\text{fitness}(x_1...x_n) = \begin{cases} \text{F}(x_1...x_n) & \text{if feasible} \\ D^2 \cdot K & \text{if infeasible} \end{cases} \qquad (2.8)$$

employs the distance *D* between the considered infeasible point and the closest to it feasible point in the population [11]. This will require evaluating the Euclidean distance (or of some other norm) between the current point and all feasible points in the population, slowing down the algorithm.

2.3.2. Infeasible-individual repair

These constraint-handling techniques require that at least one feasible individual exists in the current population. It involves a line searching (or some other crossover operation) between the current infeasible point and a selected feasible individual in the population. In the present study the following repair methods have been experimented with:

Repair 1 (repair by line search): Assign to the infeasible individual to be repaired the closest feasible individual in the population. If there are no feasible individuals in the current population, the repair operation must be suspended and the infeasible points treated in a simple *1K*-Penalty manner (this is the form in which the method was implemented in the numerical experiments performed). Alternatively, in case of non-elitist algorithms, the best point encountered so far can be used as a second point for the line search operation. After the infeasible-feasible pairs have been made, a random search is performed along the line connecting the two points until a second feasible point is generated to be introduced in the population in replace to the considered infeasible individual [12].

Repair 2 (repair by crossover): Instead of doing a line search, which requires a number of objective function evaluations, one single crossover operation can be performed (for example a midpoint crossover) between the current infeasible and its closest feasible individuals. Since the offspring that will replace the infeasible parent may in turn be infeasible, the method is more of an *incomplete repair*.

Repair 3 (repair by cloning): Replace the infeasible individual with an identical copy of the feasible individual that is closes to it. When only one or two feasible individuals are available in the population, in order to preserve diversity (particularly for elitist algorithms), it might become necessary to repair only part of the infeasible individuals (a partial repair) to avoid standard deviation becoming too small, or to impose a lower limit upon the components of the standard deviation vector.

Combined repairs: Combination of the above approaches can also be employed, like for example repairing half of the infeasible individuals using cloning and the other half using some crossover operation.

Even if they don't always eliminate the infeasible individuals, the above listed repair methods contribute to a favorable confinement of the population toward the feasible domain(s) of the search space. Repair methods 2 and 3 have the appealing feature that require less or no additional evaluations of the objective function. They are also suitable in case of discrete or integer optimization problems, when the feasible space is very fragmented or is reduced to only scattered points, a feature that will be used in Chapter 5 in finding the optimum teeth number of an multispeed automatic transmission.

2.4. Test Problems

Several numerical experiments were performed on solving three constrained objective functions. Since graphical representation and animation of the successive populations can provide a valuable insight into how algorithms work, preference was given to the following test functions of two variables:

2.4.1 Test Problem 1- the Sickle function

This is a slightly modified version of problem G6 in reference [13] which requires minimizing the function:

$$\mathrm{F}(x_1, x_2) = (x_1 - 20)^3 + (x_2 - 10)^3 \tag{2.9}$$

subject to:

$$\begin{aligned} \mathrm{g}_1 &= (x_1 - 5)^2 + (x_2 - 5)^2 - 100 \geq 0 \\ \mathrm{g}_2 &= -(x_1 - 5)^2 - (x_1 - 6)^2 + 82.81 \geq 0 \end{aligned} \tag{2.10}$$

and the side constraints:

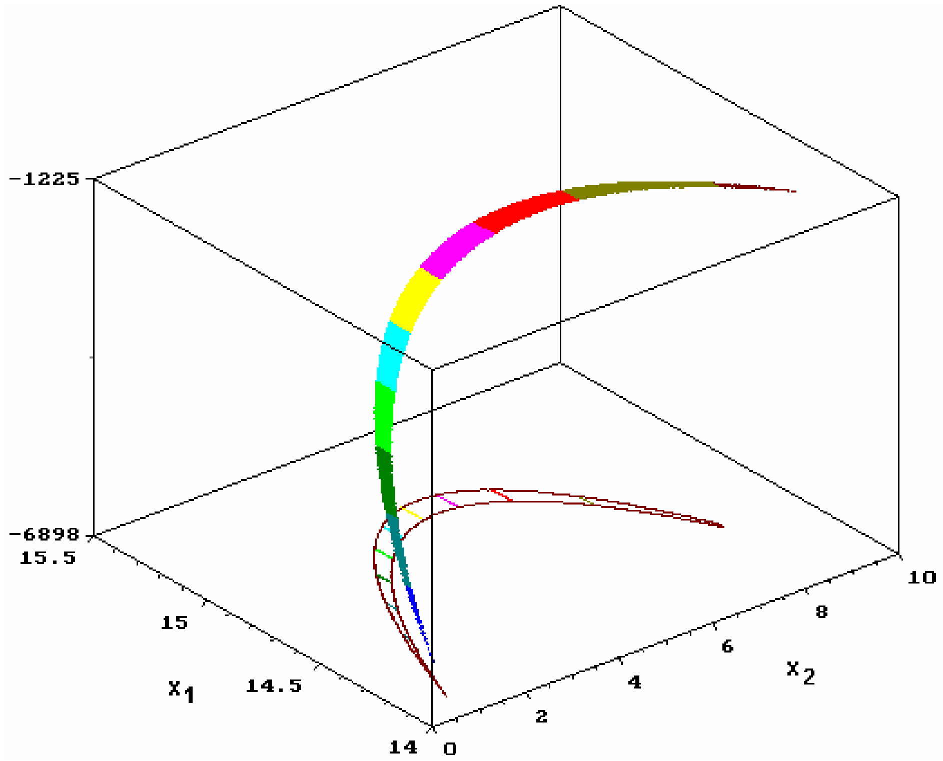


Fig. 2.1 Plot of test objective function 1 – the Sickle function.

$$0 \le x_1 \le 10 \quad \text{and} \quad 14 \le x_2 \le 15.5 \tag{2.11}$$

In its original form [13], the side constrains were over 10 times wider, making the ratio between the feasible and the infeasible spaces very small and therefore a starting feasible point hard to find. The global minimum point is located at x_1=14.095 and x_2=0.84296 for which the function value is 6961.8139 and both constraints are active. The maximum point, also double bounded, is located at x_1=14.095 and x_2=9.15704 and equals -1206.13556. As visible from the plot in Fig. 2.1, the feasible domain of this function is not convex.

2.4.2 Test Problem 2 - Koziel and Michalewicz G6 function

This second problem [13] requires finding the maximum point of:

$$\mathrm{F}(x_1, x_2) = \frac{\sin(2\pi x_1)\cdot \sin^3(2\pi x_2)}{(x_1 + x_2)\cdot x_2^3} \tag{2.12}$$

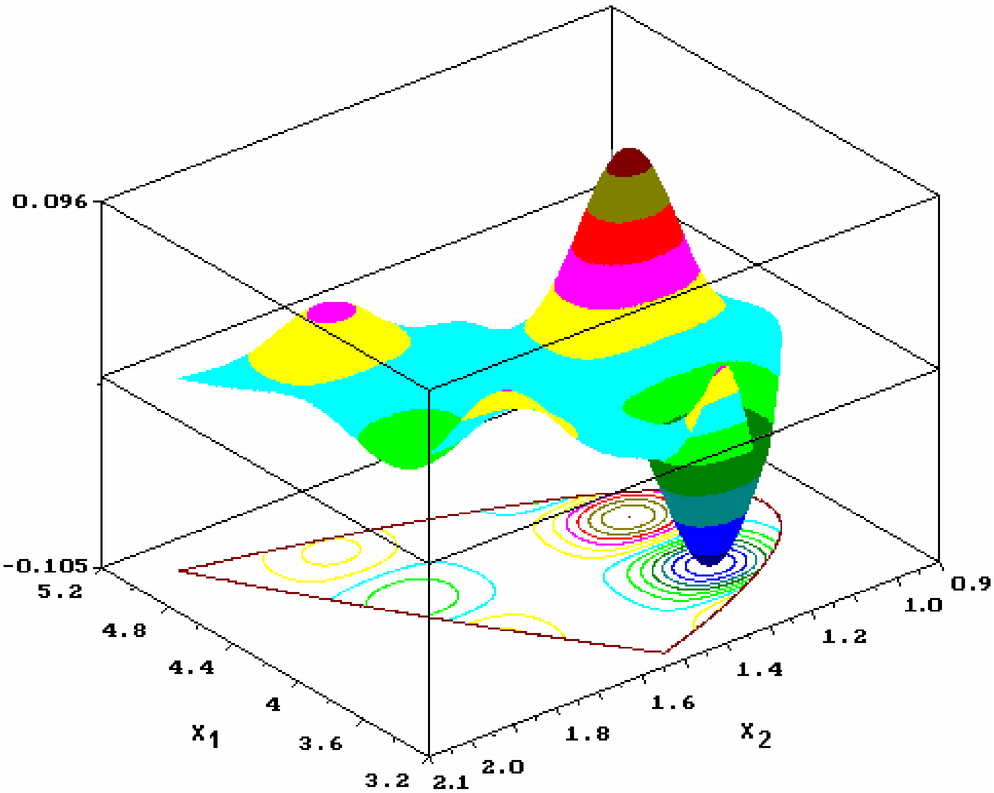


Fig. 2.2 Plot of test objective function 2 - Koziel and Michalewicz's G6 function.

subject to:

$$\begin{aligned} g_1 &= x_1 - x_2^2 - 1 \geq 0 \\ g_2 &= -(x_1 - 4)^2 + x_2 - 1 \geq 0 \end{aligned} \tag{2.13}$$

and the side constraints (modified as compared to the original form in [13] for the same reason as before):

$$3.2 \leq x_1 \leq 5.2 \quad \text{and} \quad 0.9 \leq x_2 \leq 2.1 \tag{2.14}$$

This multimodal function has its global maximum at x_1=1.24539 and x_2=4.2425 and equals 0.09582504. The global minimum is located at x_1=1.24492 and x_2=3.74154 where the function value is 0.10363448. Both the global minimum and the global maximum points are unbounded i.e. they are located inside the feasible domain (Fig. 2.2).

2.4.3 Test Problem 3 - Keane's function

The third test problem, due to Keane, also listed as problem G2 in [6], requires minimizing the function:

$$\mathrm{F}(x_1 \ldots x_n) = \frac{\left| \sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i) \right|}{\sqrt{\sum_{i=1}^{n} i \cdot x_i^2}} \tag{2.15}$$

subject to:

$$\begin{aligned} g_1 &= \sum_{i=1}^{n} x_i - 7.5n \le 0 \\ g_2 &= 0.75 - \prod_{i=1}^{n} x_i \le 0 \end{aligned} \tag{2.16}$$

and to the side constrains:

$$0 \le x_i \le 10 \quad \text{for} \quad 1 \le i \le n \tag{2.17}$$

This is a highly multimodal function that has its global minimum constrained by g_2. For n=2 its optimum equals 0.36497974 and occurs for x_1=1.60086 and x_2=0.468498. According to [6],

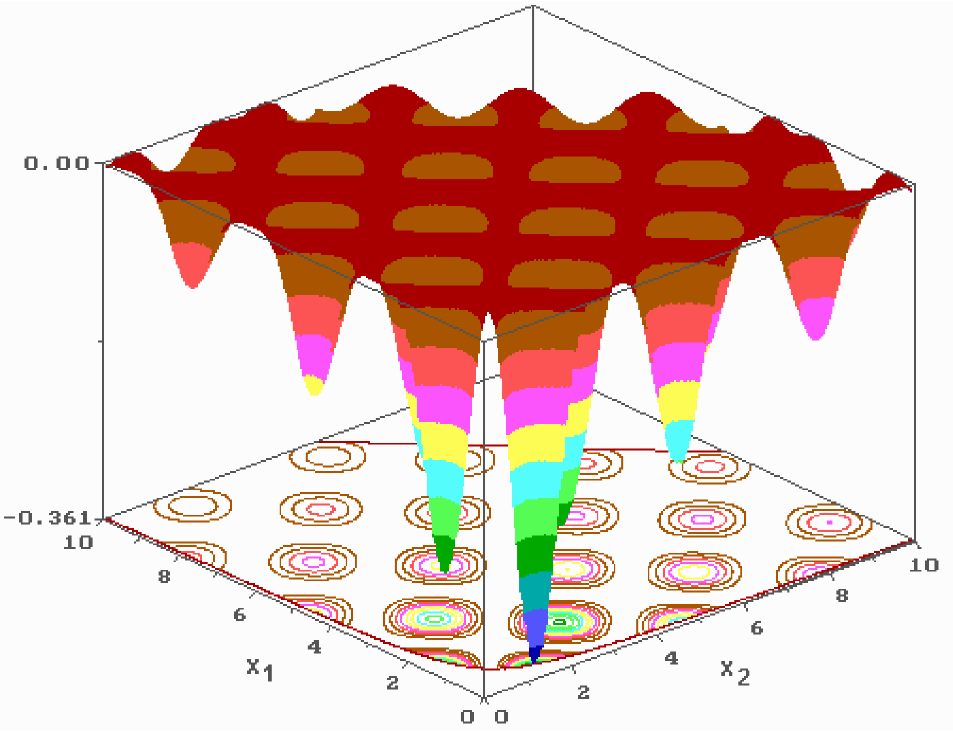


Fig. 2.3 Plot of test objective function 3 - Keane's function with n=2.

for n=20 the minimum value found so far equals 0.8036.

2.5. Numerical Results

A set of numerical experiments were conducted to test the capabilities of the UMDA and PBIL algorithms in association with the constraint handling techniques described above and the results are summarized in Tables 2.1-3.

No attempt was made during these experiments to fine tune the N, M or *w* parameters so that performances are maximized (in all cases N=50, M=25 and *w*=1 while the stopping criteria was limiting the maximum number of function evaluations to NF=5000). The main purpose of these numerical experiments was to identify promising combinations of Estimation of Distribution Algorithms and constraint handling techniques, their potential for improvement

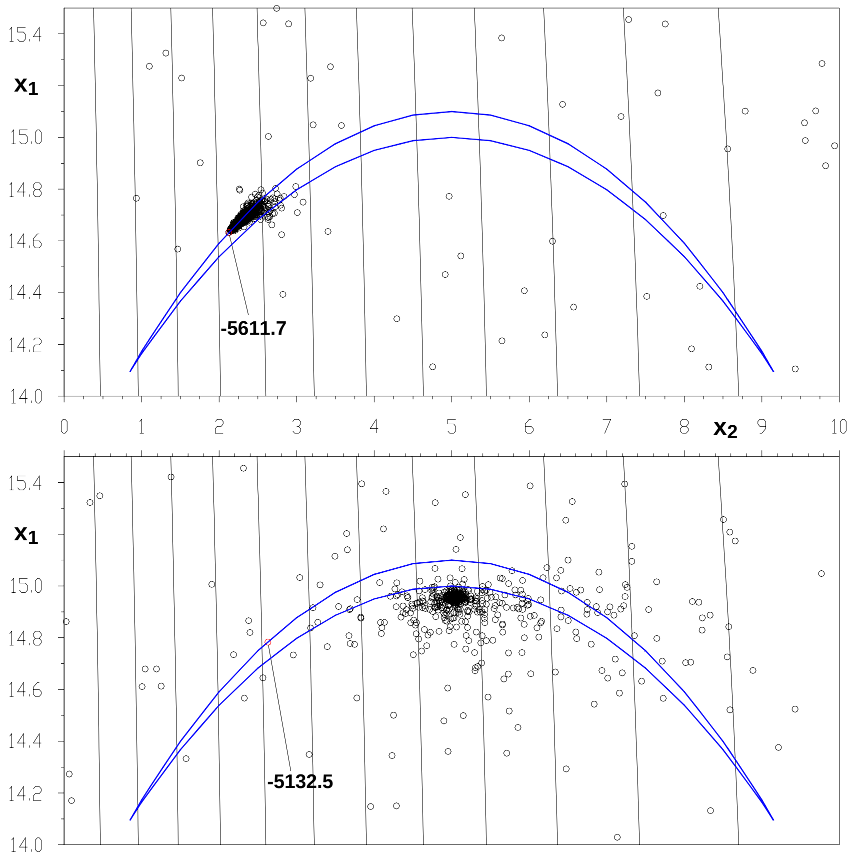


Fig. 2.4 Superimposed plots of the points generated during one run of E-UMDA + Repair 1 (top) and UMDA + *1K*-Penalty (bottom) algorithms on Test Problem 1.

Table 2.1 Results obtained for 500 runs of Test Problem 1 for M=50 and N=25 (known global optimum: -6961.81)

Algorithm	**Constraint handling technique**	**Best result**	**Average result**	**Worst result**
E-UMDA	Repair 1	-6945.91	-5607.47	-2997.05
E-PBIL	Repair 1	-6943.65	-5763.21	-3509.77
UMDA	Repair 1	-6939.16	-5553.78	-3372.03
E-UMDA	*vK*-Pen.	-6930.03	-5104.47	-2480.90
E-PBIL	*vK*-Pen.	-6912.24	-5159.63	-2249.93
PBIL	*1K*-Pen .	-6911.18	-5076.39	-2585.59
E-PBIL	*1K*-Pen .	-6903.78	-5244.17	-3117.43
E-PBIL	*DK*-Pen.	-6895.40	-5184.25	-2617.83
PBIL	Repair 1	-6892.11	-5661.05	-3373.31
E-UMDA	Repair 2	-6881.54	-4978.97	-1454.95
UMDA	Repair 3	-6874.920	-4135.440	-1321.91
E-PBIL	Repair 2	-6871.597	-5018.764	-1973.91
PBIL	Repair 2	-6870.774	-5050.324	-1636.22
UMDA	Repair 2	-6869.215	-5008.144	-1862.65
E-UMDA	*DK*-Pen.	-6860.034	-5068.846	-1542.06
UMDA	*DK*-Pen.	-6857.924	-5092.668	-2886.64
PBIL	*vK*-Pena.	-6847.324	-5120.544	-2624.04
E-UMDA	*1K*-Pena .	-6836.38	-5048.02	-2218.117
E-PBIL	Repair 3	-6821.19	-4113.49	-1286.33
E-UMDA	Repair 3	-6813.79	-4148.39	-1279.73
PBIL	Repair 3	-6763.21	-4187.59	-1396.38
UMDA	*vK*-Pen.	-6755.85	-5095.72	-1776.14
PBIL	*DK*-Pen.	-6677.15	-5132.26	-2123.77
UMDA	*1K*-Pen.	-6674.55	-5125.19	-2680.86

and reasons why they performed or did not perform well.

Problem 1 has a non-convex feasible space with only one minimum and one maximum (both double constrained). It is therefore not surprising that the elitist E-UMDA and E-PBIL algorithms with line-search repair performed well. This is because only feasible individuals were sampled during the search and the monotonicity of the function favored a constant downhill migration of the population.

This is also illustrated by Fig. 2.4-top where all individuals generated during one run of the E-UMDA + *Repair 1* algorithm (less the intermediate points occurring during line searches) are plotted.

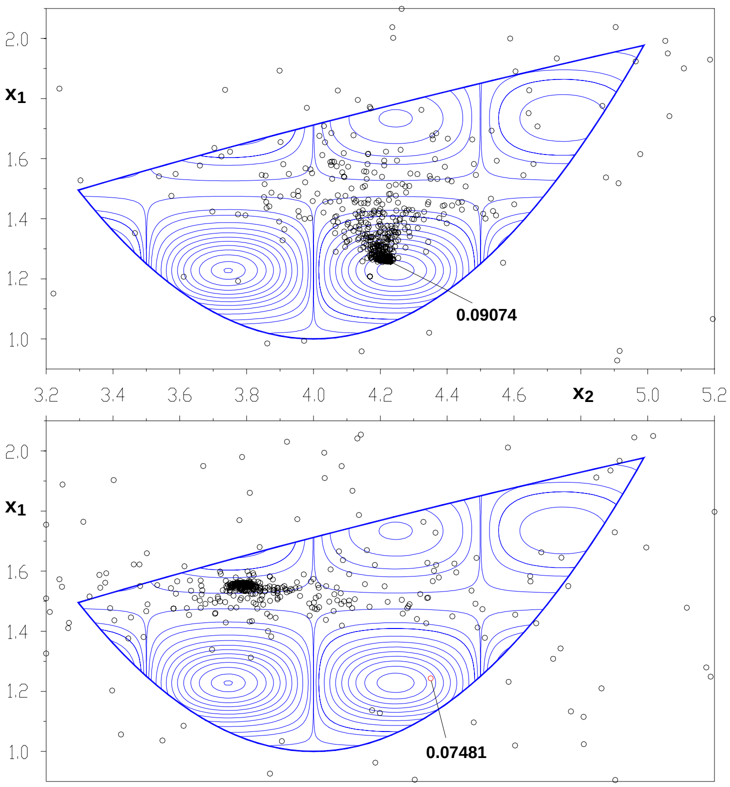


Fig. 2.5 Superimposed plots of the points generated during one run of E-PBIL + *vK*-Penalty (top) and UMDA + Repair 3 (bottom) for Test Problem 2.

The runs illustrated by the plots in Figs. 2.4 (and also in Figs. 2.5 and 2.6) were considered representative in that the best fitness found during the respective searches were very close to the average value recorded in Tables 2.1-3 (for Fig. 2.4 these fitness values were 5612 vs. 5607 and 5133 vs. 5125).

The same frames plotted superimposed in Fig. 2.4-up were animated and are available as animated GIFF files from the author or online at [14]. From these animations it can be seen that the best fit individual emerged (most likely following a repair operation) during the second generation and was preserved unchanged all the way to the end of the run. As the search progressed, the rest of the population slowly moved toward this best fit individual.

The animations also reveal that imposing the repair search to be performed along the line pointing in the direction of the closest feasible individual the displacement of the population parallel to the boundary of the feasible space is significantly diminished. One remedy towards

Table 2.2 Results obtained for 500 runs of test Problem 2 for M=50 and N=25 (known global optimum: 0.095825).

Algorithm	**Constraint handling technique**	**Best result**	**Average result**	**Worst result**
E-PBIL	*vK*-Pen.	0.095825	0.090999	0.019106
E-PBIL	*DK*-Pen.	0.095825	0.090982	0.018469
E-PBIL	*1K*-Pen.	0.095825	0.090354	0.029143
E-PBIL	Repair 1	0.095825	0.089778	0.025175
E-PBIL	Repair 2	0.095825	0.089659	0.028708
E-UMDA	*vK*-Pen.	0.095825	0.089549	0.027295
E-UMDA	*1K*-Pen.	0.095825	0.089199	0.037292
E-UMDA	*DK*-Pen.	0.095825	0.088951	0.027408
E-UMDA	Repair 2	0.095825	0.088107	0.025067
PBIL	*DK*-Pen.	0.095825	0.086992	0.022776
UMDA	*vK*-Pen.	0.095825	0.086738	0.019890
PBIL	*vK*-Pen.	0.095825	0.086693	0.026722
UMDA	*DK*-Pen.	0.095825	0.086221	0.027787
UMDA	*1K*-Pen.	0.095825	0.086214	0.021905
PBIL	*1K*-Pen.	0.095825	0.086196	0.024969
PBIL	Repair 1	0.095825	0.086046	0.023640
PBIL	Repair 2	0.095825	0.085759	0.036746
UMDA	Repair 2	0.095825	0.085409	0.024786
E-PBIL	Repair 3	0.095825	0.082726	0.001645
E-UMDA	Repair 1	0.095825	0.082144	0.024559
UMDA	Repair 1	0.095825	0.079079	0.018287
E-UMDA	Repair 3	0.095825	0.078868	0.013925
PBIL	Repair 3	0.095825	0.075557	0.012989
UMDA	Repair 3	0.095825	0.074724	0.016606

an increased exploration of the areas parallel to the boundaries of the feasible space (other than changing the line-repair strategy) can be to force (directly or indirectly) the components of the standard deviations vector to stay large during the first few generations.

As visible from Fig. 2.4–bottom, the UMDA algorithm with *1K*-Penalty (that was ranked last) had difficulties in maintaining a pool of feasible individuals in the population and was therefore unable to direct the search toward promising areas of the design space. The animated GIFF files generated using the same data as for Fig. 2.4-bottom also shows that the actual solution (labeled 5132.5) was generated during early generations, but no further exploration was performed in that same area [14].

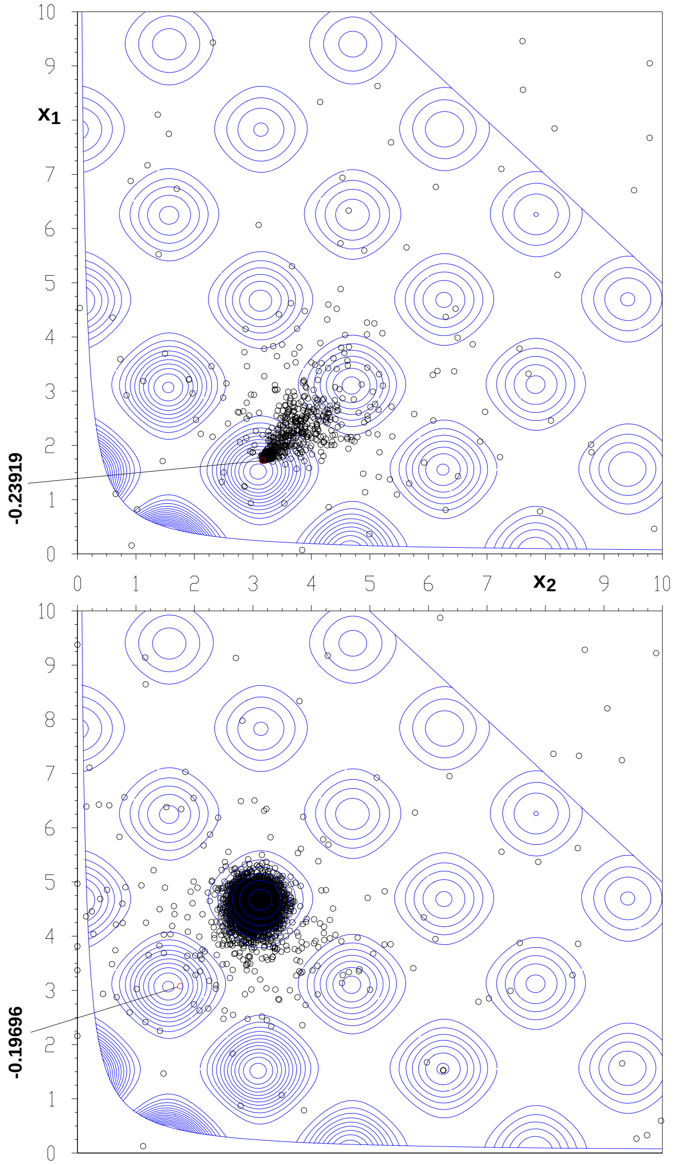


Fig. 2.6 Superimposed plots of the points generated during one run of E-PBIL + Repair 1 (top) and E-UMDA + *DK*-Penalty (bottom) algorithms over Test Problem 3 with n=2.

Problem 2 has its global maximum unbounded, and therefore the constraint handling technique employed has less effect upon the evolution of the population after several generations have passed, when more important become the hill climbing capabilities of the basic EDA employed. The elitist EDAs particularly the E-PBIL algorithm exhibits such traits and consequently performed better (although the favorable effect of a wider initial sampling of the landscape proved beneficial as the results in Table 2.2 show).

Table 2.3 Results obtained for 500 runs of Test Problem 3 with N=2 for M=50 and N=25 (known global optimum: 0.3649797).

Algorithm	Constraint handling technique	Best result	Average result	Worst result
E-PBIL	Repair 1	-0.3649797	-0.239162	-0.133003
E-PBIL	Repair 2	-0.3649797	-0.235662	-0.109429
E-PBIL	*1K*-Penalty	-0.3649796	-0.224695	-0.109429
E-PBIL	Repair 3	-0.3649793	-0.223351	-0.107983
E-PBIL	*DK*-Penalty	-0.3649722	-0.226574	-0.109418
E-PBIL	*vK*-Penalty	-0.3649683	-0.227550	-0.133230
PBIL	Repair 1	-0.3649352	-0.211658	-0.109323
E-UMDA	Repair 1	-0.3639721	-0.206205	-0.101459
UMDA	Repair 2	-0.3619513	-0.205715	-0.110912
PBIL	*vK*-Penalty	-0.3606485	-0.199515	-0.101611
E-UMDA	*vK*-Penalty	-0.3600499	-0.200429	-0.109429
UMDA	Repair 3	-0.3544603	-0.192459	-0.099642
UMDA	Repair 1	-0.3529160	-0.203255	-0.100988
E-UMDA	Repair 2	-0.3528012	-0.207182	-0.117033
UMDA	*1K*-Penalty	-0.3526836	-0.200672	-0.121185
PBIL	Repair 3	-0.3525017	-0.196533	-0.099643
PBIL	*DK*-Penalty	-0.3505644	-0.20042	-0.108292
PBIL	Repair 2	-0.3493383	-0.208814	-0.104155
E-UMDA	Repair 3	-0.3474485	-0.196391	-0.108942
E-UMDA	*1K*-Penalty	-0.3461925	-0.200399	-0.109136
UMDA	*vK*-Penalty	-0.3434922	-0.195404	-0.114145
UMDA	*DK*-Penalty	-0.3397125	-0.199832	-0.103815
PBIL	*1K*-Penalty	-0.3231532	-0.197422	-0.105956
E-UMDA	*DK*-Penalty	-0.3087370	-0.197603	-0.107092

Least performers were the non-elitist EDA algorithms using cloning repair (and repair in general). Repair operations have the effect of reducing the variability of the initial populations by forcing its individuals inside the feasible space.

Figs 2.5-top and the animated GIFF file available on [14] show the ascending path followed by the successive populations in case of the E-PBIL+ν*K*-Penalty algorithm leaded by the best fit individual. However, because of the standard deviations becoming too small, this ascent ended prematurely, suggesting again that the standard deviation should be kept at larger values longer periods of time. On the other hand, as Fig. 2.5-bottom shows, the run of the

UMDA algorithm with clone repair was trapped in the neighborhood of a bounded local-optima without being able to advance towards it.

Problem 3 was more difficult to solve due to the global optima being constrained and of the numerous local optima. Again the elitist E-PBIL algorithm performed better (see Table 2.3), this time in association with a line-search and crossover infeasible individual repair (remember that crossover repair is a variant of the line-search repair). It becomes evident that the favorable hill climbing characteristics of the E-PBIL algorithm were augmented by the boundary exploration capabilities provided by the repair method.

As happed before, and visible from Fig. 2.6-up and the animated GIFF file available on [14], the search lost momentum due to a premature reduction to very small values of the components of the standard deviation vector.

For this third test problem, the lowest ranked was the elitist UMDA algorithm coupled with a *DK*-Penalty. The sample run shows an interesting behavior, in which the best fit individual is trapped around of a local optima and the rest of the population swarms around another slightly lower optima. After viewing the animated files generated with the same data as in Fig. 2.6 below, it becomes evident that the swarming can continue forever because the standard deviation can neither go to zero nor increase sufficiently so that the swarming population can join with the best fit individual trapped on the neighboring local optima.

2.6 Conclusions

Two Estimation of Distribution Algorithm viz the Univariate Marginal Distribution Algorithm and a Population Based Incremental Learning Algorithm were tested in both elitist and non-elitist variants on solving 3 benchmark objective functions with constraints.

These test suggest that the function’s local landscape and the way the current population is distributed in this landscape should not dictate alone the probability distribution used in generating the new individuals. When normal distributions are used, forcing the standard deviation values to remain relatively large during a longer period of the search is likely to improve performance by avoiding “sinking” the population prematurely into a local optimum area. Another phenomenon that can be avoided by controlling the standard deviation values is the localized swarming in case of elitist algorithms applied to multimodal functions (as it was the case of test problem 3), when the best fit individual is trapped on one local optima while the rest of the population swarms around a neighboring lower optima.

Conversely, the same as the gradient value can be used as stopping criteria in first-order optimization algorithms, standard deviation values can be used as stopping criteria in EDAs. This was suggested by some of the numerical examples investigated, where part of the search was spent generating (almost) identical individuals due to standard deviation vector having its components approaching zero.

2.7 References

[1] Müßhlenbein H. and Paaß, G., (1996) “From Recombination of Genes to the Estimation of Distributions,” in Parallel Problem Solving from Nature IV, Springer, p. 178 187.

[2] Larrañaga, P. and Lozano, J. A., Eds., (2002) Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, *Kluwer Academic Publishers*.

[3] Larrañaga, P., (2002) “A review of Estimation of Distribution Algorithms,” in [2], p. 57 100.

[4] Paul T. K. and Iba H., (2002) “Linear and Combinatorial Optimizations by Estimation of Distribution Algorithms,” Proc. of the 9th MPS Symposium on Evolutionary Computation, IPSJ, Japan.

[5] Larrañaga, P., Etxeberria, R., Lozano, J.A. and Peña, J.(2000) “Optimization in Continuous Domains by Learning and Simulation of Gaussian Networks,” Proc. of the 2000 Genetic and Evolutionary Computation Conference, GECCO 2000, Las Vegas, NE, p. 201 204.

[6] Sebag, M. and Ducoulombier, A., (1998) “Extending population-based Incremental Learning to Continuous Search Spaces,” in Parallel Problem Solving from Nature V, *Springer*, p. 418 427.

[7] Michalewicz, Z., Dasgupta, D., Le Riche, R.G. and Schoenauer, M., (1996) “Evolutionary Algorithms for Constrained Engineering Problems,” Computers and Ind. Engr, Vol. 30, p. 851-870.

[8] Michalewicz, Z. and Schoenauer, M., (1996) “Evolutionary Algorithms for Constrained parameter Optimization Problems,” Evolutionary Computation, Vol. 4, p. 1-32.

[9] Coello Coello, C. A., (2002) “Theoretical and Numerical Constraint-Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art,” Computer Methods in Applied Mechanics and Engineering, Vol. 191, p. 1245-1287.

[10] Kuri, A., (1998) "A Universal Eclectic Genetic Algorithm for Constrained Optimization," Proc. of the 6th European Congress on Intelligent Techniques & Soft Computing, EUFIT'98, p. 518-522.

[11] Richardson, J.T., Palmer, M.R., Liepins, G. and Hilliard, M. (1989) “Some Guidelines for Genetic Algorithms with Penalty Functions,” Proc. of the 3rd Intl Conf. on Genetic Algorithms, p. 191-197.

[12] Michalewicz, Z. and Nazhiyath, G., (1995) “Genocop III: A Co-evolutionary Algorithm for Numerical Optimization Problems with Nonlinear Constraints,” Proc. of the 2nd IEEE Intl Conference on Evolutionary Computation, Perth, Australia, 29 Nov–1 Dec, 1995, Vol. 2, p.647-651.

[13] Koziel, S. and Michalewicz, Z., (1999) “Evolutionary Algorithms, Homomorphous Mappings and Constrained Parameter Optimization,” Evolutionary Computation, Vol. 7, pp. 19-44.

[14] http://www.auburn.edu/~simiope/CEC04

CHAPTER 3. VISUALIZATION OF MULTIVARIABLE OBJECTIVE FUNCTIONS BY PARTIAL GLOBAL OPTIMIZATION

Functions of the type z=F(x_1,…,x_n) where F are single-valued functions of n real variables cannot be visualized directly due to our inability to perceive dimensions higher than three. However, by projecting them down to 2 or 3 dimensions many of their properties can be revealed, like monotonicity, the existence of more than one minimum or maximum, or in case of constrained functions, the activity of the constrains. In this chapter a method to generate such projections is proposed, requiring successive global minimizations and maximizations of the function with respect to n-1 or n-2 variables. A number of numerical examples are given to show the usefulness of the method, particularly for optimization problems where there is a direct interest in the minimum or maximum domains of objective functions.

3.1. Introduction

In our attempt to analyze increasingly complex problems, visualization of multidimensional (multivariate) data can play an important role in both problem formulation and solution discovery. Therefore it continues to be an appealing research subject for many disciplines, including mathematics, statistics, computer graphics and operations research [10] [13] [22] [23]. The numerous visualization methods known so far are applicable to some but not all multidimensional objects encountered in sciences and engineering like: experimentally-obtained scattered data [3] [9] [28], hypergeometries [5] [6] [20] [21] [29] [40] [46], analytical functions [7] [14] [15] [26] [39] and inspection of the design space in optimization problems [16] [24] [25] [36] [42] [43] [44], artificial neural network design [12] [19] [33] [38] [41], etc.

The subject of the present chapter is the visualization of z=F(x_1,…,x_n) hypersurfaces, where F are single-valued functions of n real variables, possibly (but not limited to) objective functions encountered in optimization problems. As will be further seen, one benefit of the proposed approach is that once the plots are generated, they do not require a dynamic interactive environment for their display, thus the information therein becoming very easy to disseminate.

Functions of one variable F(x) can be straightforwardly represented in 2D space, where one dimension can be associated with the variable and the other dimension with the function value. As an extension, functions of two variables F(x_1, x_2), can be visualized as families of curves x_1=constant, x_2=constant or z=constant (also called level-curve or contour-plot diagrams). Another possibility is to plot their surfaces z=F(x_1,x_2) in parallel or perspective projection, which is equally intuitive since in the 3D Euclidean space, two dimensions can be associated to the variables and the third dimension to the function value.

Functions of three variables F(x_1,x_2,x_3) can also be represented graphically, an example being the triangular plots of the relative proportions of three ingredients in a mixture [11] (although the three variables are not independent because the proportions sum must be unity). Another possibility is to scan two of the variables, for example x_1 and x_2, within some prescribed limits, while maintaining constant the third variable, and generate a 3D surface or level-curve diagrams. If a number of such representations are generated for successive values of x_3, they can be displayed as computer animations where time has the role of the third variable.

For the visualization of functions of more than three variables the use of color plots and of suitable interactive computer-graphics software (possibly in a virtual reality setup) is the standard approach. Such software generates slices through the function's (hyper)surface [15], [42], [43], and although the use of computers and of Web-based hypertexts, like for example

[46], are increasingly common, the employed graphics lose effectiveness when viewed as still images.

In case of projection techniques (as well as of parallel coordinates), once the images are generated, they do not require the availability of a computer and of the appropriate software for visualization of the respective hyperobjects. Therefore such visualization methods are more appropriate when the end-results are supposed to be printed on paper in a book or a journal. However, so far, they were only applied to multidimensional geometries with simplifying characteristics: hypercubes, hyperspheres, and quadratic and cubic hypersurfaces [5] [6]. In the case of the parallel coordinates, the limitation to simple geometries is even more severe (points, lines and collection of such); moreover, the graphic representations obtained are strongly dependant on the order in which the parallel coordinates are labeled [10] [20] [21] [23].

3.2 Description of the Proposed Visualization Method

The main idea of the proposed visualization method is to perform global minimizations/maximization of the function with respect to all but 1 or 2 of the variables (a process further called n-1 or n-2 *partial-global optimization*), and to generate an appropriate number of point-sets that will be further used for producing 2D or 3D graphical representations. Earlier work of Papalambros and Wilde [30] employed partial minimizations but with respect to only one of the variables, while the remaining n-1 variables have been assigned arbitrary fixed values (and without resorting to any graphical representation), for the purpose of inspecting the design space of objective functions and studying their constraint activity. There are also coincidental similarities between the proposed technique and the

Maximum Intensity Projection (MIP) method used in medical imaging [27], which projects the maximum intensity values in a 3D volume data to a 2D plane.

Specifically, lets consider a multivariate function F($x_{1,}\ldots,x_n$) and some chosen variables' domains $x_i \in [x_{i\min,}...,x_{i\max}]$ (i=1..n, n≥2) (not explicitly side constraints of an optimization problem). One of the variables, for example x_1, can be scanned at a constant step within the interval $[x_{1\min},...,x_{1\max}]$ and a global minimization of F($x_{1,}\ldots,x_n$) performed at each step with respect to the remaining variables $x_2...x_n$; this *partial-global-minima function* will be further symbolized as $F_{\downarrow 2..n}(x_1)$ while the symbol $F_{\uparrow 2..n}(x_1)$ will be used for its *partial-global-maxima function* counterpart. In turn, x_1 will be named *scan variable* while the remaining $x_2...x_n$ will be named *search variables*.

The successive 2D points obtained (x_1, $F_{\downarrow 2..n}(x_1)$) with $x_{1\min} \le x_1 \le x_{1\max}$, can be plotted as a continuous curve on a two-dimensional graph. In a concise formulation this is equivalent to graphing the following function:

$$\begin{aligned} &\mathrm{F}_{\downarrow 2..n}(x_1) = global \min_{\mathrm{x}_2...\mathrm{x}_\mathrm{n}} \mathrm{F}(x_1...x_n) \\ &\text{subjected to } x_{j\min} \le x_j \le x_{j\max} \text{ with } j = 2...n \end{aligned} \tag{3.1}$$

which, as will be further shown, is the lower-bound of the 2D projection of function's hypersurface. Similarly, the 2D projection of the upper-bound of the same hypersurface results by plotting the following function:

$$\begin{aligned} &\mathrm{F}_{\uparrow 2..n}(x_1) = global \max_{\mathrm{x}_2...\mathrm{x}_\mathrm{n}} \mathrm{F}(x_1...x_n) \\ &\text{subjected to } x_{j\min} \le x_j \le x_{j\max} \text{ with } j = 2...n \end{aligned} \tag{3.2}$$

The total number of plots that can be generated by considering various scan variables in the above equations (3.1) and (3.2) is equal n, the number of variables of the objective function.

In order to project the given hypersurface down to 3D, two of the function's variables, for example x_1 and x_2, can be considered scan variables within the intervals $[x_{1\min},...,x_{1\max}]$ and

[$x_{2\min}$,...,$x_{2\max}$], and the following partial-global minima and partial-global maxima functions defined:

$$\begin{gathered} \mathrm{F}_{\downarrow 3..n}(x_1, x_2) = \mathit{global}\,\min_{\mathrm{x}_3...\mathrm{x}_\mathrm{n}} \mathrm{F}(x_1...x_n) \\ \text{subjected to } x_{j\min} \le x_j \le x_{j\max} \text{ with } j = 3...n \end{gathered} \tag{3.3}$$

$$\begin{gathered} \mathrm{F}_{\uparrow 3..n}(x_1, x_2) = \mathit{global}\max_{x_3...x_\mathrm{n}} \mathrm{F}(x_1...x_n) \\ \text{subjected to } x_{j\min} \le x_j \le x_{j\max} \text{ with } j = 3...n \end{gathered} \tag{3.4}$$

These functions can be plotted as level-curve diagrams or projected 3D surfaces, either separate or both on the same graph. When plotted in pair, the total number of graphs that can be generated by considering all possible combinations of scan and search variables equals C(n,2)=n(n-1)/2.

Projecting (hyper)surfaces down to 1D through global minimization and maximization is an intuitive processes as follows: Functions of any number of variables (1, 2 or more) can be projected on the z-axis (the function-value axis) by projecting infinitely many points on the function's (hyper)surface on this axis. As shown in Fig. 3.1 however, a substitute operation is to project onto the z-axis only the global minimum and maximum points. Obviously such 1D-projections are of limited benefit as they are equivalent to providing the actual numerical values of the global minimum and global maximum. They are important only in that they offer an intuitive starting ground for understanding more complex representations that make use of equations (3.1-4) above.

In the following paragraph it will be shown how z=F($x_{1,}\, x_2$) surfaces can be projected from 3D down to 2D by performing repeated partial-global minimizations and maximizations. The process will be further extended to projecting hypersurfaces or functions of more than two variables down to 3D or to 2D.

3.3 Numerical Examples

Example 1: In order to pave the way to higher dimensions and to highlight the benefits and limitation of the proposed visualization method, the following multimodal function of two variables will be considered first:

$$\begin{aligned}&\mathrm{F1}(x_1, x_2) = 3(1 - x_1)^2 \cdot exp\left[-x_1^2 - (x_2 + 1)^2\right] \\ &-10\left(\frac{x_1}{5} - x_1^3 - x_1^5\right) \cdot exp\left[-x_1^2 - x_2^2\right] - \frac{1}{3} exp\left[-(x_1 + 1)^2 - x_2^2\right]\end{aligned} \tag{3.5}$$

This function, known as MATLAB "peaks," is obtained by translating and scaling Gaussian distributions. It has 3 minimum and 3 maximum points as summarized in Table 3.1, of which the one labeled **min 1** is dominated by the other points of extrema and therefore harder to identify.

Traditionally, such a function of two variables can be visualized in several ways (see Figs. 3.1 through 3.4) and its multimodal characteristics revealed.

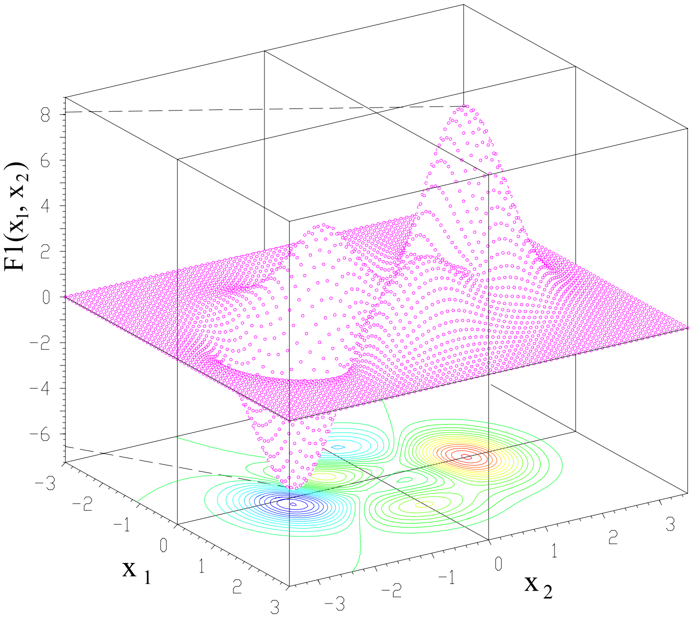


Fig. 3.1 MATLAB "peaks" function shown as a 61×71 points mapped on an opaque surface. Projecting all these points on the vertical axis is equivalent to projecting only the global minimum and maximum on the same axis.

Table 3.1 The maximum and minimum points of MATLAB "peaks" function.

	x_1	x_2	$F2(x_1,x_2)$
max 1	**-0.00932**	**1.58137**	**8.10621**
max 2	**-0.46002**	**-0.62920**	**3.77658**
max 3	**1.28568**	**-0.00485**	**3.59249**
min 1	**0.29645**	**0.32020**	**-0.06494**
min 2	**-1.34740**	**0.20452**	**-3.04985**
min 3	**0.22828**	**-1.62553**	**-6.55113**

Because they relate the most to the subject of this research, the projections on the x_1=constant or x_2=constant vertical planes, viz. the front-view and the side-view plots of the function's surface will be discussed further (Figs. 3.2 and 3.3). One way to generate such views is to project on the aforementioned vertical planes, equally-spaced "slices" through the function's surface. These successive slices can be oriented parallel (Figs. 3.2-a and b) or perpendicular (Figs. 3.3-a and b) to the projecting plane. In the former case, for both the front views (Figs. 3.2-a and b) and back views (not exemplified), the invisible lines can be eliminated using a suitable hidden line algorithm.

About the same amount of information as the parallel contour plots in Figs. 3.2-a and b (but rather without the hidden lines removed) can be obtained by projecting, on the same vertical planes, a number of points on the function's surface sampled in a regular $k_1 \times k_2$ grid (Figs. 3.2-c and d).

When the sampling points along a "slice" perpendicular to the projection-plane are infinitely numerous, such a distinction cannot be made and representations like those shown in Figs. 3.3-a and b are obtained. These plots bear exactly the same amount of information as the lower-bound and upper-bound representations in Figs. 3.3-c and d. What is important from the standpoint of the proposed visualization method is that such representations can be obtained by plotting together the following partial-global-minima and partial-global-maxima functions:

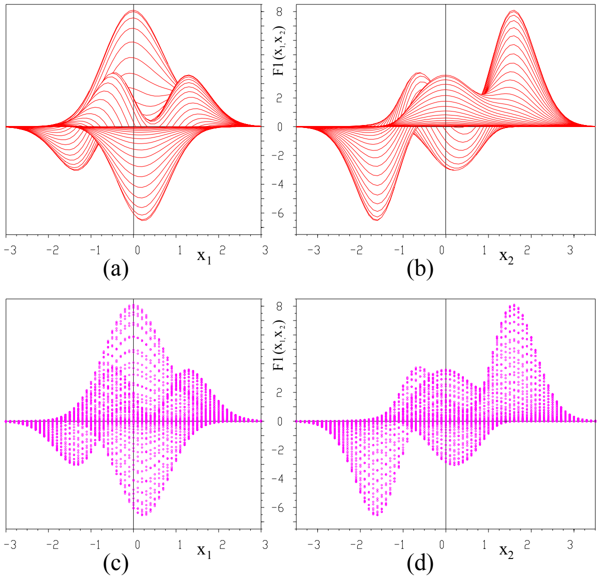


Fig. 3.2 Side views of F1(x_1,x_2) shown **(a)** as curves of constant x_2, **(b)** as curves of constant x_1 and **(c)** and **(d)** as point clouds mapped on a transparent function surface.

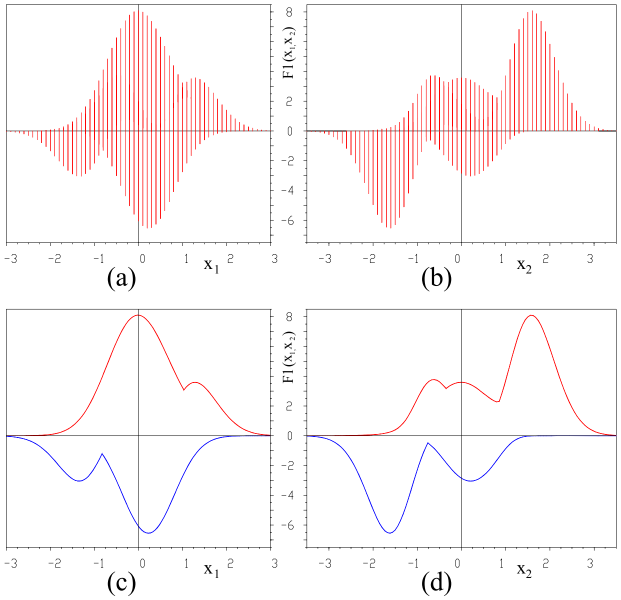
$$\begin{aligned} \mathrm{F1}_{\downarrow 2}(x_1) &= \mathit{global}\ \underset{x_2}{\mathit{min}}\, \mathrm{F1}(x_1, x_2) \\ \mathrm{F1}_{\uparrow 2}(x_1) &= \mathit{global}\ \underset{x_2}{\mathit{max}}\, \mathrm{F1}(x_1, x_2) \end{aligned} \tag{3.6}$$

for Fig. 3.3-c, and of

$$\begin{aligned} \mathrm{F1}_{\downarrow 1}(x_2) &= \mathit{global}\ \underset{x_1}{\mathit{min}}\, \mathrm{F1}(x_1, x_2) \\ \mathrm{F1}_{\uparrow 1}(x_2) &= \mathit{global}\ \underset{x_1}{\mathit{max}}\, \mathrm{F1}(x_1, x_2) \end{aligned} \tag{3.7}$$

for Fig. 3.3-d, with the benefit that a lesser number of function evaluations will be required, depending on the effectiveness of the optimization algorithm employed. For functions of more than 2 variables, a grid-search approach as previously described requires $k_1 \times k_2 \times \ldots \times k_n$ function evaluations, which for $n>3$ becomes prohibitively large and more advanced global-optimum searching techniques should be used.

The combined graphs of the partial-global minima and partial-global maxima functions of MATLAB “peaks” are also shown in Fig. 3.4, below and to the left of the level-curve plot. In this same figure, superimposed to the level-curve diagram of the original function, the values



(a) (b)

(c) (d)

Fig. 3.3 Side views of F1(x_1,x_2) shown **(a)** as lines of constant x_1, **(b)** as lines of constant x_2 and **(c)** and **(d)** as upper-bound and lower-bound contours only.

of the search variables as they result when evaluating $F1_{\downarrow 2}$, $F1_{\uparrow 2}$, $F1_{\downarrow 1}$ and $F1_{\uparrow 1}$ functions are also plotted. These points (noted throughout this dissertation $x_{1\downarrow},..x_{n\downarrow}$ and $x_{1\uparrow},..x_{n\uparrow}$) together with the respective function-values represent 3D curves mapped on the z=F1(x_1,x_2) surface, which project on the x_1=constant or x_2=constant vertical planes as the graphs of the partial-global minima and partial-global maxima functions in equations (3.6) and (3.7). These mapped curves or mapped surfaces in case of 3D projections, will be called *(projected) upper-bound* and *(projected) lower-bound paths*.

A fairly simple computer program written in C language that can be used for generating the values required for plotting the “silhouettes” in Figs. 3.3-c and d and of the corresponding upper/lower-bound paths is given in Appendix 1. The two nested *for*-loops therein (where the inner one does a grid-search) are the numerical equivalents of the plots in Fig. 3.2-a and b, with the difference that the grid-search records only the maximum and minimum heights along every slice through the function’s surface. By further nesting for loops, this simple algorithm

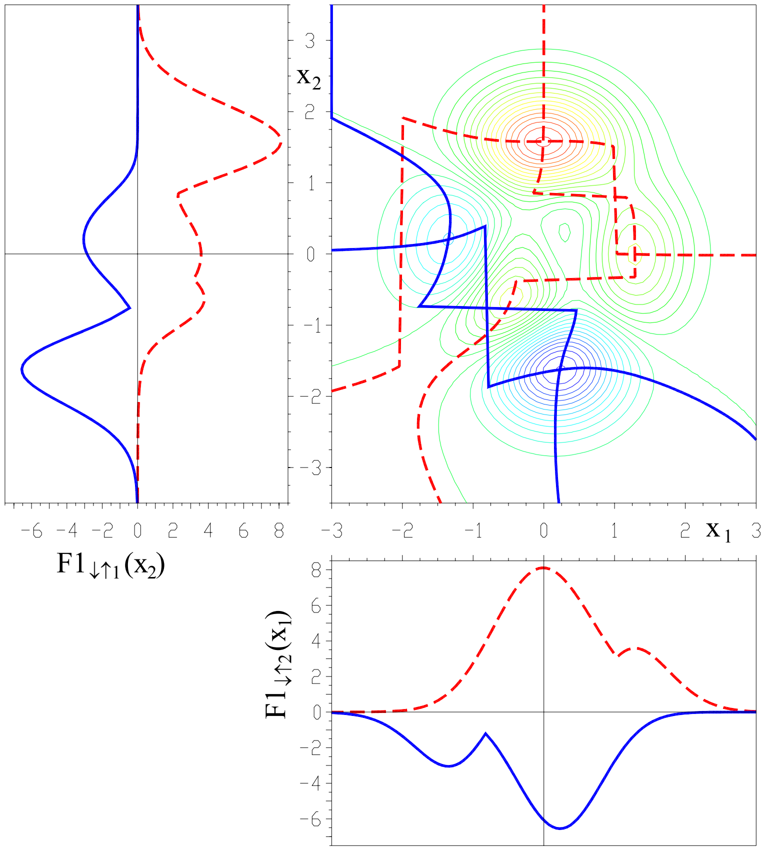


Fig. 3.4 Contour plot of MATLAB "peaks" function (middle) and plots of partial global minima and maxima functions $F1_{\uparrow\downarrow 2}(x_1)$ and $F1_{\uparrow\downarrow 1}(x_2)$. Also shown, superimposed on the contour lines, are the graphs of $x_2(x_1)$ and $x_1(x_2)$ as they result when evaluating $F1_{\uparrow\downarrow 2}(x_1)$ and $F1_{\uparrow\downarrow 1}(x_2)$ (the upper-bound and lower-bound paths). Notice how some of the jumps of these paths are associated with tangent discontinuities in the partial-global minima and maxima graphs.

can be extended to generate data for plotting functions of more than two variables, but its simplicity is quickly surpassed by the unacceptable long computation time (though it can be the starting point for more elaborate implementations). It has an intrinsic advantage, though, that is both the partial global minimum and the partial-global maximum are found during the same iteration.

A number of conclusions can be already drawn following this two-variable-function example:

1) As a consequence of specifically performing partial-global minimization and maximization of the function, the global minimum and global maximum will always be visible on the diagram (provided that the domains of the variables are chosen such that they include this point).

2) In the case of multimodal functions, it is essential that partial-global minimizations and maximizations be performed, and not some local searching operations, in order to obtain correct projections of the respective functions. This requirement makes a difference between the present (hyper)surface visualization method and the design studies called *explore optimum plots* in [31], employed to show how the optimum modifies as the values of one or two of the design variables are changed.

3) If the function is multimodal, the existence of more than one minimum/maximum point will be revealed, although the total number of minimum and maximum points will not be always visible on the diagram, since the process has an upper-leveling and a lower-leveling effect - see the **min 3** point in Table 3.1 or the Ackley-function example later in this chapter.

4) The precision with which the minimum and maximum point(s) are shown on the graph is dependent on the resolution with which the scan variable interval(s) has been sampled, and a tradeoff must be made between the resolution of the plot and the number of partial optimizations required.

5) The fact that a function is multimodal can also be disclosed by the “jumps” in the upper-bound and lower-bound paths, of which some, but not all such “jumps,” can be associated with tangent discontinuities of the partial-global minima and partial-global maxima graphs. Therefore these upper and lower-bound paths can provide additional information about the properties of the function to be visualized, and should also be recorded and plotted.

6) The appearance of various projections of the function’s (hyper)surface is dependent on the limits imposed of the search variables during the partial-global-optimization processes, as well as on the limits of the scan variables. On the other hand, by changing these limits, further insight into function’s properties can be revealed in that certain local minima and local maxima (previously overshadowed by more dominant extrema) can become visible.

Example 2: The first example of a multidimensional object that can be visualized by partial optimization is that of a unit hypersphere centered at the origin:

$$\sum_{i=1}^{n} \mathrm{x}_i^2 - 1 = 0\,. \tag{3.8}$$

Because of its simplifying characteristics, the projection to the space of any three of its variables will look the same (as will be seen shortly, no graphical representation will be required, the projections being easy to derive in analytical form and to further interpret). Consider for example the 3D space of x_1, x_2 and x_3. In order to obtain the projection of the above hypersphere onto this space, the following explicit function will be employed:

$$\mathrm{F2}(\mathrm{x}_2,...,\mathrm{x}_n) = \mathrm{x}_1 = \pm\sqrt{1 - \mathrm{x}_2^2 - \mathrm{x}_3^2 - \sum_{i=4}^{n} \mathrm{x}_i^2} \tag{3.9}$$

obtained by separating one of the variables (viz. x_1) in equation (3.8). Without resorting to any numerical optimization algorithm, it is easy to recognize that the corresponding partial-global minima and partial-global maxima functions:

$$\mathrm{F2}_{\downarrow 4..n}(\mathrm{x}_2, \mathrm{x}_3) = \underset{\mathrm{x}4...\mathrm{x}_n}{min}\left[\pm\sqrt{1 - \mathrm{x}_2^2 - \mathrm{x}_3^2 - \sum_{i=4}^{n} \mathrm{x}_i^2}\right] \tag{3.10}$$
$$\text{subject to}: -1 \le \mathrm{x}_i \le 1 \text{ with } i = 4...n$$

and

$$\mathrm{F2}_{\uparrow 4..n}(\mathrm{x}_2, \mathrm{x}_3) = \underset{\mathrm{x}4...\mathrm{x}_n}{max}\left[\pm\sqrt{1 - \mathrm{x}_2^2 - \mathrm{x}_3^2 - \sum_{i=4}^{n} \mathrm{x}_i^2}\right] \tag{3.11}$$
$$\text{subject to}: -1 \le \mathrm{x}_i \le 1 \text{ with } i = 4...n$$

can be reduced to the following analytical expressions:

$$F2_{\downarrow 4..n}(x_2, x_3) = -\sqrt{1 - x_2^2 - x_3^2} \tag{3.12}$$

$$F2_{\uparrow 4..n}(x_2, x_3) = +\sqrt{1 - x_2^2 - x_3^2} \tag{3.13}$$

which, as expected, are unit spheres centered at the origin.

The projection of the same hypersphere (3.8) down to 2D, say to the space of x_1 and x_2, can be done in two ways: One is to start from the same explicit function (3.9) and plot the corresponding partial-global minima and partial-global maxima functions of one-variable:

$$\begin{gathered} F2_{\downarrow 3..n}(x_2) = \underset{x4...x_n}{min}\left[\pm\sqrt{1 - x_2^2 - \sum_{i=3}^{n} x_i^2} \right] \\ \text{subject to: } -1 \le x_i \le 1 \text{ with } i = 4...n \end{gathered} \tag{3.14}$$

and

$$\begin{gathered} F2_{\uparrow 3..n}(x_2) = \underset{x4...x_n}{max}\left[\pm\sqrt{1 - x_2^2 - \sum_{i=3}^{n} x_i^2} \right]. \\ \text{subject to: } -1 \le x_i \le 1 \text{ with } i = 4...n \end{gathered} \tag{3.15}$$

The other approach is to project from 3D down to 2D the partial-global minima functions of two variables already available in equations (3.12) and (3.13). Either way the conclusion remains the same that is a unit hypersphere projects down to 2D as a unit circle:

$$F2_{\downarrow\uparrow 3..n}(x_2) = \pm\sqrt{1 - x_2^2} \ . \tag{3.16}$$

This is an intuitive result (although not in full agreement with [5]) which should strengthen ones confidence in the validity of the proposed method.

Example 3: Another example of a multivariate function to be visualized by partial optimization is the generalized Rosenbrock's function [45]:

$$F3(x_1...x_n) = \sum_{i=1}^{n-1}\left[100 \cdot \left(x_{i+1} - x_i^2\right)^2 + \left(1 - x_i\right)^2 \right] \tag{3.17}$$

better know in its n=2 form under the name of “banana” function - a classic test objective function used since the early days of computational optimization in assessing the performance of optimization algorithms [34]. It will be further shown that the highly acclaimed, narrow, parabolic-shaped valley that makes “banana” function difficult to minimize by steepest-descent-type algorithms, is also present in its multivariate form, and that the same valley harbors the global minimum point $x_1=x_2=\ldots x_n=1$.

In order to investigate the properties of function (3.17) via 3D projections, the following partial-global extrema functions have been defined:

$$\begin{aligned} &F3_{\downarrow 3..n}(x_1, x_2) = \underset{x_3 \ldots x_n}{\textit{global min}}\, F3(x_1 \ldots x_n) \\ &\text{subject to } x_{j\min} \le x_j \le x_{j\max} \text{ with } j = 3 \ldots n \end{aligned} \tag{3.18}$$

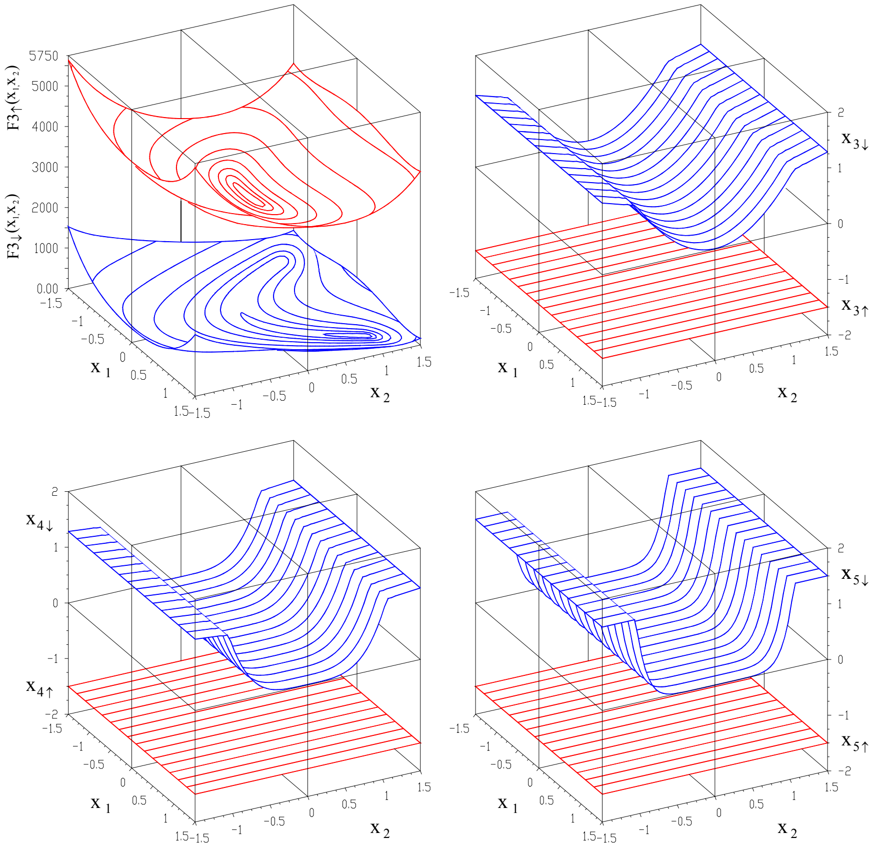


Fig. 3.5 Projection of the generalized Rosenbrock’s function of 5 variables on the 3D space (x_1,x_2,z) for $-1.5 \le x_j \le 1.5$ (j=1..5) together with the graphs of the upper-bound and lower-bound paths $x_{3\downarrow}(x_1,x_2)$, $x_{4\downarrow}(x_1,x_2)$, $x_{5\downarrow}(x_1,x_2)$ and $x_{3\uparrow}(x_1,x_2)$, $x_{4\uparrow}(x_1,x_2)$, $x_{5\uparrow}(x_1,x_2)$.

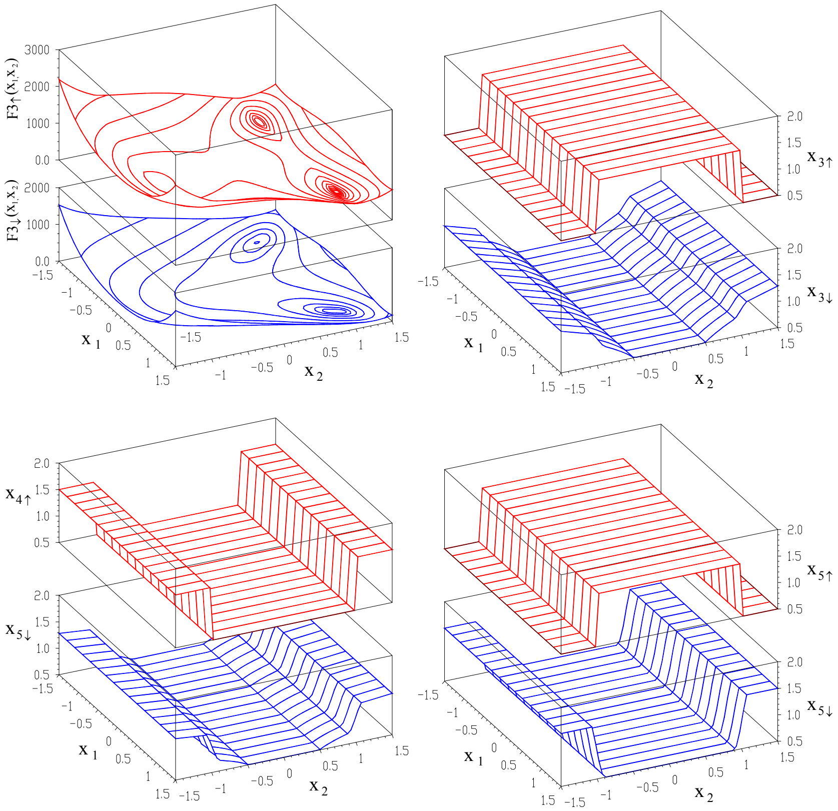


Fig. 3.6 Same graphs as in Fig. 2.5 generated for the search variables x_3, x_4 and x_5 restricted to the interval [0.5,1.5].

$$\begin{aligned} &\mathrm{F3}_{\uparrow 3..n}(\mathrm{x}_1,\mathrm{x}_2)=\mathit{global}\max_{\mathrm{x}_3\ldots\mathrm{x}_n}\mathrm{F3}(\mathrm{x}_1\ldots\mathrm{x}_n) \\ &\text{subject to } \mathrm{x}_{j\min}\le \mathrm{x}_j\le \mathrm{x}_{j\max} \text{ with } j=3\ldots n \end{aligned} \tag{3.19}$$

Their plots for the case of n=5 and -1.5[x_j[1.5 (j=1..n) together with the corresponding graphs of the minimum and maximum recorded values of the search variables (the upper-bound and lower-bound paths) $x_{3\downarrow\uparrow}$, $x_{4\downarrow\uparrow}$ and $x_{5\downarrow\uparrow}$ are given in Fig. 3.5. The subroutine used for evaluating the respective partial-minima/maxima functions combines a 35×35×35 grid-search step followed by Powell's Direction Set Algorithm [32].

In reference [37] are available the plots of the remaining nine 3D projections of generalized Rosenbrock's function with n=5, over the same interval [-1.5, 1.5] of the variables. All these graphs confirm the existence of parabolic valley along which the global minimum point x_1=…x_5=1 is located, similar to the n=2 variant of the function.

Since away from this global minimum point the function is convex and monotonically increasing, all partial-global maxima occur for the least admissible values of the search variables (i.e. -1.5). However, in spite of the recursive association of the variables in equation (3.17), the 2D upper-bound and lower-bound paths do not exhibit a repeating pattern.

It is therefore to be expected that for different intervals of the search variables, the same function will project differently down to 3D or to 2D. To exemplify this, the graphs in Fig. 3.6 have been produced for the case of the search variables being restricted to a smaller interval, viz. [0.5, 1.5]. The changes are surprising in that the 3D projections of Rosenbrock's function of 5 variables exhibits not one but two minima when subject to the mentioned side constraints.

Example 4: Another example of a multivariate function considered was Bäck's generalization [4] of Ackley's function [1]:

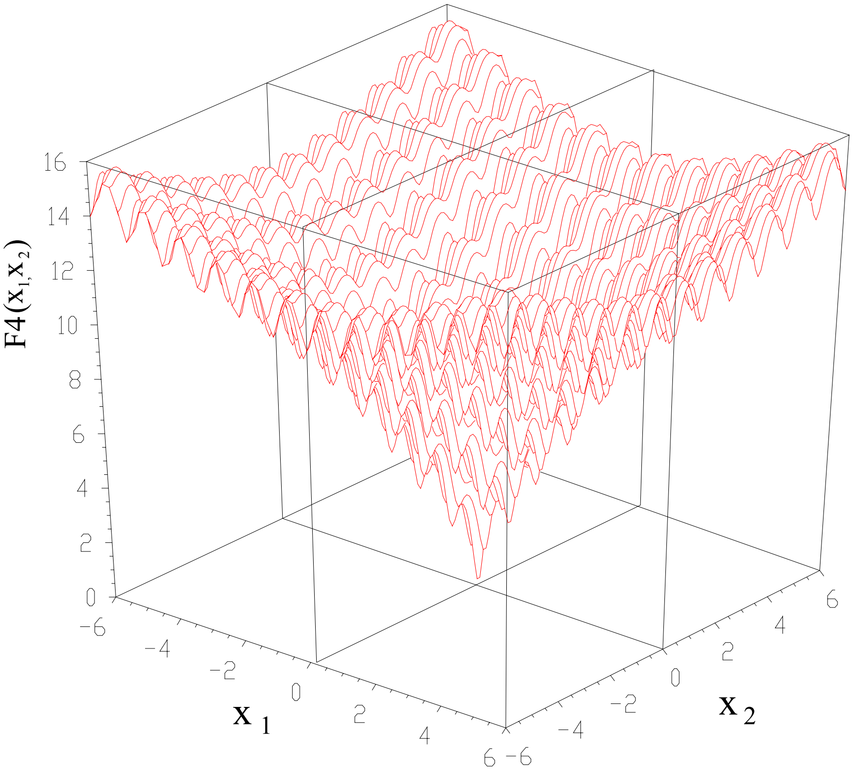


Fig. 3.7 Surface of Ackley's function of two variables.

$$\mathrm{F4}(\mathrm{x}_1...\mathrm{x}_n) = 20\left[1 - exp\left(-\frac{1}{2}\sqrt{\frac{1}{n}\sum_{i=1}^{n}\mathrm{x}_i^2}\right)\right] - exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi\,\mathrm{x}_i)\right) + e \qquad (3.20)$$

This is a highly multimodal, test objective function commonly used in rating Genetic Algorithms. Irrespective of n, its global minimum equals 0, and occurs for $x_1=\ldots x_n=0$. A plot of F4 for n=2 is shown in Fig. 3.7. It will be shown that the same overall appearance is also present in the 2D and 3D projections of Ackley's function of more than 2 variables.

As mentioned earlier, functions of 3 variables can be visualized through animations of plots generated for successive values of the third variable being kept fix. If the respective surfaces are represented on the same graph as shown in Fig. 3.8 for the case of Ackley's function with n=3, then it becomes obvious that the lower and upper envelopes of these surfaces can be obtained by retaining only the minimum and the maximum values of the function along the verticals through every (x_1, x_2) grid point. This is another example of the proposed visualization technique confirming known results and being complementary to existing approaches.

In order to further inspect the propertied of function (3.20), the following partial-global minima and partial-global maxima functions:

$$\begin{aligned}&\mathrm{F4}_{\downarrow 3..n}(\mathrm{x}_1,\mathrm{x}_2) = global\,\min_{\mathrm{x}_3...\mathrm{x}_n}\mathrm{F4}(\mathrm{x}_1...\mathrm{x}_n)\\&\text{subject to } -6 \le \mathrm{x}_j \le 6 \text{ with } j = 3...n\end{aligned} \qquad (3.21)$$

$$\begin{aligned}&\mathrm{F4}_{\uparrow 3..n}(\mathrm{x}_1,\mathrm{x}_2) = global\,\max_{\mathrm{x}_3...\mathrm{x}_n}\mathrm{F4}(\mathrm{x}_1...\mathrm{x}_n)\\&\text{subject to } -6 \le \mathrm{x}_j \le 6 \text{ with } j = 3...n\end{aligned} \qquad (3.22)$$

have been plotted over [-6, 6]×[-6, 6] for the case of for n=10 (Figs. 9-a and b). The subroutine used in the evaluation of $\mathrm{F4}_{\uparrow 3..n}$ and $\mathrm{F4}_{\downarrow 3..n}$ combines a standard Genetic Algorithm [4] followed by the same Powell's algorithm.

The recorded values of the upper-bound paths have a sinusoidal variation as shown in Fig. 3.10 while the lower-bound paths are all equal $x_{3\downarrow}=x_{4\downarrow}=\ldots x_{10\downarrow}=0$, irrespective of the values of the scan variables. It is therefore safe to conclude that for any *n* and for any domains of the function's variables that include the global-minimum point, the 3D projection of generalized Ackley's function has the following analytical expression:

$$F4_{\downarrow 3..n}(x_1, x_2) = 20\left[1 - exp\left(-\frac{1}{2}\sqrt{\frac{x_1^2 + x_2^2}{n}}\right)\right] - exp\left(\frac{\cos(2\pi x_1) + \cos(2\pi x_2) - 2}{n} + 1\right) + e \qquad (3.23)$$

which explains the resemblance between the graph of $F4_{\downarrow 3..n}$ in Fig. 3.8-b and Ackley's function of 2 variables in Fig. 3.7.

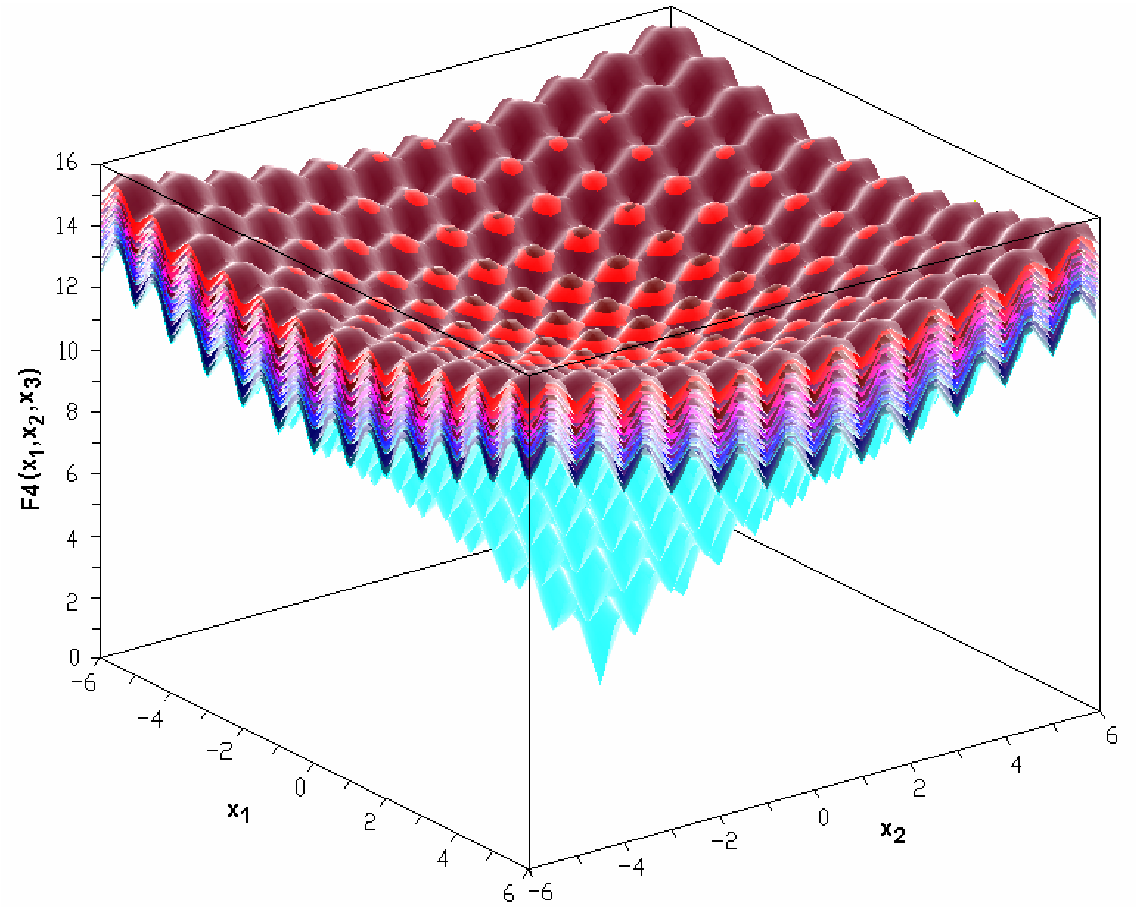


Fig. 3.8 Superimposed frames of an animated representation of Ackley's function of 3 variables with time corresponding to the third variable $x_3 \in [-6..6]$. The lower envelope of these surfaces correspond to $x_3=0$, while the upper envelope is the result of the intersection of several of the upper surfaces.

Interesting results were further obtained by graphically displaying the one variable partial-global minima function:

$$F4_{\downarrow 2..n}(x_1) = \underset{x_2..x_n}{\mathit{global\ min}}\, F4(x_1...x_n) = 20\left[1 - \mathit{exp}\left(-\frac{1}{2}\sqrt{\frac{x_1^2}{n}}\right)\right] - \mathit{exp}\left(\frac{\cos(2\pi x_1) - 1}{n} + 1\right) + e \tag{3.24}$$

and partial-global maxima function:

$$F4_{\uparrow 2..n}(x_1) = \underset{x_2..x_n}{\mathit{global\ max}}\, F4(x_1...x_n) \tag{3.25}$$

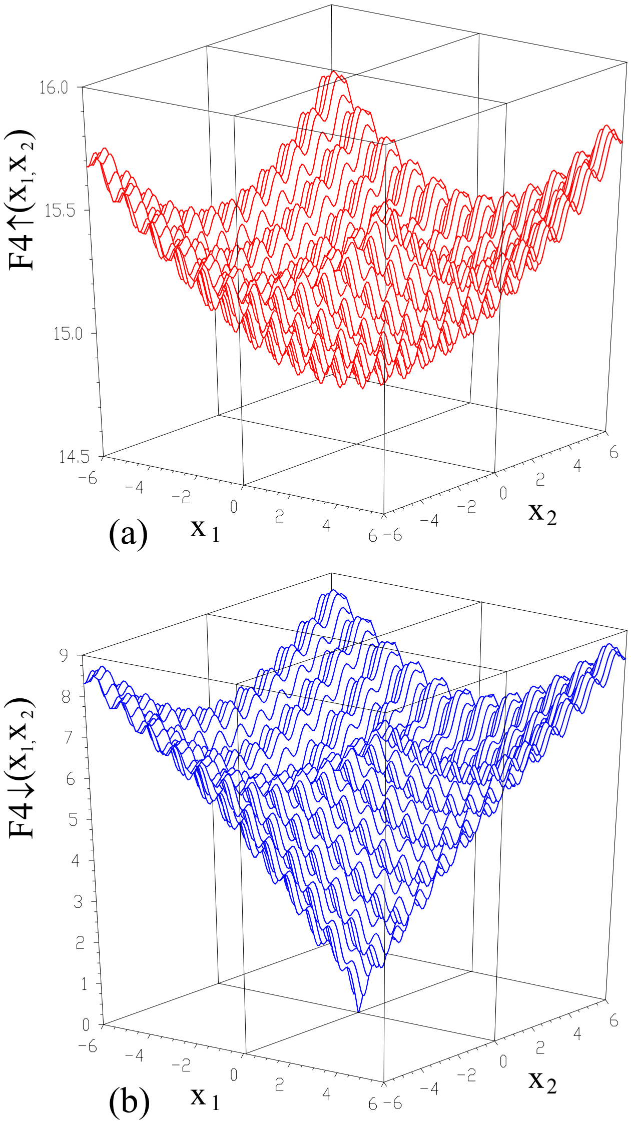


Fig. 3.9 Surface plots of the partial-global maxima $F4_{\uparrow 3..n}$ (**a**) and partial-global minima $F4_{\downarrow 3..n}$ (**b**) functions of generalized Ackley function of 10 variables.

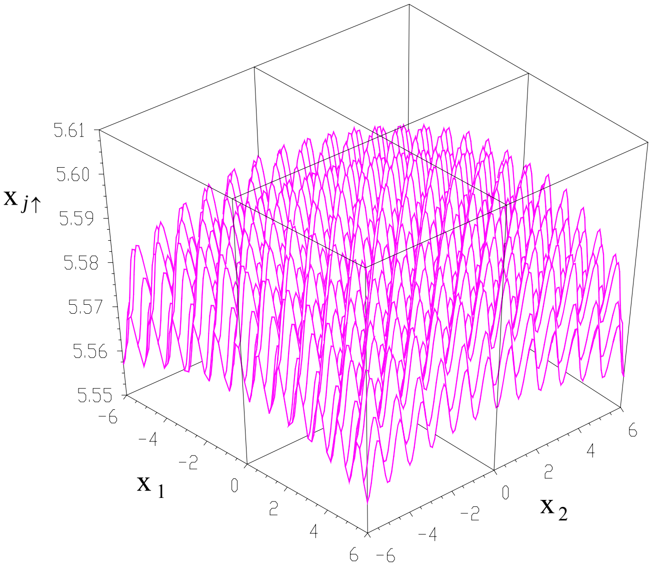


Fig. 3.10 The upper-bound path $x_{j\uparrow}(x_1,x_2)$ (j=3..10) corresponding to the diagram in Fig. 2.9-a.

for different n between 1 and 10. It is to be noticed that for n=1 equation (3.24) actually becomes the one dimensional variant of Ackley's function. The graphs in Figs. 3.9 and 3.11-a reveal that as n increases, the upper and lower envelopes of the 3D and 2D projections of Ackley's function smoothen and flatten. Same as for the 3D projections, the one dimensional upper-bound paths have a sinusoidal variation of decreasing amplitude as n increases (Fig. 3.11-b) while the corresponding lower-bound paths are all constant and equal to zero.

Furthermore, because F4($x_1..x_n$) is symmetric with respect to all variables, it can be inferred that the projection of the function's hypersurface onto any x_i=constant vertical plane or on any (x_i, x_j, z) space will look the same; conversely, for any scan variable(s), the partial minima functions in equations (3.23) and (3.24) will have the same analytical expressions.

These conclusions, now evident, are however less obvious without performing the partial-global minimizations and maximizations based projections shown above.

Example 5: It is known that in optimum design, "the knowledge of a family of good solutions is far more important than obtaining an isolated optimum" [2]. Therefore objective function visualization is always desirable in the design process since it provide an overview

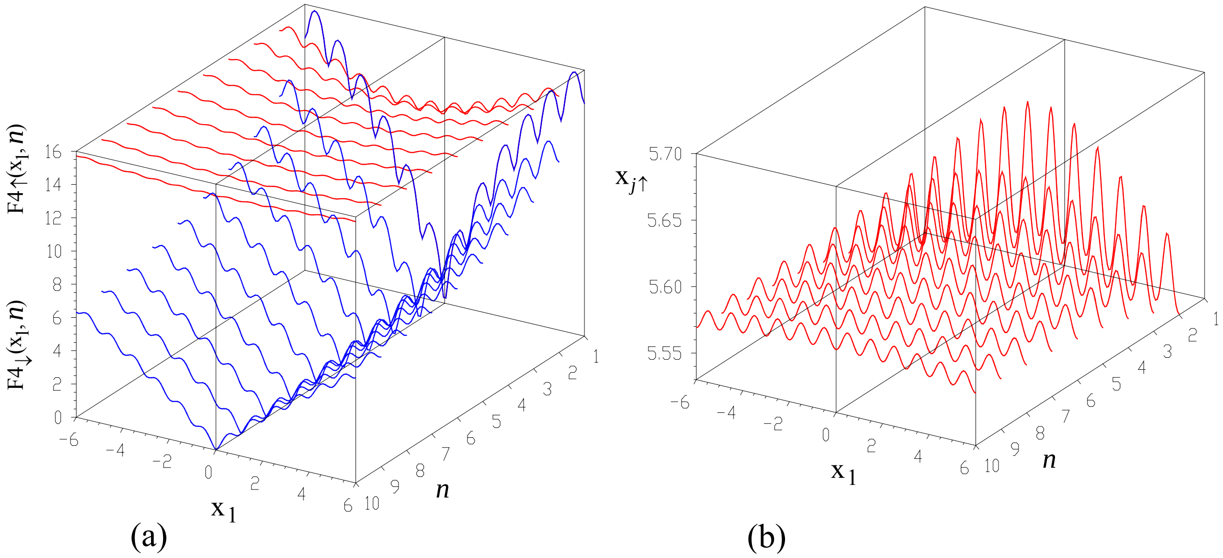


Fig. 3.11 Surface plot of the partial maxima and partial minima functions F4$_{\uparrow 2..n}$ and F4$_{\downarrow 2..n}$ for the case of n=10 **(a)** and plot of the corresponding upper-bound paths $x_{j\uparrow}(x_1)$ with j=2..10 **(b)**.

upon the minimum or maximum domains and can reveal sensitiveness of the optimum found or if it is bounded. Furthermore, parametric studies and constraint activity analyses when performed, are easier to interpret when the results are available in graphical form. This last example, based on problem number 5 in reference [18], is intended to shows how constraint-activity analyses can be perform through graphical representations of the partial-global minima of the given objective function.

Consider the problem of minimizing:

$$\begin{aligned}&F5(x_1,x_2,x_3)=(x_1^2+x_2^2+x_3^2)/2-(x_1+x_2+x_3)\\&\text{subject to:}\\&g_1=x_1\geq 0,\quad g_2=x_2\geq 0,\quad g_3=x_3\geq 0\quad\text{and}\\&g_4=1-x_1-x_2-x_3\geq 0.\end{aligned}\tag{3.26}$$

The global optimum equals -5/6=-0.8333 and occurs for x_1=x_2=x_3=1/3 for which constraint g_4 is active. It is to be noticed that both the function and the constraints are symmetric with respect to the design variables, which will fortuitously limit the maximum number of 3D projection plots that need to be examined.

Using Brent's algorithm [8], the partial-global minima diagram of F5 subject to g_1, g_2 and g_3 only has been generated first (Fig. 3.12). The graphs shows that F5 is monotonically strictly decreasing over the considered interval (Fig. 3.12-a), while the values of the search variables corresponding to these partial minima (the lower-bound paths) are constant and equal to 1 (Fig. 3.12-b).

When constraint g_4 is applied, the appearance of 3D projection of the objective function changes (Fig. 3.13-a), and the same the appearance of the lower-bound paths (Fig. 3.13-b). It is interesting to notice that the global optimum point appears inside the partial-global minima graph and not somewhere on the boundaries. This is due to overshadowing and of the lower-leveling/upper-leveling effect and the proposed dimension-reduction technique has. Therefore care should be exercised when interpreting constraint-activity analysis plots of multivariate functions in that the change in location of the global optimum with the application/suspension of constraint(s) must be studied. One should not expect to see the bounded optimum located on one of the boundaries of the projected multivariate function as in the case with constrained

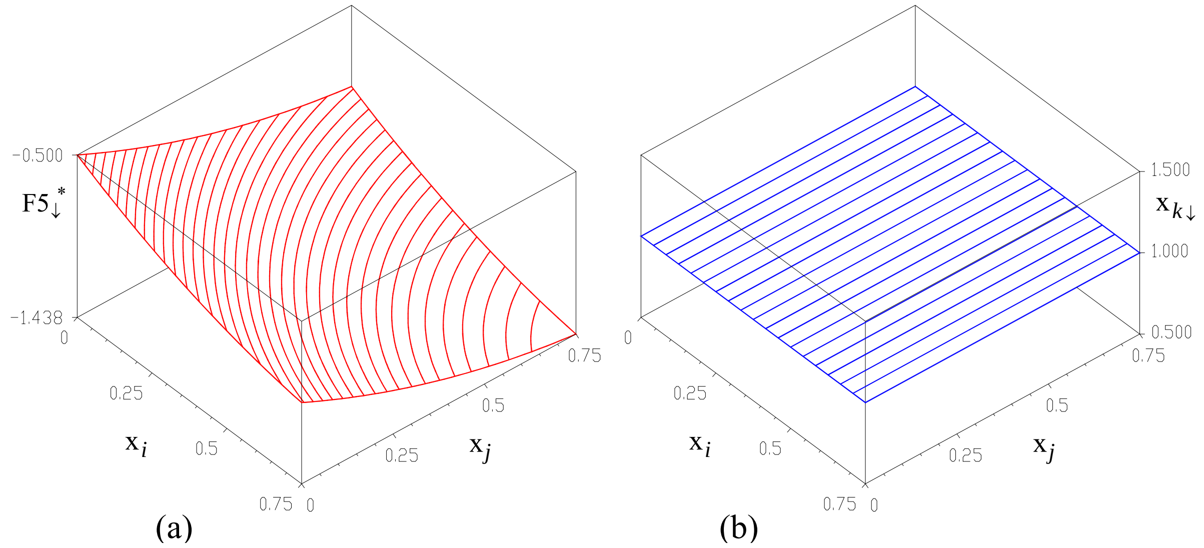


Fig. 3.12 Surface plot of the partial global minima function(s) $F5_{\downarrow k}(x_i,x_j)$ with i,j,k=1..3 and $i \neq j \neq k$ subject to g_1, g_2 and g_3 **(a)** and of the corresponding lower-bound path(s) $x_{k\downarrow}$ **(b)**.

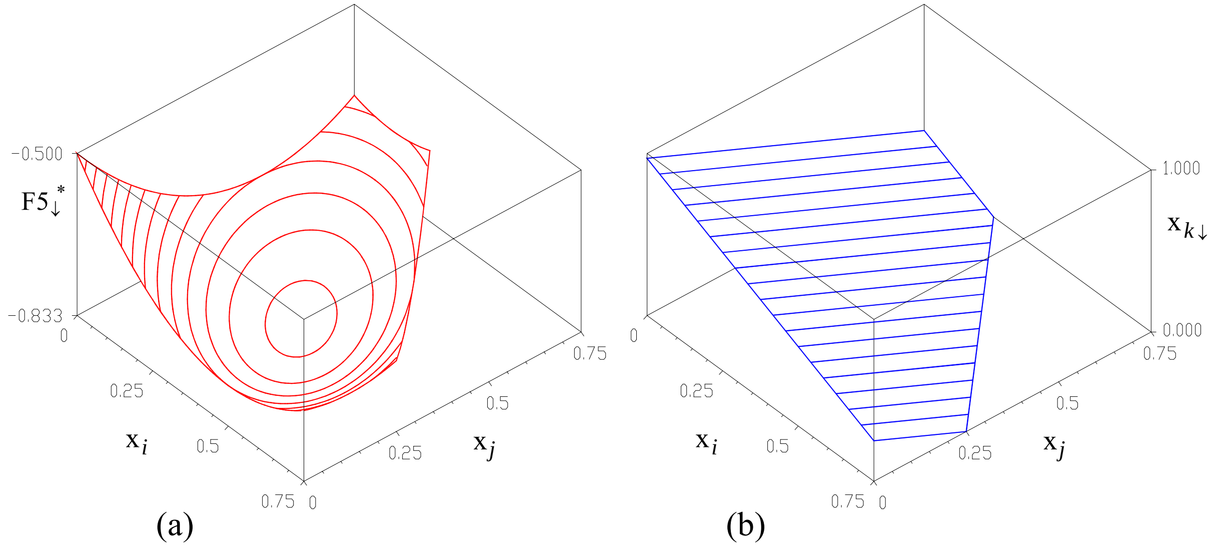


Fig. 3.13 Surface plot of the partial global minima function(s) $F5_{\downarrow k}(x_i,x_j)$ with i,j,k=1..3 and $i \neq j \neq k$ subject to g_2, g_3, g_4 and g_5 **(a)** and of the corresponding lower-bound path(s) $x_{k\downarrow}$ **(b)**.

objective functions of two variables.

It is agreed that monotonicity analysis techniques [31] can provide similar results through inspection of the objective function and of the constraints. When such techniques can be applied (if the function and the constraints are available in analytical form), the visualization technique proposed can be used for checking the results and for suggestively presenting them, particularly to an inexperienced audience.

3.4 Conclusions

A technique for visualizing $z=F(x_1,\ldots,x_n)$ hypersurfaces with direct application to inspecting the design space of objective functions and constraint-activity analyses was presented. It requires repeated partial-global minimization and/or maximization of the function with respect to all but 1 or 2 of the design variables (the scan variables) and plotting (preferably on the same graph) the resulting global minima and global maxima and of the corresponding values of the search variables.

Solving multiple-objective optimization problems using the ε-constraint method [17] can also benefit from the use of the visualization methods presented. Similarly, *what-if* studies and *worst-case-scenario* analyses, commonly done in practice, can be performed and presented more suggestively using the proposed techniques.

In Chapter 4 an optimum mechanism design example are presented, where the design space of the problem has been visualized through partial-global minimizations and some very useful characteristics of the respective mechanism revealed. For this particular design problem, meaningful scan variables proved to be the actual x and y coordinates of the working plane of the mechanism. Another example of application of the method is presented in Chapter 5, where the problem of finding the optimum number of teeth of an automatic transmission used in automobiles is aided by visualizing the design space of a mixed-integer optimization problem of eight variables.

When performed upon some properly chosen test objective functions, the repeated partial-global optimizations can be used to verify the speed and robustness of different optimization subroutines. For example, if the resulting partial-global minima and partial-global maxima plots have noisy, irregular appearances with spikes or sudden discontinuities, it is likely that the optimization subroutine employed converged prematurely and its parameters must be readjusted or the subroutine discharged.

On the other hand, if the tangent and curvature discontinuities on the partial-global minima/maxima graphs are not caused by premature convergences of the searching algorithm, then it is a sign that the original objective function is either non-smooth, multimodal or that some of the constraints change their activity as the scan variables change their values. In such cases further investigations are required to make sure that the observed discontinuities are accompanied by jumps of the upper-bound or lower-bound paths as discussed earlier with reference to Fig. 3.4.

It has been shown that there are function visualization problems for which the respective partial-global minima and partial-global maxima can be derived in closed form, either by inspection or using the methods of calculus. Other problems may require a significant computational effort (depending on dimensionality), while other problems can be insolvable. It is believed though that with the advent of very high-speed personal computers, and the intense research underway in the field of Evolutionary Computation, the number of practical problems that could benefit from the use of the proposed visualization technique should grow larger. The fact that the procedure is well suited to parallel processing should also encourage its use in practice.

A desirable implementation of the proposed visualization technique would be that in which the computer generates the data required to plot some (if not all) of the partial-global minima and partial-global maxima graphs to be interpreted later on, after a longer, unattended run of the computer. Such a software implementation would necessary require the availability of several searching subroutines (either deterministic, stochastic or combinations of the two) the user can choose from the one considered best suited for the problem at hand.

Finally it is mentioned that, although not a global optimization technique, the proposed method can have some merit with respect to global optimum finding, in that an *n*-variable problems can be reduced to repeated optimization of a function of *n*-2 variables.

3.5 References

[1] Ackley D H (1987) *A Connectionist Machine for Genetic Hill Climbing*. Kluwer Academic Publishers, Boston-MA.

[2] Arabeyre J, Steiger J, Teather W (1969) "The Airline Crew Scheduling Problem: A Survey," Transportation Science, Vol. 3(2), p. 140-163.

[3] Asimov D (1985) "The Grand Tour: A Tool for Viewing Multidimensional Data," SIAM Journal on Scientific and Statistical Computing Vol. 6(1), p. 128-143.

[4] Back T (1997) *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford University Press, New York.

[5] Bajaj C L (1990) "Rational Hypersurface Display," ACM Computer Graphics Vol. 24(2), p. 117-127

[6] Banchoff T T (1986) "Visualizing Two-Dimensional Phenomena in Four-Dimensional Space," In: Wegeman E J, DePriest D J (eds) Statistical Image Processing and Graphics. Dekker, New York, p. 187-202.

[7] Bhaniramka P, Wenger R, Crawfis R (2000) "Isosurfacing in higher dimensions," In: Proceedings of the IEEE Visualization Conference, Oct 8-13 Salt Lake City-UT, p. 267-273+566

[8] Brent R P (1973) *Algorithms for Minimization Without Derivatives*, Prentice Hall, Upper Saddle River-NJ

[9] Carr D B, Nicholson W L, Littlefeld R J, Hall D L (1986) "Interactive Color Display Methods for Multivariate Data," In: Wegeman E J, DePriest D J (eds) Statistical Image Processing and Graphics. Dekker, New York, p. 215-250

[10] Chen J X, Wang S, (2001) "Data Visualization: Parallel Coordinates and Dimension Reduction," Computing in Science and Engineering, Vol. 3(5), p. 110-113

[11] Cornell J A (2002) *Experiments with Mixtures,* Wiley, New York

[12] Craven M W, Shavlik J (1992) "Visualization Learning and Computation in Artificial Neural Networks," International Journal on Artificial Intelligence Tools Vol. 1(3): 399-425

[13] Encarnação J L, Lindner R, Schlechtendahl E G (1990) Computer Aided Design. Fundamentals and System Architecture. Springer, New York

[14] Feiner S, Beshers C (1990) "Visualizing *n*-Dimensional Virtual Worlds with *n*-Vsion," Computer Graphics, Vol. 24(2), p. 37-38

[15] Feiner S, Beshers C (1990) "Worlds within Worlds: Metaphors for Exploring *n*-Dimensional Virtual Worlds," in: Proceedings of the 3rd ACM SIGGRAPH Symposium on User Interface Software and Technology, Oct 3-5 Snowbird-UT, p. 76-83.

[16] Grüne L, Metscher M, Ohlberger M (1999) "On Numerical Algorithm and Interactive Visualization for Optimal Control Problems," Computing and Visualization in Science Vol. 1(4), p. 221-229.

[17] Haimes Y Y, Lasdon L S, Wismer D A (1971) "On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization," IEEE Transaction on System, Man and Cybernetics, Vol. 1(3), p. 296-297.

[18] Hansen P, Jaumard B, Lu S H (1989) "Some Further Results on Monotonicity in Globally Optimal Design," ASME Journal of Mechanisms, Transmissions and Automation in Design, Vol. 111(3), p. 345-352

[19] Hinton G E, Sejnowski T J, Ackley D H (1984) “Boltzman Machines: Constraint Satisfaction Networks that Learn,” Technical Report CMU-CS-84-119, Carnegie-Mellon University, Pittsburgh-PA

[20] Inselberg A (1985) “The Plane with Parallel Coordinates,” The Visual Computer 1(2): 69-91

[21] Inselberg A, Dimsdale B (1990) Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry,” In: Proceedings of the 1st IEEE Conference on Visualization, Oct 23-26 San Francisco-CA, p. 361-378.

[22] Jones C V (1996) *Visualization and Optimization*, Kluwer Academic Publishers, Boston-MA.

[23] Jones C V (1998) Interactive Transactions of OR/MS. http://catt.bus.okstate.edu/jones98/

[24] Keiper J, Wickam-Jones T (1994) “Design Tools for Visualization and Optimization,” ORSA Journal on Computing, Vol. 6(3), p. 273-277.

[25] Messac A, Chen X (2000) “Visualizing the Optimization Process in Real-time Using Physical Programming,” Engineering Optimization, Vol. 32(6), p. 721-747.

[26] Mihalisin T, Timlin J, Schwengler J (1991) “Visualizing Multivariate Functions, Data and Distributions,” IEEE Computer Graphics & Applications, Vol. 11(3), p. 28-35.

[27] Napel S, Marks M P, Rubin G. D, Jeffrey R B, Dake M D, Enzmann D R, McDonnell C H, Song S M (1992) “CT Angiography Using Spiral CT and Maximum Intensity Projections,” Radiology, Vol. 185(2), p. 607-610.

[28] Nielson G M, Foley T A, Hamann B, Lane D (1991) “Visualizing and Modeling Scattered Multivariate Data,” IEEE Computer Graphics & Applications, Vol. 11(3), p. 47-55.

[29] Noll A M (1967) “A Computer Technique for Displaying n-Dimensional Hyperobjects,” Communications of the ACM, Vol. 10(8), p. 469-473.

[30] Papalambros P Y, Wilde D J (2000) *Principles of Optimal Design: Modeling and Computation*, Cambridge University Press.

[31] Parkinson A R, Balling R J (2002) “The OptdesX Design Optimization Software,” Structural and Multidisciplinary Optimization, Vol. 23(2), p 127-139.

[32] Press W H, Flannery B P, Teukolsky S, Vetterling W T (1989) *Numerical Recipes*. Cambridge University Press

[33] Pudmenzky A (1998) “A Visualization Tool for N-Dimensional Error Surfaces,” Australian Conference on Neural Networks ACNN'98, Feb 11-13, University of Queensland, Brisbane, Australia

[34] Rosenbrock H H (1960) “An Automatic method for finding the greatest or least value of a function,” The Computer Journal, Vol. 3(3), p. 175-184.

[35] Rossnick S, Laub G, Braeckle R, Bachus R, Kennedy D, Nelson A, Dzik S, Starewicz A (1986) “Three dimensional display of blood vessels in MRI,” In: Proceedings of the IEEE Computers in Cardiology Conference, New York, p. 193-196

[36] Shaffer C A, Knill D L, Watson LT (1998) “Visualization for Multiparameter Aircraft Designs,” In: Proceedings of the IEEE Visualization Conference, Oct 18-23 Triangle Research Park-NC, p. 491-494+575

[37] Simionescu P A, Beale D (2004) “Visualization of Multivariable (Objective) Functions by Partial Global Optimization,” The Visual Computer Journal, Vol. 20(10), p. 665 - 681.

[38] Streeter M, Ward M, Alvarez S A (2001) “N^{2}VIS: An Interactive Visualization Tool for Neural Networks,” In: Proceedings of the International Society for Optical Engineering (SPIE) Visual Data Exploration and Analysis VIII, Jan 22-23 San Jose-CA, 4302 p. 234-241.

[39] Van Wijk J J, Van Liere R (1993) “HyperSlice: Visualization of Scalar Functions of Many Variables,” In: Proceedings of the IEEE Visualization Conference, Oct 25-29 San Jose-CA, p. 119-125.

[40] Wegenkittl R, Löffelmann H, Gröller E (1997) “Visualizing the Behavior of Higher Dimensional Dynamical Systems,” In: Proceedings of the IEEE Visualization Conference, Oct 19-24 Phoenix-AZ, p. 291-296+569.

[41] Wejchert J, Tesauro G (1990) “Neural Network Visualization,” In: Touretzky D S (ed) Advances in Neural Information Processing Systems. Morgan Kaufmann, San Mateo-CA, p. 465-472.

[42] Winer E H, Bloebaum C L (2001) “Visual Design Steering for Optimization Solution Improvement,” Structural and Multidisciplinary Optimization Vol. 22(3), p. 219–229.

[43] Winer E H, Bloebaum C L (2002) “Development of Visual Design Steering as an Aid in Large-Scale Multidisciplinary Design Optimization,” Part I & II, Structural and Multidisciplinary Optimization, Vol. 23(6), p. 412-435.

[44] Wright H, Brodlie K, David T (2000) “Navigating High-Dimensional Spaces to Support Design Steering,” In: Proceedings of the IEEE Visualization Conference, Oct 8-13 Salt Lake City-UT, p 291-296+569

[45] Yao X, Liu Y, Lin G (1999) “Evolutionary Programming Made Faster,” IEEE Transactions on Evolutionary Computation, Vol. 3(2), p 82-102.

[46] Wolfram Research (2004) Math-World, http://mathworld.wolfram.com/topics/MultidimensionalGeometry.html

CHAPTER 4. GENERAL SOLUTIONS FOR THE OPTIMUM DESIGN OF SLIDER-ROCKER AND OSCILLATING-SLIDE ACTUATORS

The planar slider-rocker and oscillating-slide mechanisms are widely used for converting the rectilinear displacement provided by a hydraulic, pneumatic or electric linear motor, into swinging motion of an output member rocker. Using the visualization technique presented in Chapter 3 some new properties of these mechanisms are revealed, like the existence of well-defined optimum solutions that occur in pairs and of particular geometric configurations associated with these optima. For aiding the design process, parametric chart and performance charts are provided, which allow a rapid selection of the dimensions of the mechanism that generate an imposed displacement of the output member, while simultaneously ensuring maximum motion transmission efficiency.

4.1. Introduction

Slider-rocker and oscillating-slide mechanisms are the two inversions of the slider-crank mechanism with input translating members. These mechanisms have very many applications in robotics, aerospace, automotive, agricultural and earth moving machinery etc. The input member can be a hydraulic or pneumatic linear motor as well as a screw drive, rack-and-pinion or solenoid. The output member can be attached to, or be the actual element of the landing gear of an aircraft, of the steering-system of a truck or tractor, a door that opens automatically, a robotic or excavator arm etc. [3].

The synthesis of such a mechanisms for the generation of a prescribed swinging amplitude of the output member with a given displacement of the input member can be carried relatively

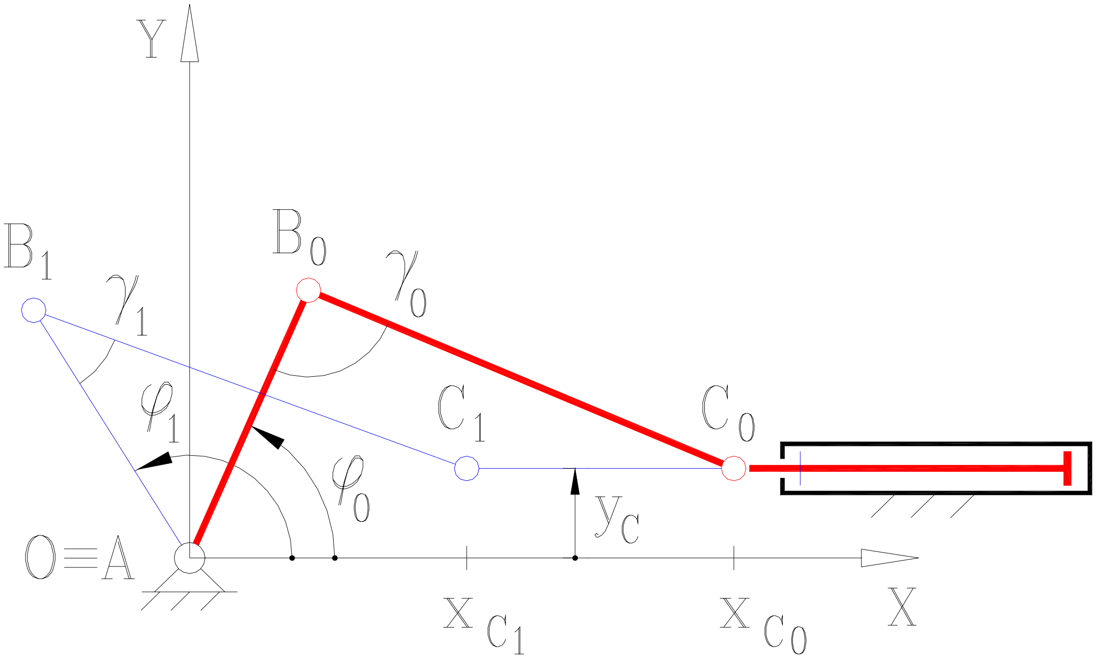


Fig. 4.1 Slide-rocker mechanism shown in two extreme positions (x_{C0}, φ_0) and (x_{C1}, φ_1).

easy graphically, facilitated by the fact that mechanism's loop is a simple triangle [6]. However, obtaining a good motion transmitting efficiency is not guaranteed, and a trial-and-error search must be carried out until the transmission angle (noted γ) remains in-between some minimum and maximum limits, usually greater than 45° and less than 135° or within 90°±45° (an equivalent formulation which emphasizes that the transmission angle should vary as little as possible from the ideal value of 90°). If a self-return of the output member is ensured due to weight or some other active forces, values of the transmission angle within the limits 90°±60° can be considered acceptable.

In this chapter the capabilities of the slider-rocker and oscillating-slide planar mechanisms were investigated for the generation of a given maximum swing of the output member, while simultaneously ensuring a minimum possible deviation from 90° of the transmission angle.

4.2. Synthesis of the Slider-Rocker Mechanism

Given a slider-rocker mechanism (Fig. 4.1), a $\Delta\varphi=\varphi_1-\varphi_0$ displacement of the output member is required to be generated with a given displacement of the slider corresponding to joint C moving horizontally between the limits x_{C0} and x_{C1}. For the sake of generality, a unit value of the maximum stroke of the input member will be further considered (i.e. $x_{C0}-x_{C1}=1$), the actual dimensions of the real mechanism being obtained by multiplying the normalized link-lengths with the actual stroke of the chosen linear actuator.

With the notations in Fig. 4.1 and for the two imposed relative positions of the input and output members (x_{C0},φ_0) and (x_{C1},φ_1), the synthesis equations of the mechanism are:

$$\begin{aligned}(x_{B0}-x_{C0})^2+(y_{B0}-y_{C0})^2&=BC^2\\(x_{B1}-x_{C1})^2+(y_{B1}-y_{C1})^2&=BC^2\end{aligned}\tag{4.1}$$

where

$$\begin{aligned}x_{B0}&=AB\cdot\cos\varphi_0 \qquad y_{B0}=AB\cdot\sin\varphi_0\\x_{B1}&=AB\cdot\cos\varphi_1 \qquad y_{B1}=AB\cdot\sin\varphi_1\end{aligned}.\tag{4.2}$$

Substituting (4.2) into (4.1), and for $y_C=y_{C0}=y_{C1}$ results:

$$BC^2=x_{C0}^2-2AB\cdot x_{C0}\cos\varphi_0-2AB\cdot y_C\sin\varphi_0+AB^2+y_C^2\tag{4.3}$$

$$BC^2=x_{C1}^2-2AB\cdot x_{C1}\cos\varphi_1-2AB\cdot y_C\sin\varphi_1+AB^2+y_C^2\tag{4.4}$$

By subtracting these two equations and substituting $x_{C1}=x_{C0}-1$ and $\varphi_1=\varphi_0+\Delta\varphi$, the normalized length of the rocker is obtained:

$$AB=\frac{0.5\cdot[x_{C0}^2-(x_{C0}-1)^2]}{x_{C0}\cos\varphi_0-(x_{C0}-1)\cdot\cos(\varphi_0+\Delta\varphi)+y_C[\sin\varphi_0-\sin(\varphi_0+\Delta\varphi)]}\tag{4.5}$$

For given values of the design variables φ_0, x_{C0} and y_C and for an imposed maximum displacement of the rocker $\Delta\varphi$, the normalized lengths AB and BC (Fig. 4.1) can be determined using equations (4.5) and (4.3) respectively.

As can be seen, the problem has a triple infinity of solutions, and it is therefore possible to select the values of φ_0, x_{C0} and y_C such that supplementary conditions upon link length ratios, ground-joint relative disposition or limits of the transmission angle can be satisfied. In the followings, only the last of these possible requirements will be consider i.e. the transmission angle γ to have a minimum deviation from the ideal value of 90° over the entire working range of the mechanism, so that good motion transmission efficiency is ensured. For a current displacement x_C of the piston, the angle γ can be calculated with:

$$\cos\gamma(x_C) = \frac{AB^2 + BC^2 - x_C^2 - y_C^2}{2AB \cdot BC}, \tag{4.6}$$

derived by applying Cosine Law in the ABC triangle, and in the right-angle triangle of hypotenuse AC built on the OX axis (Fig. 4.1). Apart from the extreme positions, the configuration of the mechanism for which x_C equals zero will also induce a critical value (maximum or minimum) of the transmission angle. Therefore this third position must also be taken into consideration if joint C crosses the vertical axis OY.

In order to find the configurations for which γ has minimum variation about 90°, the following *min-max* problem in the variables φ_0, x_{C0} and y_C has been defined:

$$\begin{aligned}
&\text{minimize}\\
&\text{F1}(\varphi_0, x_{C0}, y_C) = \max\{c_0, c_1, c_e\}\\
&\text{where}\\
&c_0 = |\cos\gamma(x_{C0})|\\
&c_1 = |\cos\gamma(x_{C1})|\\
&c_e = \begin{cases} |\cos\gamma(0)| & \text{if } \; x_{C0} \cdot x_{C1} < 0 \\ 0 & \text{if } \; x_{C0} \cdot x_{C1} \geq 0 \end{cases}
\end{aligned} \tag{4.7}$$

where the output member stroke $\Delta\varphi = \varphi_1 - \varphi_0$ is considered known and γ is calculated with the above equation (4.6). After several numerical experiments, it proved necessary to check that the vector loop ABC maintains the same orientation (i.e. the cross products $\mathbf{AB_0} \times \mathbf{B_0C_0}$ and

$\mathbf{AB_1 \times B_1C_1}$ have the same sign) during the working range of the mechanism. If this requirement is not fulfilled, the two extreme positions cannot be attained without breaking the joints, and therefore the value of the objective function must be penalized in some way. One can observe that for $\Delta\varphi$ approaching zero, the global minimum of $F1(\varphi_0, x_{C0}, y_C)$ also approaches zero, which correspond to $\gamma(x_{C0})=\gamma(x_{C1})=90°$. For nonzero displacements $\Delta\varphi$ of the output member, the global minimum of F1 will correspond to the minimum possible deviation of the transmission angle γ from 90°.

By using $\cos\gamma$ instead of the actual transmission angle in the expression of the objective function F1, the inverse cosine function is not repeatedly called, thus saving on CPU time. This is particularly beneficial when partial minima plots and design charts are generated as will be further explained.

In order to study the properties of the objective function F1, a systematic inspection of its design space has been performed. Using the method described in Chapter 2 of visualizing multivariate functions and hypersurfaces by partial-global optimization, a partial minima function in two variables has been defined:

$$G1(x_{C0}, y_C) = \mathit{arccos}\left(\underset{\varphi_0}{\text{global min}}\, F1(x_{C0}, y_C, \varphi_0) \right) \tag{4.8}$$

where the global minima of one variable has been calculated using a grid search followed by Brent’s algorithm [2]. This function has been studied graphically and the level curve diagrams in Fig. 4.2 generated for some practical values of the output member stroke $\Delta\varphi$. These graphs reveal that the design space of objective function F1 is symmetric about the vertical line $x_{C0}=0.5$. Further investigation show that for any $\Delta\varphi$ value, there are two global optimum mechanism solutions with one the mirror image of the other relative to y=0 vertical line.

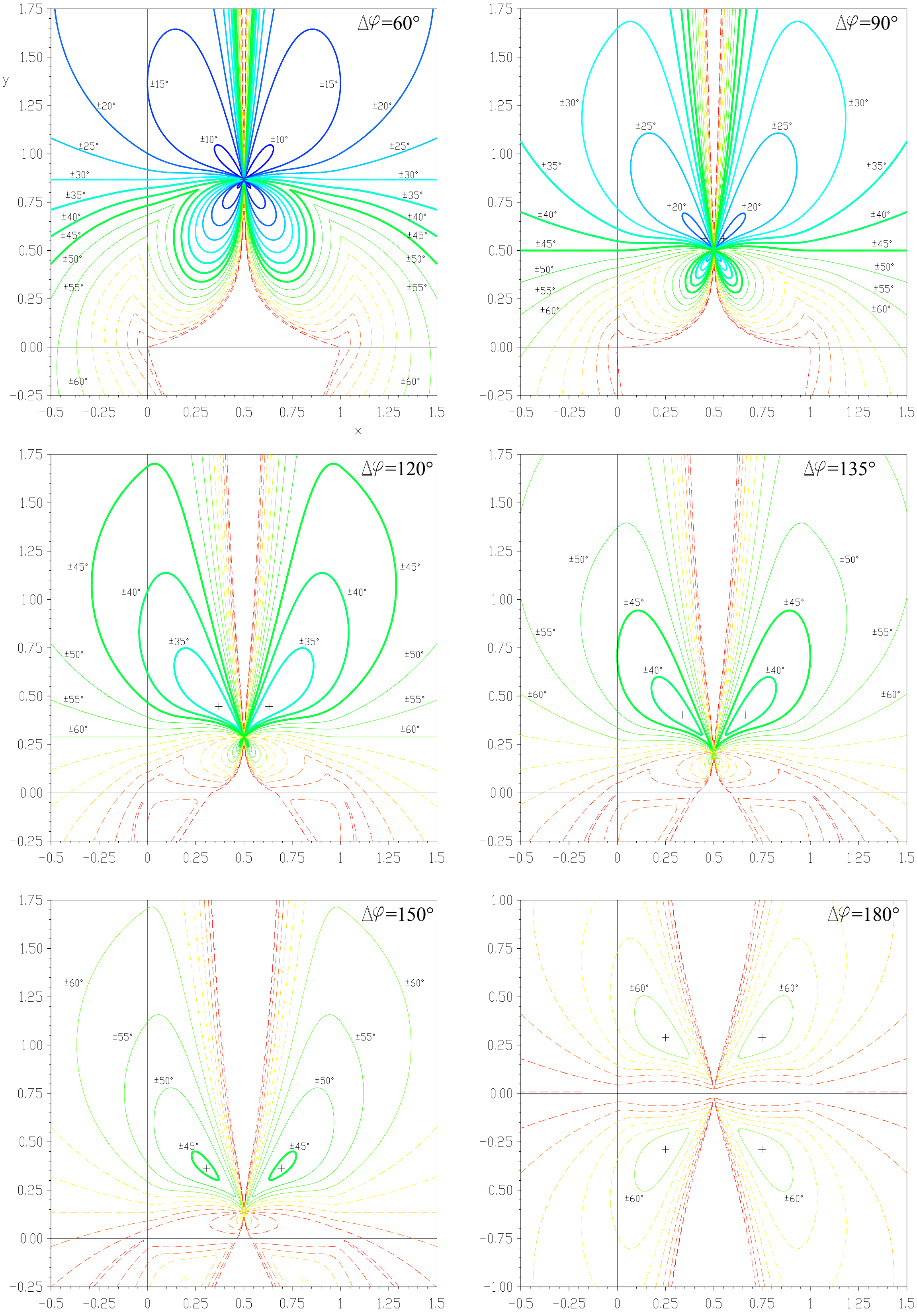


Fig. 4.2 Contour plots of G1(x_{C0},y_C) for different values of the output member stroke Δφ (global optima are marked with a cross "+").

Table 4.1 Optimum slider-rocker dimensions for some common swinging angles Δφ.

$\Delta\varphi$	x_{C0}	y_C	**AB**	**BC**	φ_0	γ
60°	0.50000	0.86603	0.70009	0.62039	97.8741	90±8.3°
90°	0.55006	0.56253	0.45298	0.51238	83.6436	90±19.0°
120°	0.63015	0.44623	0.37466	0.50724	69.3715	90±31.5°
135°	0.66353	0.40294	0.35254	0.50815	62.3633	90±37.9°
150°	0.69338	0.36276	0.33889	0.50708	55.2062	90±44.4°
180°	0.75000	0.28867	0.33072	0.50518	40.8934	90±57.3°
186.3°	0.76192	0.27345	0.33198	0.50480	44.1796	90±60.0°
190	0.76911	0.26433	0.33321	0.50458	36.1375	90±61.6°
200	0.78886	0.23952	0.33841	0.50398	31.3971	90±65.8°

For this reason, Table 4.1, where several global optimum slider-rocker mechanism solutions are gathered, only set of parameters is given, specifically only one set of φ_0 values.

Although for Δφ=60° the partial global minima graph G1(x_{C0},y_C) shows the existence of only one optimum (according to the next paragraph this is valid for any Δφ<77°), in actuality they are two optima. As explained in Chapter 2, this phenomenon is due to the fact that when projecting multimodal function from higher dimensions down to three dimensions, some minima or maxima can become obscured by other extrema located along the same projecting line.

Since the slider-rocker mechanisms that generate large swinging amplitudes at the output member have many potential applications, for example as orientation mechanism for antennas, solar panels, surveillance cameras, spotlights etc. (Fig. 4.3), the particular cases of Δφ=120° and Δφ=180° swing angles of the output members will be discussed in more details.

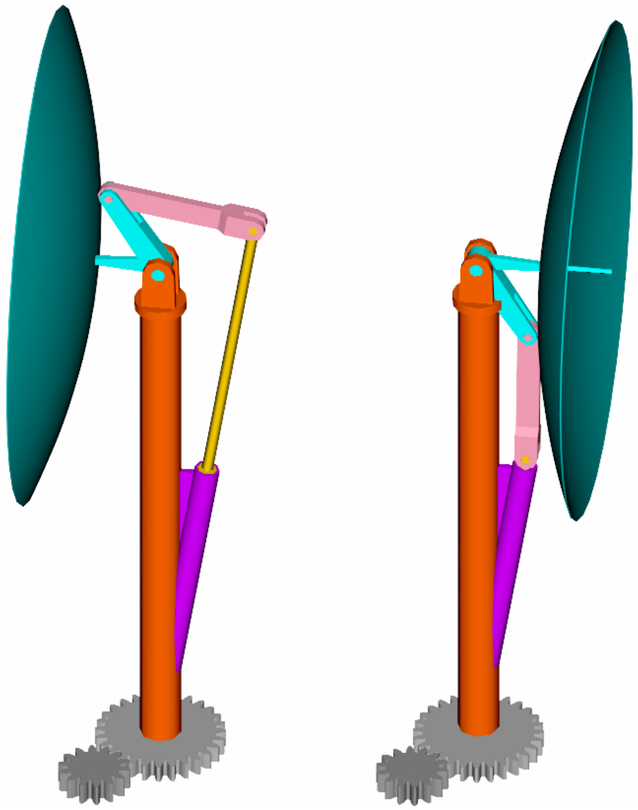


Fig. 4.3 Slider-rocker mechanism with 180° output member swing angle used as pitch drive in antenna or solar panel-orientation mechanisms for mobile application (airplane, ships, space crafts, satellites etc.) [5].

4.2.1 Relevant Numerical Examples

In Fig. 4.4-a and b are shown schematics of the optimum mechanisms which generate an imposed output angle $\Delta\varphi$=120°. As mentioned before, one mechanism is the mirror image of the other one, and therefore both have the same normalized member lengths AB=0.37466 and BC=0.50724, while the initial angles of the rocker are 69.37° and -9.37° respectively. For the actual case of mechanism (a), the variations of the output-member angle φ and transmission angle γ with the displacement of the input member were plotted in Fig. 4.4-c. The same graphs are valid for the complementary mechanism (b), provided that the slider is moved in reverse i.e. from x_{C1} to x_{C0}. For both cases the maximum deviation of the transmission angle γ from 90° will be only ±31.5°. Based on the results presented in the second part of the chapter,

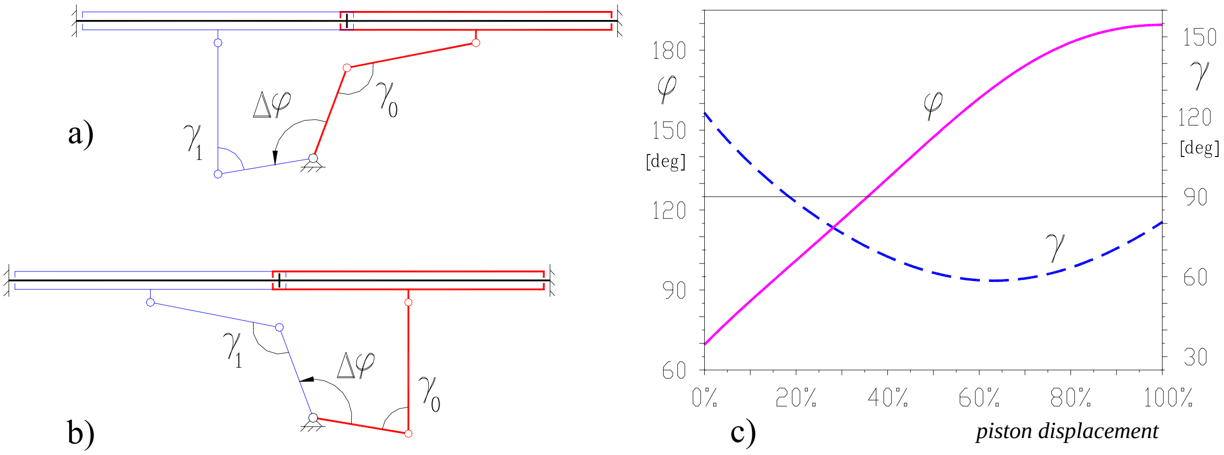


Fig. 4.4 Optimum slider-rocker mechanisms with Δφ=120°. The φ and γ diagrams (**c**) correspond to mechanism (**a**), while (**b**) is the complementary mechanism.

the optimum oscillating-slide mechanism that generates the same output stroke, exhibits variations of the transmission angle that are twice-as-large, rendering this type of mechanism unpractical for applications where the output member must rotate in excess of 120°.

Another slider-rocker mechanism configuration analyzed was that in which the imposed swing of the output member is Δφ=180°. As visible from Fig. 4.5-c, the maximum deviation of the transmission angle γ is still acceptable i.e. 90±57.32°. Fig. 4.2 shows that the level curve diagram generated for Δφ=180° exhibit in the lower part of the graph, two additional optimum points. These points correspond to the same base mechanism but mirrored with respect the x=0 line, which suffer from order defect (the rocker will rotate CW for the slider moving to the left). Such points occurred because of imposing the rocker to satisfy two prescribed positions that are exactly opposite.

It is to be noticed that the mechanisms in both Fig. 4.4 and Fig. 4.5 exhibit for about 80% of the input-member stroke, an almost linear input-output function, which can be sometimes desirable.

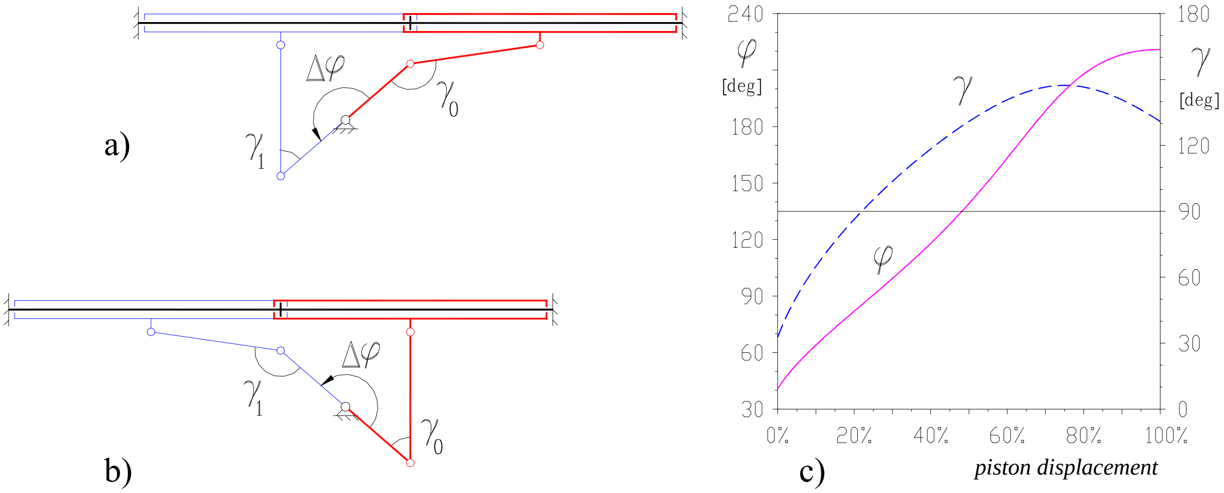


Fig. 4.5 Optimum slider-rocker mechanisms with Δφ=180°. The φ and γ diagrams (**c**) correspond to mechanism (**a**), while (**b**) is the complementary mechanism.

Another property both mechanisms described above have is that there are positions in which they would fully lock if motion were to be transmitted in reverse (from the rocker to the linear actuator). Such particular initial or final positions can be found in all optimum slider-rocker mechanisms with Δφ>77°, and can be utilize in some applications, for example as latch mechanisms or, in case of Δφ=180°, as stamping or printing press mechanisms, although sometimes this can be a drawback. In the past Bagci [1] studied the problem of synthesizing a slider-rocker mechanism for two prescribed positions of which one is a locking position, but without any concern about the transmission angle optimization.

4.2.2 Optimum Slider-Rocker Mechanism Design Chart

In order to help the design engineer in quickly sizing slider-rocker mechanism that generates an imposed stroke at the output member while simultaneously ensuring maximum motion transmission properties, the design charts in Fig. 4.6 have been generated. The values used for plotting the respective curves were obtained by repeatedly minimizing the objective function F1 for successive values of Δφ in the range [30°..200°], using Nelder and Mead's

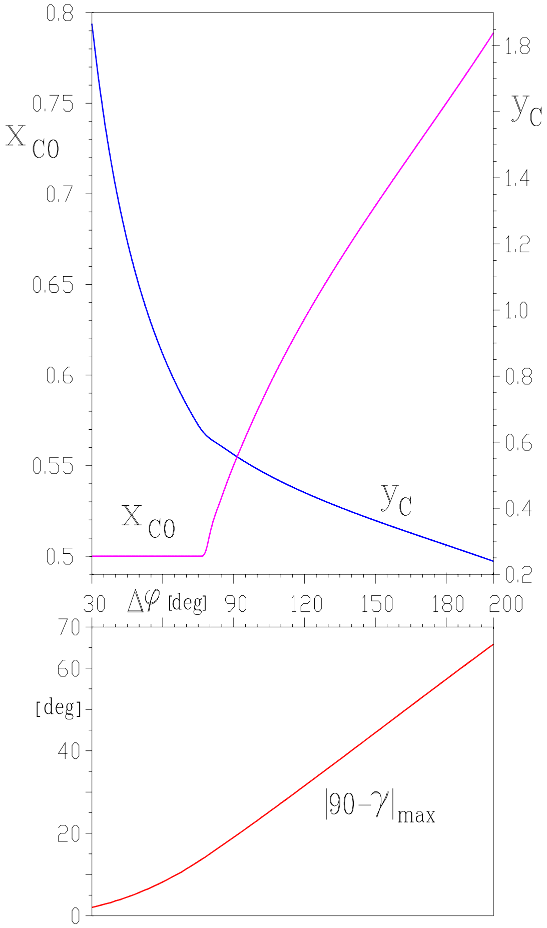


Fig. 4.6 Design chart for selecting the optimum values of x_{C0} and y_{C0} for a given output swing angle Δφ (above) and the corresponding maximum deviation of the transmission angle γ from 90° (below) of a slider-rocker mechanism. Additional parameters are determined using equations (4.9) (4.5) and (4.3).

algorithm [6] preceded by an elitist Population Based Incremental Algorithm. The constraints were handled using the *1K*-Penalty method as described in Chapter 2. For these optimum mechanisms, the dependence between the initial angle of the rocker and the imposed angle Δφ proved to be almost linear according to the formula:

$$\varphi_0 = -0.474925 \cdot \Delta\varphi + 126.415° \tag{4.9}$$

(where Δφ and φ_0 are both in degrees). With the values x_{C0} and y_C read from the design chart and the optimum angle φ_0 calculated with the above equation (4.9), the normalized lengths AB and BC of the rocker and of the coupler can be determined using equation (4.5) and either of equation (3). The maximum deviation of the transmission angle γ from 90° can be estimated using the graph provided in the lower part of Fig. 4.6.

Since for output angles $\Delta\varphi$ in excess of 77° the curves $x_{C0}(\Delta\varphi)$ and $y_{C0}(\Delta\varphi)$ have smooth appearances, the following approximate equations can be employed in substitute to the design chart in Fig. 4.6:

$$x_{C0}(\Delta\varphi) = 0.253944 + 0.003781 \cdot \Delta\varphi - 5.63703 \cdot 10^{-6} \cdot \Delta\varphi^2 \qquad (4.10)$$

$$y_{C}(\Delta\varphi) = 1.00473 - 0.0057709 \cdot \Delta\varphi + 9.88005 \cdot 10^{-6} \cdot \Delta\varphi^2 \qquad (4.11)$$

determined by solving appropriate second-degree curve fit problems.

If the complementary mechanism is wished to be used instead, in view of earlier findings, this can be obtained by mirror imaging about the x=0 vertical the mechanism obtained with the aid of the design chart in Fig. 4.6 or equations (4.10) and (4.11), with the difference that the initial angle of the rocker will be φ_0=180°-φ_0-$\Delta\varphi$ and the initial position of joint C of the input member will be x_{C0}=1-x_{C0}.

4.2.3 How to Handle Workspace Limitations

In some cases restrictions upon the mechanism link-lengths or ground joint dispositions are imposed, leading to a constrained optimization problem that can be formulated starting from the same objective functions F1. Alternatively, the designer can solve the problem interactively using appropriate contour-line plots similar to those in Fig. 4.2 (provided that the scale factors over x and y axes are equal) following the steps listed below:

1) Locate the ground pivot joint of the rocker at (0,0).

2) Draw at scale (i.e. normalized with respect to the piston stroke) over the same contour plots any surrounding object that might constrain the location and size of the moving links of the mechanism to be designed.

3) Position on the diagram y_C and x_{C0} i.e. the piston's axis and the center joint C for the piston fully retracted. The closest level curve available will help approximate the maximum deviation of the transmission angle γ from 90°.

4) If the transmission angle determined at #3 is satisfactory, find φ_0 that minimize F1 in equation (4.7) for the chosen values of x_{C0} and y_C – one easy way of doing this is to plot F1(φ_0) and extract from the graph the value of φ_0 at minima.

5) Calculate the normalized lengths AB and BC using equations (4.5) and (4.3) respectively and check for interference between the coupler and the rocker the surrounding objects plotted on the graph.

The final dimensions of the slider-rocker mechanism will be obtained by scaling the normalized lengths AB and BC determined above with a factor equal to the stroke of the linear actuator selected.

4.3. Synthesis of the Oscillating-Slide Mechanism

The oscillating-slide mechanism (Fig. 4.7) is the preferred means of converting the rectilinear motion of a linear actuator into swinging motion of a rocker member when this do not exceed 120°. This mechanism has the advantage of developing lower transverse forces upon the piston rod (negligible for slow moving machinery), and conversely between the piston and the cylinder.

Same as before, the maximum displacement of the output member $\Delta\varphi$ is measured between the initial position φ_0 and the final position φ_1 of the rocker. The requirement is to generate this imposed stroke $\Delta\varphi$ for an extension of the linear-actuator from a minimum length B_0C to a maximum length B_1C. As compared to the slider-rocker mechanism, where link-lengths were normalized with respect to the piston stroke, in this case the dimensions of the mechanism will

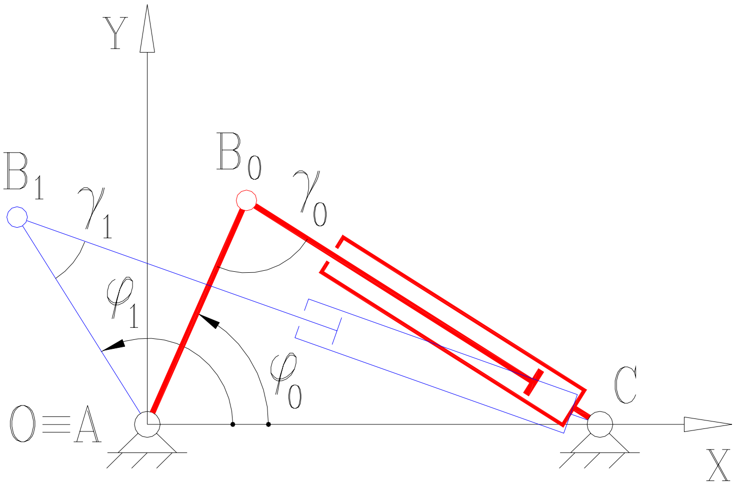


Fig. 4.7 Oscillating-slide mechanism shown in the extreme positions (B_0C, φ_0) and (B_1C, φ_1).

be normalized by assigning the ground member a unit length i.e. AC=1. For this normalized mechanism, the range of motion of the input member will be described using the k=B_1C/B_0C ratio, that will be further called *extension coefficient*.

In the deformable triangular loop ABC, the initial angle φ_0 of the output member will be considered as independent design variable, relative to which the remaining unknown parameters AB and B_0C can be calculated using the following equations:

$$\begin{aligned}(x_{B0}-x_C)^2+(y_{B0}-y_C)^2&=B_0C^2\\(x_{B1}-x_C)^2+(y_{B1}-y_C)^2&=B_1C^2\end{aligned}\tag{4.12}$$

where x_C=1, y_C=0 and B_1C=k×B_0C, while the coordinates of the joint center B are:

$$\begin{aligned}x_{B0}&=AB\cdot\cos\varphi_0 \qquad y_{B0}=AB\cdot\sin\varphi_0\,{}_\circ\\x_{B1}&=AB\cdot\cos\varphi_1 \qquad y_{B1}=AB\cdot\sin\varphi_1\end{aligned}\tag{4.13}$$

Equations (4.12) and (4.13) give:

$$\begin{aligned}AB^2-2\cdot AB\cdot\cos\varphi_0+1&=B_0C^2\\AB^2-2\cdot AB\cdot\cos\varphi_1+1&=B_1C^2=k^2\cdot B_0C^2\end{aligned}\tag{4.14}$$

and by further eliminating B_0C^2, a quadratic equation in AB is obtained:

$$AB^2-2\cdot AB\cdot(k\cdot\cos\varphi_0-\cos\varphi_1)/(k^2-1)+1=0\tag{4.15}$$

with the solutions:

$$\mathrm{AB} = \frac{k \cdot \cos\varphi_0 - \cos\varphi_1}{k^2 - 1} \pm \sqrt{\left(\frac{k^2 \cdot \cos\varphi_0 - \cos\varphi_1}{k^2 - 1}\right)^2 - 1} \tag{4.16}$$

For a chosen value of the initial angle φ_0 and of an *extension coefficient k*, relation (4.16) returns two values of the output member length AB, and therefore two different mechanism solutions exist. The solution obtained for the double sign in equation (4.16) being minus will be called *short-rocker oscillating-slide mechanism*, while that in which the double sign is plus will be called *long-rocker oscillating-slide mechanism*. Once the normalized length AB has been established, the corresponding normalized length of the fully retracted linear actuator B_0C can be determined using any of the equations (4.14).

Since the initial angle φ_0 of the output member can take any value within (0..180°), it is reasonable to search for those particular values of φ_0 for which the transmission angle γ has minimum deviation from 90° during the working range of the mechanism. For a current displacement φ of the output member between φ_0 and φ_1, the transmission angle γ can be evaluated using Cosine Law applied to the deformable triangle ABC, with AC=1 (Fig. 4.7):

$$\cos\gamma(\varphi) = \frac{\mathrm{AB}^2 + \mathrm{BC}(\varphi)^2 - 1}{2 \cdot \mathrm{AB} \cdot \mathrm{BC}(\varphi)} \tag{4.17}$$

For all practical mechanisms of oscillating-slide type, the displacement of the output-member varies monotonically with the slider displacement. Therefore the function $\cos\gamma(\varphi)$ given by relation (4.17) will also be monotonic and consequently the maximum deviation of the transmission angle from 90° will occur in the extreme positions of the rocker. This property has been applied in defining the following mini-max problem in one variable φ_0:

$$\begin{aligned}
&\text{minimize}\\
&F2(\varphi_0) = \max\{c_0, c_1\}\\
&\text{where} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (4.18)\\
&c_0 = |\cos\gamma(\varphi_0)|\\
&c_1 = |\cos\gamma(\varphi_1)|
\end{aligned}$$

By minimizing the above function F2 for an imposed value $\Delta\varphi$ of the output-member stroke, the dimensions of the mechanism with minimum deviation of the transmission angle from 90° can be determined. It is evident that the objective function F2 has to be penalized for the cases where the discriminant of equation (4.15) is negative and, similarly to objective function F1 in equation (4.7), for the cases where the vector loop ABC have different orientations in its initial and final position.

4.3.1 Numerical Example

Before advancing a number of conclusions and design recommendations, a numerical example will be considered, that of synthesizing an optimum oscillating-slide mechanism with a swinging angle of the output member $\Delta\varphi=35°$, to be driven with a linear actuator having an *extension coefficient k*=1.6. For this input data, the two minima of the objective function F2 obtained using Brent's algorithm [2], will correspond to the following two mechanism solutions:

The first solution, a *short-rocker mechanism* (Fig. 4.8-a) has the normalized lengths AB=0.6193, B_0C=0.6207 and initial angle of the rocker φ_0=36.23°. As can be seen from the kinematic diagram in Fig. 4.8-c, the maximum deviation of the transmission angle from 90° is ±17.5°.

The second solution Fig. 4.8-b, corresponds to a *long-rocker mechanism* and has the normalized dimensions: AB=1, B_0C=0.8437 and the initial angle of the rocker φ_0=49.9°.

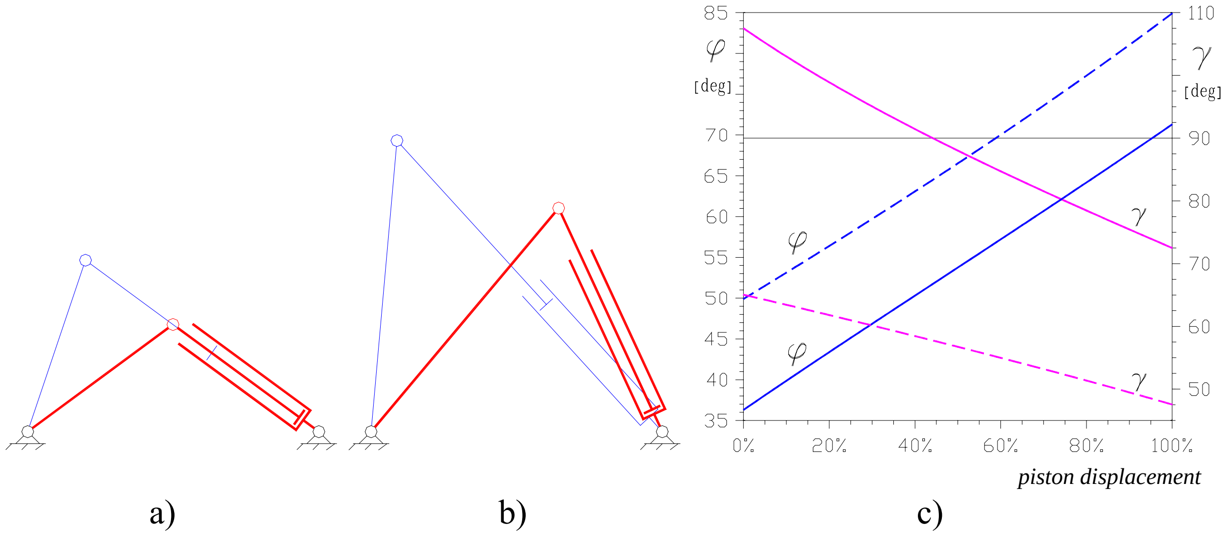


Fig. 4.8 Optimum oscillating-slide mechanisms with Δφ=35° and *k*=1.6 (**a** and **b**) together with the input-output and transmission angle diagram (**c**). The mechanism (**a**) is of the short-rocker type (the continuous lines in the diagram) while the mechanism (**b**) is the long-rocker type (the dashed lines in the diagram).

From the kinematic diagram Fig. 4.8-c (the dashed curves) it can be seen that the transmission angle γ has a larger variation than in case of the short-rocker mechanism i.e. 90±42.5°.

4.3.2 Oscillating-Slide Mechanism Design Recommendations

The study of a number of optimum solutions of the objective function F2 for various combinations of Δφ≤120° and *k*>1, revealed a number of general properties of the optimum *short-* and *long-rocker oscillating slide mechanism*:

1) The *short-rocker mechanism* ensures a better transmitting angle compared to its counterpart *long-rocker mechanism.*

2) The *long-rocker mechanism* is more appropriate for applications where the output member swings less than 90°.

3) For a given angle Δφ the transmission angle γ of the optimum *long-rocker mechanism* can be improved by selecting a linear actuator with a larger *extension coefficient k.*

4) The transmission angle of the optimum *short-rocker mechanism* is not influenced by *k*.

5) For all optimum *long-rocker mechanisms*, the rocker length will be equal to the base length i.e. AB=AC or, in case of the normalized mechanisms, AB=AC=1.

6) For a given swinging amplitude Δφ of the output member, the transmission angle maximum range is equal to the range of the rocker angle i.e. γ_{max}-γ_{min}=Δφ for both the short and long-rocker optimum mechanism.

7) For the *short-rocker mechanism*, the deviation of the transmission angle γ is symmetric with respect to 90°, a consequence of the fact that in the extreme positions the joints B_0, B_1 and C are collinear.

4.3.3 Short-Rocker Oscillating-Slide Mechanism Design Procedure

This last of the above properties suggests the following *graphical method* of synthesizing an oscillating-slide mechanism of the short-rocker type:

1) Draw three collinear points C, B_0 and B_1 (in this order) such that B_0C is the minimum length of the linear actuator to be used and B_1C is the maximum length of the actuator.

2) Locate point A on the perpendicular line to the middle of the segment B_0B_1 such that the angle B_0AB_1 is the desired swing angle Δφ of the rocker.

3) The maximum deviation of the transmission angle from 90° of the mechanism will be:

$$\gamma_{\min} = 90° - \Delta\varphi/2 \quad \text{and} \quad \gamma_{\max} = 90° + \Delta\varphi/2 \tag{4.19}$$

Equation (4.19) gives indication upon the applicability limits of the oscillation-slide mechanisms mentioned earlier i.e. for Δφ angles of the rocker in excess of 120°, the transmission angle γ varies more than ±60° from 90°, making the mechanism prone to locking during operation.

4.3.4 Long-Rocker Oscillating-Slide Mechanism Design Procedure

Although optimum oscillating-slide mechanisms with long-rocker have lower motion transmission performances than their short-rocker counterparts, the latter ones have their own potential for applications in association with large *extension coefficient* actuators, particularly when compact arrangements are sought. One such example is that of dump truck mechanisms that employ telescopic actuators with extension coefficients *k* greater than 3.

In the following a simple synthesis method of this type of mechanisms will be described, and a transmission angle performance chart will be provided, which gives an overview upon the motion transmission properties of mechanisms with various Δφ and *k* parameters.

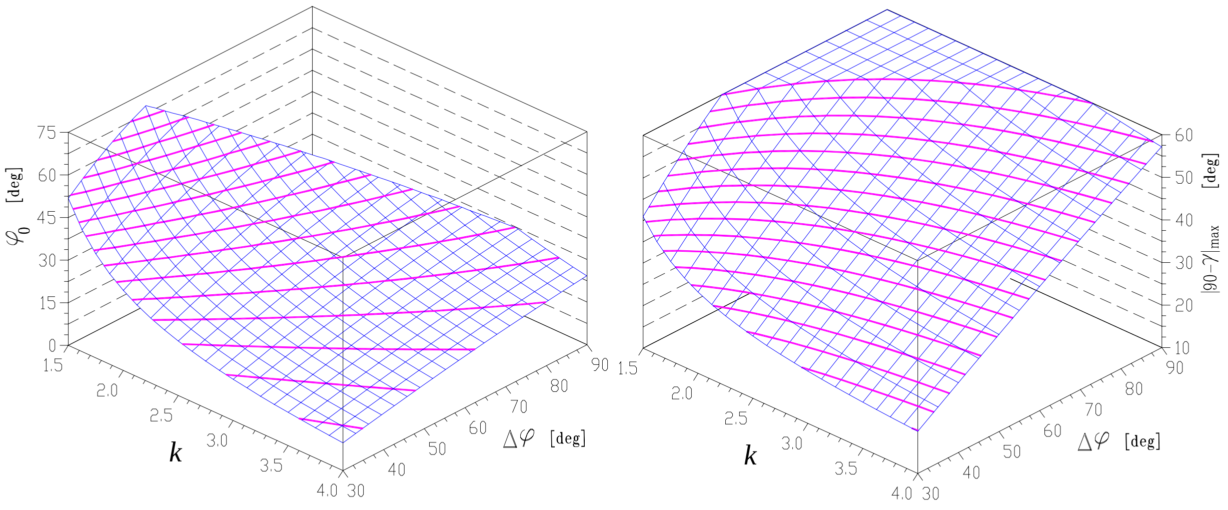
The proposed design procedure uses the fact that the loop of the mechanism is an isosceles triangle with AC=AB. Therefore, by applying Cosine Law for the initial and final configuration of the mechanism (for $\varphi=\varphi_0$ and $\varphi=\varphi_1$) the following equation is obtained:

$$k^2 = \left(\frac{\mathrm{B_1C}}{\mathrm{B_0C}}\right)^2 = \frac{1-\cos(\varphi_0+\Delta\varphi)}{1-\cos\varphi_0} \tag{4.20}$$

For a given extension coefficient *k* and maximum swinging angle Δφ this equation can be solved in the unknown φ_0, using an iterative method, and the synthesis problem solved. The minimum and maximum values of the transmission angle γ can then be calculated with the following relations (easy derived based on Fig. 8-b):

$$\begin{aligned} \gamma_{\min} &= 90° - \varphi_0/2 \\ \gamma_{\max} &= 90° - (\Delta\varphi+\varphi_0)/2 \end{aligned} \tag{4.21}$$

Alternative to numerically solving equation (4.20), one can select the initial angle of the rocker φ_0 using the design chart in Fig. 4.9-a (generated by repeatedly minimizing objective function F2).



a) b)

Fig. 4.9 3D-design chart (**a**) for choosing the optimum value of the initial angle φ_0 of a long-rocker oscillating-slide mechanism, and performance chart (**b**) showing the expected maximum deviation from 90° of the transmission angle γ.

In searching for an optimum mechanism configuration and of the corresponding initial angle φ_0 of the rocker, the extension coefficient *k* can also be considered as design variable of a discrete type; for example, when searching for the best long-rocker mechanism solution, the designer can experiment with more than one linear actuator available from suppliers. Therefore, in order to have an overview upon the motion capabilities of the oscillating-slide mechanism of the long-rocker type, the performance diagram in Fig. 4.9-b has been generated. This diagram shows that the oscillating-slide mechanism with long rocker is suitable for generating maximum amplitudes of the output member less than 90°, and should be used in association with large extension-coefficient actuators.

4.4. Conclusions

Two mechanisms widely used for convert the input motion of a linear actuator into the rotary motion of an output-member rocker were investigated. The slider-rocker mechanism proved to have very good capabilities of generating swinging amplitudes of the output member of 180° and over, while still ensuring an acceptable transmission angles. The oscillating-slide mechanism (which is preferred due to the reduced transversal forces upon the linear motor) comes in two optimum configurations, one having a relatively short-rocker and the other one a long rocker. The optimum short-rocker mechanism can generate swing amplitudes of the output member up to 120° with acceptable transmission angle variations. The long rocker mechanism should be used in association with large stretch-coefficient actuators and are recommended when compact arrangements are needed and when the output-member stroke is less than 90°.

For all these mechanisms design charts or easy to apply synthesis procedures were proposed, which permit quick selection of the optimum geometry. Also given are performance charts that allow a convenient overview upon the motion transmission capabilities of these mechanisms useful for design problems when the range of motion of the input and output member do not have strictly imposed values.

4.5. References

[1] Bagci, C. (1987) “Synthesis of Linkages to Generate Specified Histories of Forces and Torques-The Planar Slider-rocker Mechanism,” Proc. of the 13th ASME DETC Vol. 10-2, p. 237-244.

[2] Brent, R.P., 1973, *Algorithms for Minimization Without Derivatives*, Prentice Hall.

[3] Erdman, A.G., Sandor, G. N. and Kota, S. (2001) *Mechanism Design: Analysis and Synthesis,* Prentice Hall.

[4] Nelder, J. A. and Mead, R. A, (1965) “Simplex Method for Function Minimization,” Computer Journal, Vol. 7, 308-313.

[5] Simionescu, P.A. and D.G. Beale (2004) "Orientation Mechanism," US Provisional Patent Application, filed with the USPTO on 04/12/2004.

[6] Tao, D.C. (1964) *Applied Linkage Synthesis* Addison-Wesley.

CHAPTER 5. OPTIMUM TEETH-NUMBER SYNTHESIS OF A MULTISPEED PLANETARY TRANSMISSION

The gear-teeth number synthesis of an automatic planetary transmission used in automobiles is formulated as a constrained optimization problem that is solved with the aid of an elitist Population Based Incremental Learning algorithm (PBIL). The design parameters are the teeth number of each gear, the number of multiple planets and gear module, while the objective function is defined based on the departure between the imposed and the actual gear ratios, constrained by teeth-undercut avoidance, limiting the maximum overall diameter of the transmission and ensuring proper planet spacing. For the actual case of a Ravigneaux planetary transmission with 3+1 speeds, the design space is visualized to show the effect of various constraints, and some optimum results presented.

5.1 Introduction

The wide applicability of planetary gears in the aircraft, marine and mainly automotive industry (particularly as automatic multispeed transmissions), has brought a great deal of attention to this topic. The literature on the design of planetary automatic transmissions covers conceptual design [3] [4] [13] [14] [19] [20] [27] [33] [41], kinematic analysis [9] [12] [20] [21] [26] [33] [40], power flow and efficiency analysis [22] [23] [30] [31]. Less work however has been done on the design of multispeed planetary transmissions from the condition of satisfying imposed gear ratios - the available literature covers mostly fixed axles transmissions [5] [6] [7] [11] [35] and design of single-ratio planetary units [2] [6] [16] [29].

Specific to teeth number synthesis of multispeed planetary transmissions are the design variables which must be integers (gear teeth and the number of multiple planets) and the numerous constraints. These constraints reduce significantly the feasible domain of the design space, making the synthesis problem quite difficult to solve. The work published on teeth number synthesis of multispeed planetary transmissions are, for the most part, hand-calculation oriented [8] [28] [36], or in the case of computer implemented approaches, only some of the numerous constraints were actually considered [1] [24] [25].

The constraints imposed on multispeed planetary transmissions derive from:

a) the minimum allowed number of teeth each gear can have so that undercut do not occur,

b) the maximum allowed diameter of the whole assembly,

c) the condition of central gears having coaxial axes,

d) the requirement of equally spacing multiple planets and

e) the noninterference condition of neighboring gears.

A maximum mechanical efficiency requirement can also be imposed, although for most multispeed planetary transmissions which closely fulfill the prescribed gear ratios, the mechanical efficiency can be calculated beforehand. Aspects like gear material and bearing selection from the condition of volume and cost minimization and of satisfying a required design life can also be prescribed early in the design process. However, since these can be decoupled from the gear-teeth selection, it is preferable to be solved as a subsequent multiobjective optimization problem once a satisfactory teeth-number combination becomes available [34].

For this particular design problem the gear-teeth numbers and the number of equally spaced planets are imposed integer values, while the module of the gears can have only discrete values in accordance to gear standards. As discussed earlier, such design variables

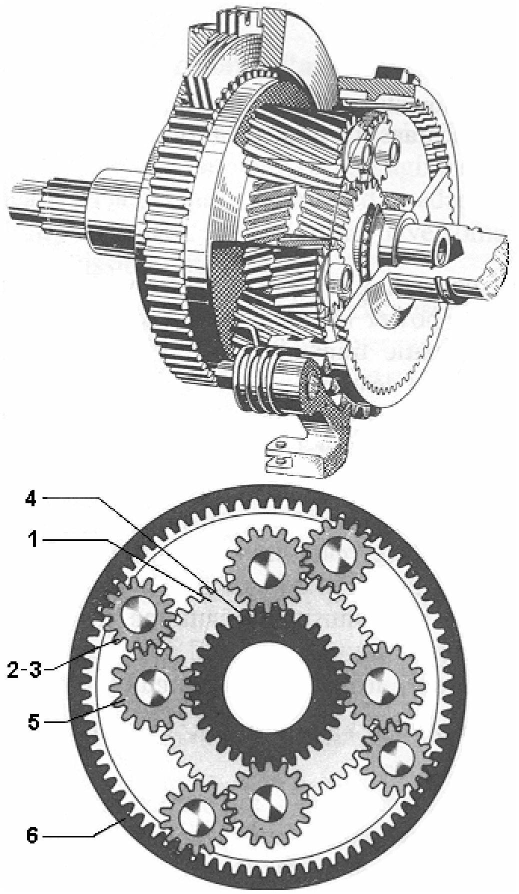


Fig. 5.1 Ravigneaux planetary gear [17]: 1 small sun gear; 2-3 broad planet gear; 4 large sun gear; 5 narrow planet gear; 6 ring gear.

may be considered from the beginning integers, or a continuous-variable nonlinear programming problem can be solved first, and afterwards a supplementary search for integer optimum values performed in the vicinity of the continuous optimum found. The former approach is proven to have an increased capacity of locating a true integer/discrete optimum [35] and was therefore adopted for the present example of optimum synthesis a 3+1 speeds Ravigneaux planetary transmission.

5.2 The Ravigneaux 3+1 Gear Transmission

Fig. 5.1 shows a planetary transmission of the Ravigneaux type with 3 forward and 1 reverse gears used in automobiles. A kinematic diagram of the transmission is available in Fig. 5.2, where the broad planet gear is shown as two compound gears 2 and 3. Based on the

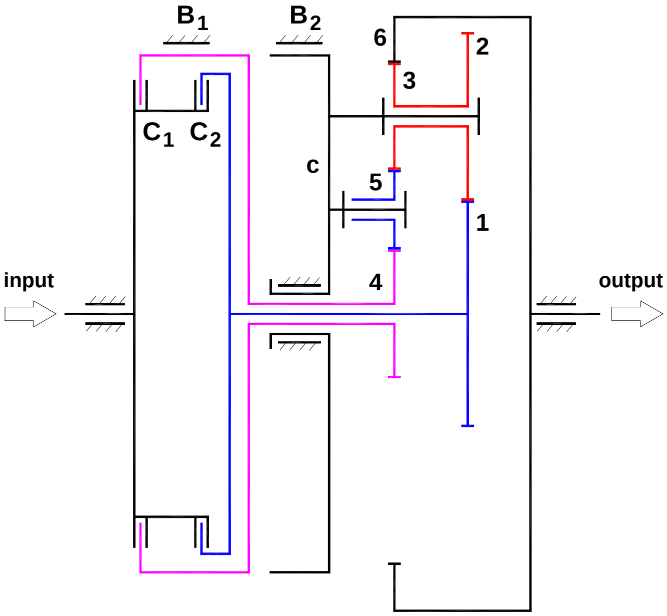


Fig. 5.2 Kinematic diagram of a 3+1 gear ratios Ravigneaux planetary transmission. Note that the broad planet gear consists now of two distinct gears 2 and 3.

clutch/brake activation required in each gear (Table 5.1), it can be shown that for the first and reverse gears the planet carrier is immobile and the equivalent transmission is a fixed-axle one with the following gear ratios:

$$i_1 = N_6/N_4 \tag{5.1}$$

and

$$i_R = -(N_2N_6)/(N_1N_4). \tag{5.2}$$

In the third gear, the planet carrier, sun gears and ring gear rotate together as a whole:

$$i_3 = 1 \tag{5.3}$$

i.e. a direct drive, which ensures an increased mechanical efficiency of the transmission.

The second gear configuration is the only case when the transmission works as a planetary gear set. Considering the planet carrier **c** immobile, three *basic gear ratios* can be defined as follows:

Table 5.1 Clutch/brake activation table of the Ravigneaux planetary transmission.

Speed	**Clutch/Brake**			
	C_1	C_2	B_1	B_2
First	x			x
Second		x	x	
Third	x	x		
Reverse		x		x

$$i_{16}^{c} = -\frac{N_2 N_6}{N_1 N_3},\; i_{46}^{c} = \frac{N_6}{N_4},\; i_{14}^{c} = -\frac{N_2 N_4}{N_1 N_3}. \tag{5.4}$$

Through motion inversion, which converts the planetary gear into a fix axle transmission, the following additional relations between the angular velocities (ω) of the sun gears 1 and 4, ring gear 6 and planet carrier **c** can be written as:

$$i_{16}^{c} = \frac{\omega_2 - \omega_c}{\omega_6 - \omega_c},\; i_{46}^{c} = \frac{\omega_4 - \omega_c}{\omega_6 - \omega_c},\; i_{14}^{c} = \frac{\omega_1 - \omega_c}{\omega_4 - \omega_c}. \tag{5.5}$$

Eliminating ω_c between any two of the above equations and for ω_4=0, the sought-for second gear ratio can be obtained:

$$i_2 = \frac{N_6\left(N_1 N_3 + N_2 N_4\right)}{N_1 N_3\left(N_6 - N_4\right)}. \tag{5.6}$$

The expressions of the gear ratios i_1, i_2 and i_R previously derived will be further used in formulating the teeth number optimum synthesis problem.

5.3 The Optimization Problem

For the present optimum design problem, the design variables are the teeth numbers of the sun, ring and planet gears and the number of equally-spaced, identical planets to be mounted on the planet carrier. Other design variables are modules m_1=m_2 and m_3=m_4=m_5=m_6 which can have only discrete values in accordance with gear standards.

Appropriate objective function to be used for minimizing the departure between the actual (i_k) and the imposed (i_{0k}) gear ratios could be a weighted maximum norm:

$$f_1(N_1 ... N_n, p, m_j) = \max_k \left(w_k \cdot \left| i_k - i_{0k} \right| \right) \tag{5.7}$$

or sum of weighted squared residuals:

$$f_2(N_1 ... N_n, p, m_j) = \sum_k w_k (i_k - i_{0k})^2 \,. \tag{5.8}$$

where k={1, 2, R} is the number of gear ratios (less the direct drive) and w_k are weighting coefficients which will differentiate the importance of these gear ratios during the design process (for example lowering the importance of the reverse gear which is engaged for shorter periods of time).

As it will be seen, some of the constraints (as well as the objective function itself) have meaning for both continuous and integer values of the design variables (i.e. the gears inside the transmission can be considered ideal friction wheels), while other constraints require explicitly the design variables to be integers. Therefore treating some of the variables as continuous and some as discrete or integers may bring additional complications to the problem formulation.

Noting with ns=2 the number of sun gears, np=3 the number of distinct planets (gears 2, 3 and 5) and nr=1 the number of ring gears, the lower side constraints upon teeth numbers have the following general expressions:

$$\mathrm{N\,min}_j \le \mathrm{N}_j \quad (1 \le j \le ns + np) \tag{5.9}$$

where Nmin$_j$ (the minimum number of teeth the sun or planet gears can have) are specified from the condition of undercut avoidance as 17 or 14 even 12 teeth if the use of nonstandard gears is acceptable,.

Limiting the maximum outer diameter of the transmission D$_{\text{limit}}$ can be accounted for by imposing an upper value to the root diameter of the ring gear as follows:

$$m_3(\mathrm{N}_6 + 2.5) \le \mathrm{D}_{\mathrm{limit}}\,. \tag{5.10}$$

Imposing a maximum outside diameter of the transmission also requires that the working space of planet 2 to be constrained:

$$2\left[m_1\left(\mathrm{N}_1 + \mathrm{N}_2\right)/2 + m_1\left(\mathrm{N}_2/2 + 1\right)\right] \le \mathrm{D}_{\mathrm{limit}}\,. \tag{5.11}$$

i.e. the standard center distance of gears 1 and 2 (the first term) plus the outside radius of planet 2 (the second term) to be less then maximum admissible radius of the transmission.

The condition of solar and ring gears to have coaxial axles can be written either as equality constraints, or, if nonstandard gears are acceptable, as inequality constraints. The latter case is more advantageous to the searching process and, for the actual case of gears 1 and 6 the following relation must be imposed between the standard center distances of gears 1 and 2 and 6 and 3. This can be generically written as:

$$\left|m_1\left(\mathrm{N}_1 + \mathrm{N}_2\right)/2 - m_3\left(\mathrm{N}_6 - \mathrm{N}_3\right)/2\right| \le \left(m_1 + m_3\right)/2 \tag{5.12}$$

The neighborhood condition refers to adjacent, non-meshing gears, the teeth of which are required to operate at a distance greater than a certain minimum value $d\min_{ij}$:

$$d_{ij} \ge d\min_{ij} \tag{5.13}$$

where d_{ij} is the distance between the addendum circles of the teeth of the respective neighbor wheels. For broad planet 2-3 and idler planet 5 (Fig. 5.3) these distances can be approximated with:

$$d_{22} = 2\left[m_1 \frac{\mathrm{N}_1 + \mathrm{N}_2}{2} \sin\left(\frac{\pi}{p}\right) - m_1\left(\frac{\mathrm{N}_2}{2} + 1\right)\right] \tag{5.14}$$

$$d_{33} = 2\left[m_3 \frac{\mathrm{N}_6 - \mathrm{N}_3}{2} \sin\left(\frac{\pi}{p}\right) - m_3\left(\frac{\mathrm{N}_3}{2} + 1\right)\right] \tag{5.15}$$

and

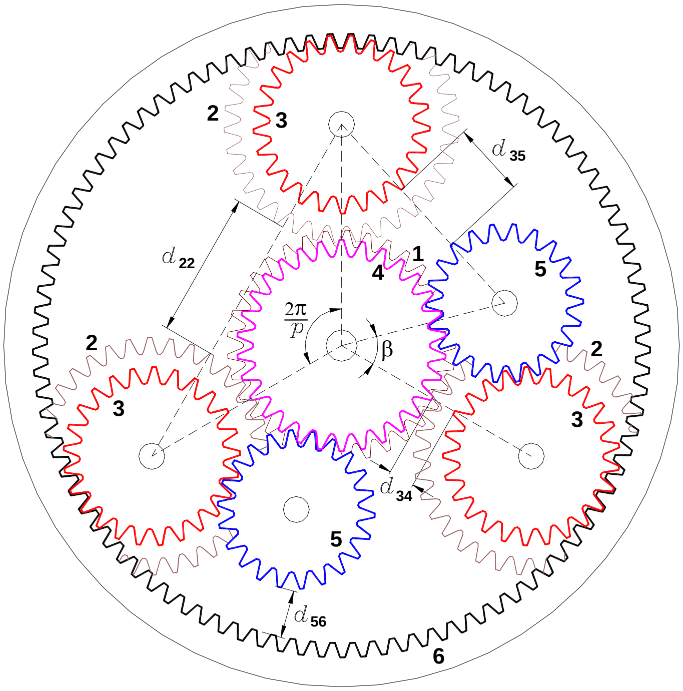


Fig. 5.3 Schematic for calculating distances d_{22}, d_{34}, d_{35} and d_{35}. Notice that one of idler planets 5 has been removed for clarity.

$$d_{55} = 2\left[m_3 \frac{N_4 + N_5}{2} \sin\left(\frac{\pi}{p}\right) - m_3\left(\frac{N_5}{2} + 1\right)\right]. \tag{5.16}$$

For checking the interference avoidance of planets 3 and 5 the following distance must also be evaluated:

$$d_{35} = \sqrt{C_{36}^2 + C_{45}^2 - 2C_{36}\,C_{45}\cos\left(\frac{2\pi}{p} - \beta\right)} - m_3\left(\frac{N_3}{2} + 1\right) - m_3\left(\frac{N_5}{2} + 1\right) \tag{5.17}$$

with

$$\beta = \cos^{-1}\left(\frac{C_{36}^2 + C_{45}^2 - C_{35}^2}{2C_{36} \cdot C_{45}}\right), \; C_{36} = m_3 \frac{N_6 - N_3}{2}$$
$$C_{45} = m_3 \frac{N_4 + N_5}{2} \text{ and } C_{35} = m_3 \frac{N_3 + N_5}{2} \tag{5.18}$$

The distances between the addendum circles of planet gear 3 and sun gear 4:

$$d_{34} = m_3 \frac{N_6 - N_3}{2} - m_3\left(\frac{N_3}{2} + 1\right) - m_3\left(\frac{N_4}{2} + 1\right) \tag{5.19}$$

and between addendum circles of ring gear 6 and idler planet gear 5:

$$d_{56} = m_3\left(\frac{N_6}{2} - 1\right) - m_3\frac{N_4 + N_5}{2} - m_3\left(\frac{N_5}{2} + 1\right). \tag{5.20}$$

must also be kept larger than a certain value in order to allow satisfactory lubricant flow.

When the minimum admissible values $d\text{min}_{ij}$ in equation (5.13) are defined as a multiple of the modulus of the respective neighboring gears, the corresponding inequalities simplify to some extent (see Appendix 2 where the optimization problem has been summarize for conciseness and where $d\,\text{min}_{ij}$ were considered equal to multiples d_{ij} of either m_1 or m_3).

The most restrictive constraints of all are the conditions of assembling equally spaced planets. These are equality constraints in integer numbers. Based on the theory developed in [37], for the planetary gear in Fig. 5.2 the condition of having equally spaced identical compound planets 2-3 writes:

$$Frac\left(\frac{1}{p}\left|\frac{1}{i^{c}_{1-2}} - \frac{1}{i^{c}_{6-3}}\right|\right) = \left|\frac{A}{N_2} \pm \frac{B}{N_3}\right|. \tag{5.21}$$

where A and B are integers less-equal than N_2/p and N_3/p respectively, *Frac*(..) is the fractional part of the expression in parentheses while

$$i^{c}_{1-2} = -N_2/N_1 \text{ and } i^{c}_{6-3} = N_3/N_6 \tag{5.22}$$

are called *partial basic ratios* of the planetary gear (i.e. the gear ratios between the designated gears when the planet carrier is maintained fix).

For the planetary transmissions in Fig. 5.1 and 5.2 a second assembly condition requirement must be imposed to idler planets 5. The equivalent expression in basic ratios applied to either gear 4, 5, 3 or 6 (and without resorting to the *Frac* operator so that simplifications can be made across the equal sign), reduces to [37]:

$$(N_6 - N_4)/p = integer. \tag{5.23}$$

5.4 Search Algorithm

The optimization problem previously formulated has been solved with the aid of an elitist Estimation of Distribution Algorithm of the Population Based Incremental Learning (PBIL) with 1*K-Penalty* as described in Chapter 2. The general structure of the algorithm presented in connection with equations (2.2, 2.3 and 2.4) was modified in order to avoid premature convergence due to standard deviation becoming too small as follows:

If the infinite norm of the standard deviation vector $\{\sigma_i\}$ is less than a certain small value σ_{min} while G_c (the number of the current generation) is less than $r \cdot G_{\text{max}}$, with $r < 1$, then the algorithm is restarted but without discarding the best solution found so far.

The numerical results reported below were obtain in 200 runs of the elitist PBIL algorithm with M=80, N=40, r=0.75 and w=1. The stopping criteria considering was exceeding G_{max}=500 generations.

5.5 Numerical Results

The design problem detailed above and systematized in Appendix 2 was solved for a maximum outer diameter of the transmission D_{limit}=220 mm and for the following imposed gear ratios: i_{01}=3.11, i_{02}=1.84, i_{03}=1.0, i_R=-3.22.

For simplicity a maximum norm-based objective function f_1 (Eq. 5.7) with all weightings w_k equal to 1.0 has been considered.

In order to facilitate the searching process, the teeth number combinations for which only assembly condition (5.23) holds were not rejected, but rather assigned the objective function 40-50 times its current value. This is because according to [37] there are alternative solutions available for the cases when equally spaced multiple planets cannot be assembled together as follows:

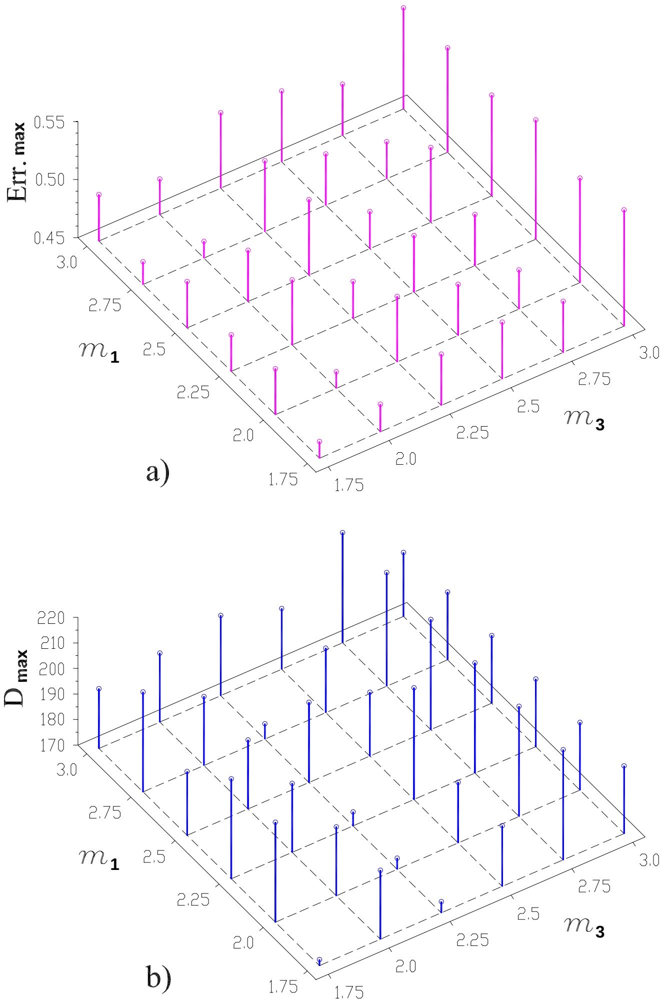


Fig. 5.4 Projection of the lower envelope of objective function f_1=Err. max with $N_2 \neq N_3$ on the (m_1,m_3,f_1) space (**a**), and plot of the corresponding outer diameter of the transmission (**b**) for the case of equally spaced identical planets.

a) One possibility is to assemble identical planets at different spacing angles (this approach is more conveniently applicable to planetary units restricted by only one assembly-condition equation, which is not the case of the current problem).

b) In case of the planetary transmission in Fig. 5.2, equally spaced *p* nonidentical compound planets can be assembled, provided that gears 2 and 3 are manufactured rotated relative to each other by a certain angle $\delta\varphi_k$ as follows [37]: one compound planet must be considered as reference while the *k*-th planet (counting in the clockwise direction) requires for assembly to have gear 2 and 3 rotated counterclockwise by the following two angles about the planet carrier:

$$\varphi_{2-k} = \frac{2\pi}{3} \cdot \left(-\frac{N_1}{N_2} \right) \text{ and } \varphi_{3-k} = \frac{2\pi}{3} \cdot \left(\frac{N_6}{N_3} \right) \tag{5.24}$$

which translate into the following relative angle at which assembly is ensured:

$$\delta\varphi_k = \varphi_{2-k} - \varphi_{3-k} = 2\pi \cdot Frac\left[\frac{k}{3} \cdot \left(-\frac{N_1}{N_2} - \frac{N_1}{N_2} \right) \right] \tag{5.25}$$

Since gears 2 and 3 have periodic profiles, angles $\delta\varphi_k$ are actually equivalent to a much

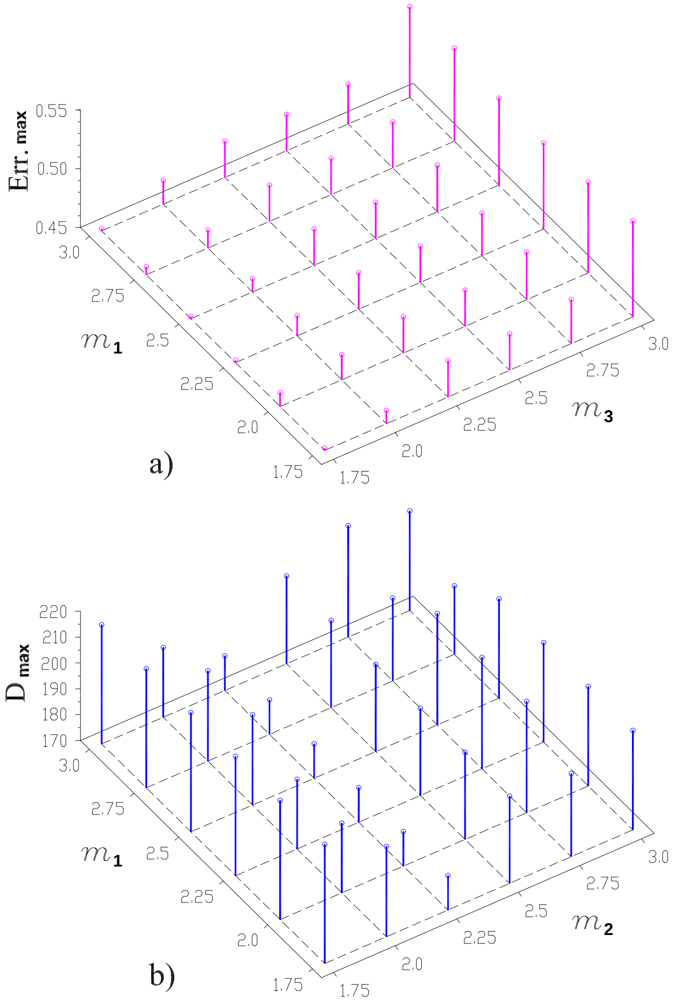


Fig. 5.5 Projection of the lower envelope of objective function f_1=Err. max with $N_2 \neq N_3$ on the (m_1,m_3,f_1) space (**a**), and plot of the corresponding outer diameter of the transmission (**b**) for the case of equally spaced nonidentical compound planets 2-3.

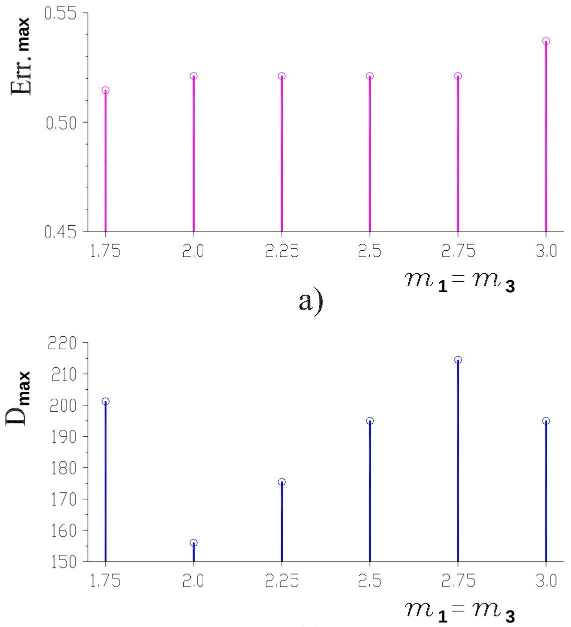
smaller angles, which may allow identically manufactured planets to be plastically torsioned a small amount so that assembling becomes possible.

In order to assist the design process, a visualization of the objective function f_1 has been performed by projecting its hypersurface down to the 3D space of (m_1, m_3, f_1). As shown in Chapter 3, the lower envelope of the hypersurface of a single valued function of more than two variables $f(x_{1,}\, x_2 \ldots x_n)$ projects down to the 3D space formed with the function value f and two of the variables, say x_1 and x_2, as the partial global minima function:

$$f_{\downarrow 3..n}(x_1, x_2) = global \min_{x_3 \ldots x_n} f(x_1 \ldots x_n)\,. \tag{5.26}$$

where x_1 and x_2 are *scan variables* and $x_{3..n}$ are *search variables*. In a similar manner, by considering only one scan variable, the lower envelope of the same hypersurface can be plotted as a 2D graph:

$$f_{\downarrow 2..n}(x_1) = global \min_{x_2 \ldots x_n} f(x_1 \ldots x_n)\,. \tag{5.27}$$



a)

b)

Fig. 5.6 Projection of the lower envelope of objective function f_1=Err. max with m_1=m_3 and N_2=N_3 on the (m_1,f_1) plane (**a**), and plot of the corresponding outer diameter of the transmission (**b**) for the case of equally spaced identical planets.

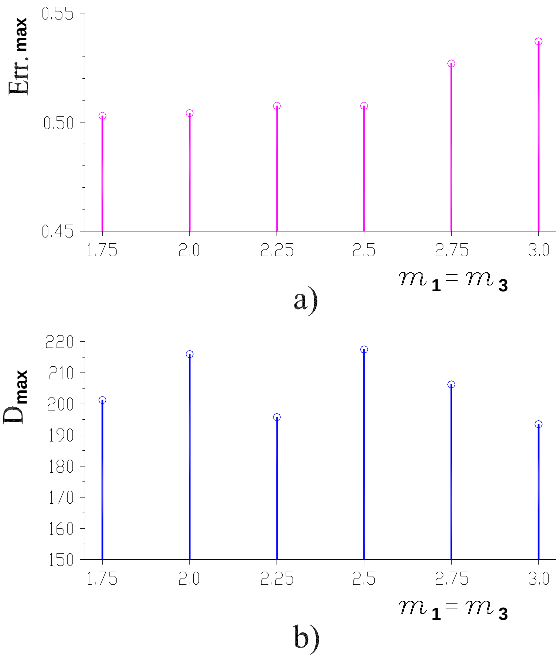


Fig. 5.7 Projection of the lower envelope of objective function f_1=Err. max with m_1=m_3 and N_2=N_3 on the (m_1,f_1) plane (**a**), and plot of the corresponding outer diameter of the transmission (**b**) for the case of equally spaced nonidentical planets 2-3.

In case of objective function f_1, choosing as scan variables the module m_1 and m_3 of the gears, the graph in Fig. 5.4-a has been generated for a transmission with equally spaced planets and identical compound gears 2 and 3. Similar plots have been generated (Fig. 5.5-a) for the cases when the compound planets must be manufactured with gears 2 and 3 rotated at different angles.

The plots in Fig. 5.4-a and Fig. 5.5-a allow the designer to select a suboptimum teeth number combination based on additional criteria, like the availability or cost of gear cutting tools, or the requirement of all gears having the same modulus i.e. m_1=m_3.

The accompanying graphs (Fig. 5.4-b and Fig. 5.5-b) showing the actual maximum diameter of the transmission D_{max} calculated with the left-hand side of equations (5.10) and (5.11), provide additional information to the designer, who can select the teeth number combination and module m_1, m_2 which ensure an outer diameter of the transmission smaller than the actual imposed value D_{limit}. They also indicate whether or not an increased of the maximum allowed diameter of the transmission can insure further reduction of the departure between the imposed and the actual gear ratios.

Referring back to Fig. 5.1, it is evident that it is more advantageous to manufacture planetary transmission with planets 2-3 having identical gears 2 and 3. The corresponding optimum solution can be obtained by minimizing objective function f_1 subject to the same constraints and additionally imposing $N_2=N_3$ and $m_1=m_3$. Fig. 5.6-a and Fig. 5.7-a show 2D projections (m_1 is the only scan variable) of the lower envelope of the objective-function's hypersurface when subject to these additional constraints. Fig. 5.6-a was generated for the case of equally spaced identical planets while Fig. 5.7-a corresponds to the gears 2 and 3 of the compound planets being manufactured at different relative angle according to equations (5.24) and (5.25).

The global optimum solution corresponding to the transmission variants studied through the graphs in Figs. 5.4 through 5.7 are gathered in Table 5.2. The planetary transmission that

Table 5.2 Results obtained through the optimization process (note the double global optimum obtained in case of the transmission with $N_2 \neq N_3$ and with identical compound planets – solutions 2 and 3).

	Solution 1	**Solutions 2, 3**	**Solution 4**	**Solution 5**	**Solution 6**
Maximum Error	0.45186 (24.6%)	0.46435 (25.2%)	0.46435 (25.2%)	0.50290 (26.9%)	0.51462 (27.4%)
N_1	32	36	27	45	45
N_2	25	28	21	34	34
N_3	41	32	32	$=N_2$	$=N_2$
N_4	34	27	27	31	31
N_5	29	24	24	36	18
N_6	121	96	96	112	111
m_1	2.50	1.75/2.0	2.75	1.75	1.75
m_3	1.75	1.75/2.0	2.00	1.75	1.75
p	3	3	3	3	4
i_1	3.5588	3.5556	3.5556	3.6129	3.5806
i_2	2.2919	2.3044	2.3044	2.3353	2.3433
i_R	-2.7803	-2.7654	-2.7654	-2.7298	-2.7054
Dmax mm	216.1	172.4/197	197	201.3	201.3
Identical planets	No	Yes	Yes	No	Yes

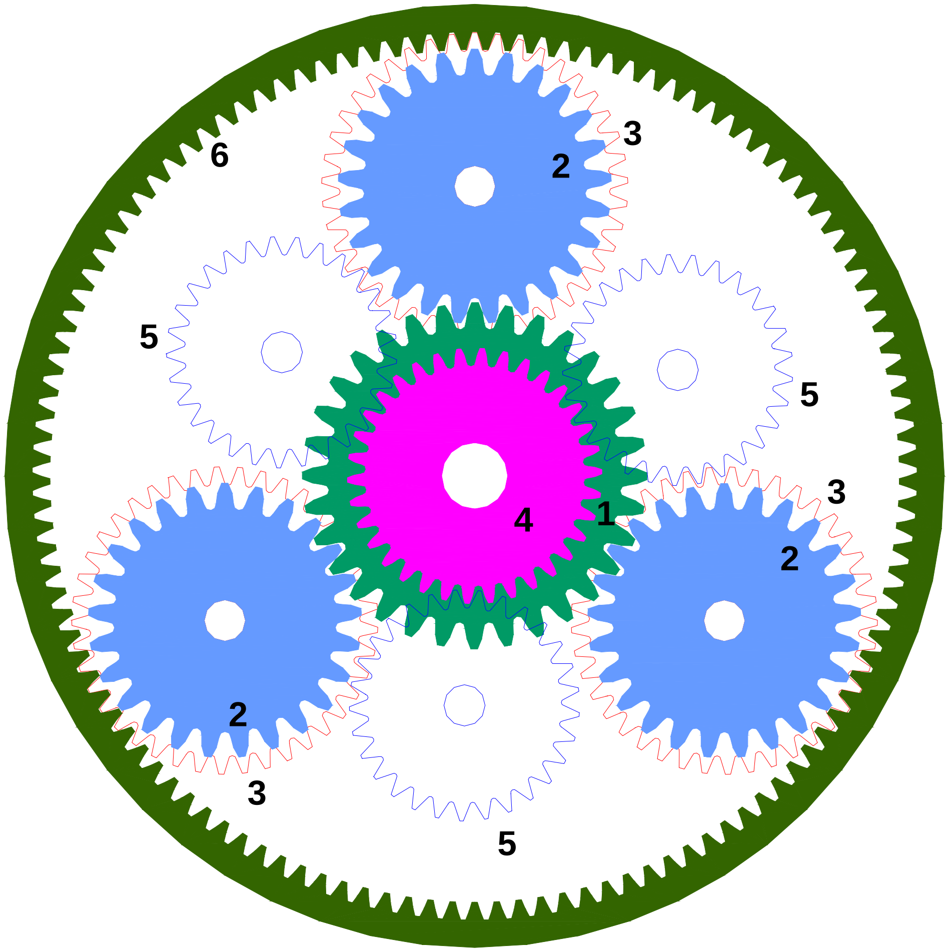


Fig. 5.8 Front view of the transmission with N_1=32, N_2=25, N_3=41, N_4=34, N_5=29, N_6=121, p=3, m_1=2.5 and m_3=1.75 mm.

ensures the least error (solution number 1 in Table 5.2) is shown drawn at scale in Figs. 5.8. It requires two of the compound planets to have gears 2 and 3 rotated relative to each other by an angle ±147.746º, which is equivalent to only ±0.117º due to the periodic profiles of the two wheels.

Solution number 6 in Table 5.2, which can be considered the most convenient to manufacture due to the identical, equally spaced planets, is shown drawn at scale in Fig. 5.9.

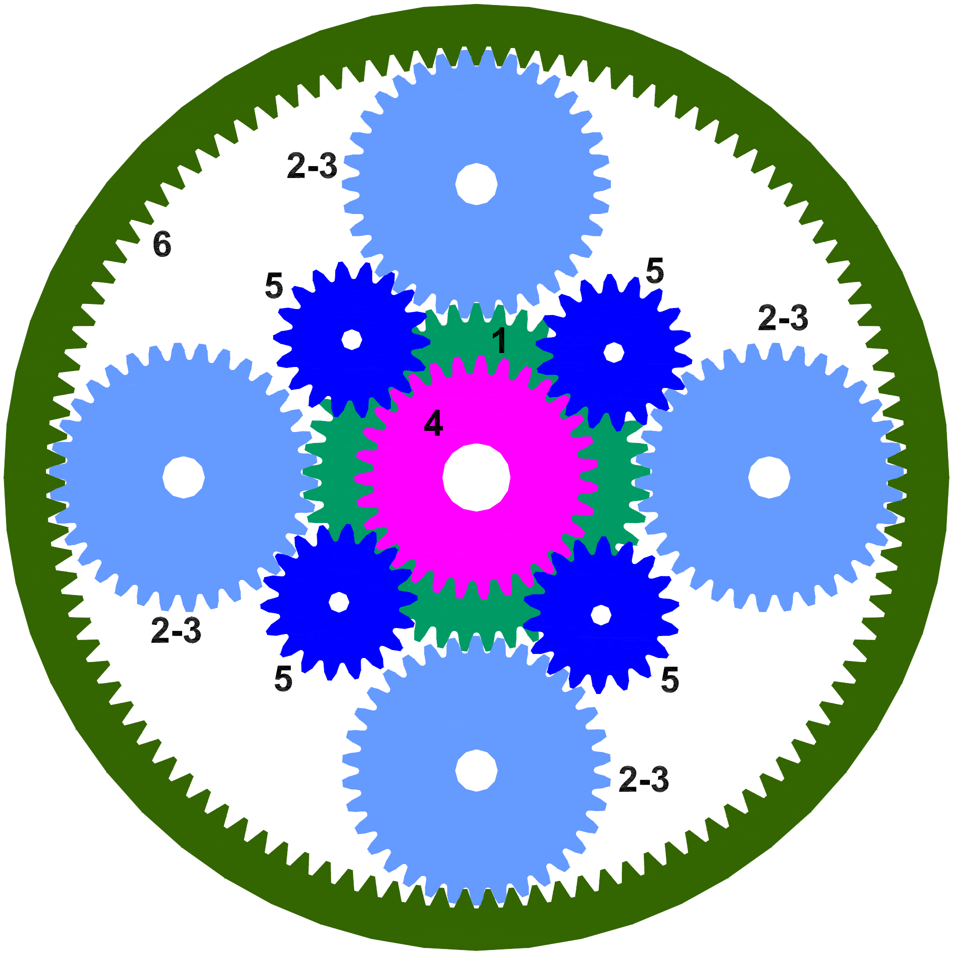


Fig. 5.9 Front view of the transmission with N_1=45, N_2=N_3=34, N_4=31, N_5=18, N_6=111, p=4 and m_1=m_3=1.75 mm.

5.6 Conclusions

The gear-teeth number synthesis of an automatic planetary transmission of the Ravigneaux type was solved with the aid of an Estimation of Distribution Algorithm. All possible assembly and interference avoidance requirements were accounted for in the form of constraints.

By allowing nonstandard involute gears to be used, an increase of the feasible domain was obtained, favorable to the design process. Visualization of the design space through partial global minima plots added insight to the synthesis problem in that it allows selecting the numerical solution based on additional requirements, like ensuring a reduced diametral size of the transmission or using unified gear cutting tools in the manufacturing process.

The presented approach can be easily extended to the teeth number synthesis of automatic planetary transmission with more than 3 forward gears that include a Ravigneaux gear-set: When only simple planetary units are associated in an automatic transmission, the number of geometric constraints will occur in lesser number, although there will be additional equally-spaced-planet assembly condition requirements that must be accounted for in the form of constraints.

5.7 References

[1] Bagci, C. (1990) “Efficient Methods for the Synthesis of Compound Planetary Differential Gear Trains for Multiple Speed Ratio Generation,” Gear Technology, Vol. 7, No. 4, pp. 16.

[2] Brewer, R.C. (1960) “Synthesis of Epicyclic Gear Trains Using the Velocity Ratio Spectrum,” Proc. of the IMechE, J. of Engineering for Industry, pp. 173-178.

[3] Chatterjee, G., and Tsai, L. W. (1995) “Enumeration of Epicyclic-Type Automatic Transmission Gear Trains,” SAE Transactions, J. of Passenger Cars, Sec. 6, Vol. 103, pp. 1415–1426.

[4] Chatterjee, G., and Tsai, L. W. (1996) “Computer Aided Sketching of Epicyclic-Type Automatic Transmission Gear Trains,” Trans. of the ASME, J. of Mechanical Design, Vol. 118, No. 3, pp. 405–411.

[5] Chong, T. H. and Lee, J.S. (2000) “Genetic Algorithm Based Design for Gear Trains,” Proc. of the ASME-DETC 2000.

[6] Cleghorn, W.L., Fenton, R.G. and Fu, J.-F. (1989) “A General Method for Optimum Design of Gear Boxes Through Nonlinear Programming,” Proc. of the ASME-DETC 1989, Vol. 19-2, pp. 153-160.

[7] Deb, K. and Jain, S. (2003) “Multi-Speed Gearbox Design Using Multi-Objective Evolutionary Algorithms,” Trans. of the ASME, J. of Mechanical Design, Vol. 125, No. 3, pp. 610-619.

[8] Fitzgeorge D. (1971) “Synthesis of Single-ratio and Multi-ratio Epicyclic Gear Trains,” Proc. of the IMechE, J. of Mechanical Engineering Science, Vol. 13, No. 6, pp. 404-415.

[9] Fogarasy, A.A. and Smith, M.R. (1995) “A New Simplified Approach to the Kinematic Analysis and Design of Epicyclic Gearboxes,” Proc. of the IMechE, J. of Mechanical Engineering Science, Vol. 209C, pp. 49-53.

[10] Gott, P.G. (1991) *Changing Gears: The Development of the Automotive Transmission*, SAE, Warendale, PA, pp. 167-172.

[11] Golinski, J. (1970) “Optimal Synthesis Problem Solved by Means of Nonlinear Programming and Random Methods,” J. of Mechanisms, Vol. 5, pp. 285-309.

[12] Hsieh, H.-I. and Tsai, L.-W. (1995) “Kinematic Analysis of Epicyclic-Type Transmission Mechanisms Using the Concept of Fundamental Geared Entities,” Proc. of the ASME-DETC 1995, Vol. 1, pp. 545-552.

[13] Hsu, C.-H. and Hsu, J.-J. (2000) ”Epicyclic Gear Trains for Automotive Automatic Transmissions” Proc. of the IMechE, J. of Automobile Engineering, Vol. 214D, No. 5, pp. 523-532.

[14] Johnson, R.C. and Towfig, K. (1967) “Creative Design of Epicyclic Gear Trains Using Number Synthesis,” Trans. of the ASME, J. of Engineering for Industry, pp. 309-314.

[15] Kelley, O.K., Jones, E.L. (1994) “Design of Planetary Gear Trains,” in *Design Practices: Passenger Car Automatic Transmissions*, SAE, Warendale, PA, pp. 167-172.

[16] Kim, S.S. and Newcombe, W.R. (1979) “Computer Aided Kinematic Design of Planetary Gear Trains,” Proc. of the 5th World Congress on the Theory of Machines and Mechanisms, Montreal, CA, pp. 148-152.

[17] Larranaga, P. and Lozano, J.A. (2002) *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*, Kluwer.

[18] Lechner, G. and Naunheimer, H. (1999) *Automotive Transmissions. Fundamentals. Selection, Design and Application*, Springer.

[19] Lloyd, R.A. (1983) “Triple Epicyclic Four-clutch Six-ratio Change Speed System,” Proc. of the IMechE., Vol. 197C, pp. 127-140.

[20] Love, P.P. (1936) “Epicyclic Gearing,” Proc. of the IMechE., Vol. 134, pp. 547-568.

[21] Macmillan, R. H. (1949) “Epicyclic Gear Trains,” The Engineer, March 25, pp. 318-320.

[22] Macmillan, R. H. (1961) “Power Flow Loss in Differential Mechanisms,” Proc. of the IMechE, J. of Mechanical Engineering Science, Vol. 3, No. 1, pp. 37-41.

[23] Macmillan, R. H. (1965) “Analytical Study of Systems for Bifurcated Power Transmission,” Proc. of the IMechE, J. of Mechanical Engineering Science, Vol. 7, No. 1, pp. 40-47.

[24] McCue, J. J. and Olson, D. G. (1990) “Optimization of Complex Planetary Gear Trains” ASME-DETC 1990, Vol. 26, pp. 57-62

[25] Meng, C.-F., Lu, X-N., Cha, J.-Z. and Shi, Z-C. (1990) “Optimal Synthesis of Planetary Chain-link Compound Mechanisms,” Proc. of the ASME-DETC 1990, Vol. 26, pp. 181-184.

[26] Merritt, H.E. (1941) “Epicyclic Gear Trains,” The Engineer, March 21, pp. 190-215.

[27] Molian, S. (1971) “Kinematics of Compound Differential Mechanisms,” Proc. of the IMechE., Vol. 185, pp. 733-739.

[28] Pazak, A., Chrobak, J. and Klimo, V. (1984) “Method of the Kinematic Synthesis of the Epicyclic Gear Trains”, International Symposium on Design and Synthesis, Tokyo, Japan, pp. 307-310

[29] Pennestrì, E. (1992) “Kinematic Synthesis of Ordinary and Epicyclic Gear Trains for a Prescribed Velocity Ratio” Proc. of the ASME-DETC 1992, Vol. 44, pp. 75-82.

[30] Pennestrì, E. and Freudenstein, F. (1993) “A Systematic Approach to Power-Flow and Static Force Analysis in Epicyclic Spur-Gear Trains,” ASME J. of Mechanical Design, Vol.115, pp.639-644.

[31] Pennestrì E. and Valentini, P. P. (2003) “A Review of Formulas for the Mechanical Efficiency Analysis of Two Degrees-of-Freedom Epicyclic Gear Trains,” Trans. of the ASME, J. of Mechanical Design, Vol. 125, No. 3, pp. 602-608.

[32] Prayoonrat, S. and Wlaton, D. (1988) “Practical Approach to Optimum Gear Train Design,” Computer-Aided Design, Vol. 20, No. 2, pp. 83-92.

[33] Ravigneaux, P. (1930) "Théorie nouvelle sur les trains épicycloïdaus et les mouvements relatifs," La Technique Automobile et Aérienne, Vol. 21, No. 151, pp. 97-106.

[34] Rogers, J.L. and Bloebaum, C.L. (1994) "Ordering Design tasks Based on Coupling Strengths," Proc. of the 5th Symposium of Multidisciplinary Analysis and Optimization, Panama City, FL, pp. 708-717.

[35] Sandgren, E. (1990) "Nonlinear integer and Discrete Programming in Mechanical Design Optimization," Trans. of the ASME, J. of Mechanical Design, Vol. 112, no. 2, pp. 223-229.

[36] Sanger, D. J. (1972) "Synthesis of Multiple-Speed Transmissions of the Planetary-Gear Type," Proc. of the IMechE, J. of Mechanical Engineering Science, Vol. 14, No. 5, pp. 353-362.

[37] Simionescu, P.A. (1998) "A Unified Approach to the Assembly Condition of Epicyclic Gears," Trans. of the ASME, J. of Mechanical Design, Vol. 120, No. 3, pp. 448-452.

[38] Simionescu, P.A., Beale D. and Dozier, G.. (2004) "Constrained Optimization problem Solving using Estimation of Distribution Algorithms," 2004 Congress on Evolutionary Computation, June 20-23, 2004, Portland, OR, pp. 296-302.

[39] Simionescu, P.A. and Beale D. (2004) "Visualization of Multivariable (Objective) Functions by Partial Global Optimization," The Visual Computer (to appear).

[40] Wilson, W.G. (1932) "Epicyclic Gearing," Proc. of the Inst. of Automobile Engineers, Vol. 26, pp. 216-257.

[41] Yan, H.-S. and Hsieh, L.-C. (1994) "Conceptual Design of Gear Differentials for Automotive Vehicles," Trans. of the ASME, J. of Mechanical Design, Vol. 116, no. 2, pp. 565-570.

CHAPTER 6. SYNTHESIS AND ANALYSIS OF THE FIVE-LINK REAR SUSPENSION SYSTEM USED IN AUTOMOBILES

In this chapter the problem of optimum kinematic synthesis and analysis of the five-link independent suspension system (also known as *multi-link* suspension mechanism symbolized 5S-5S) is investigated. The synthesis goal is fulfilling a minimum variation of the wheel-track, toe angle and camber angle during jounce and rebound of the wheel. Two variants obtained by synthesis are analyzed and compared to an existing solution, and the displacement, velocity and acceleration of the wheel carrier relative to the car body are determined, together with the variation of the momentary screw axis and the rear axle roll-center height. Both the kinematic synthesis and the analysis are performed in a simplified, easy to implement manner.

6.1. Introduction

The five-link suspension mechanism was first introduced by Deimler-Benz on their W201 and W124 series under the name "multi-link suspension" (Fig. 6.1-a). Ever since has been successfully implemented both in independent suspension systems and in rear axle guiding mechanisms by many automobile manufacturers. Due to the larger number of design parameters, it has the capability of better fulfilling the complex kinematic and dynamic requirements imposed on suspension systems of today’s automobiles. It is however much more difficult to synthesize than any other suspension mechanism, due to its general spatial configuration. In case of multi-link front suspensions the design problem is even more complex due to the fact that the kingpin is a virtual one [1] [2] corresponding to the

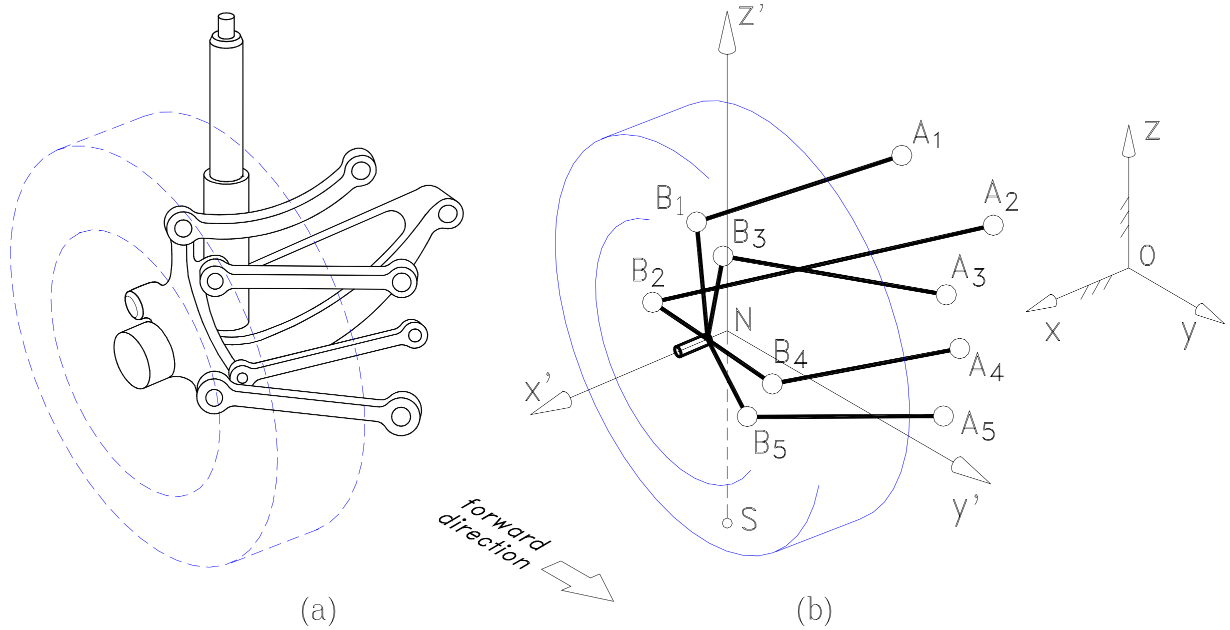


Fig. 6.1 Five-link independent suspension mechanism (**a**) and its kinematic diagram (**b**).

momentary screw axis of the wheel-carrier performing the steering motion relative to the chassis.

The rear independent wheel or axle guiding mechanism(s) are, in the sense of Mechanism Theory, spatial motion generators (also known as rigid body guidance mechanisms). Research on motion generators synthesis and analysis has been carried out on both abstract and applied mechanisms by many researchers in the past. A general formulation of the mechanism synthesis problem for path, function and rigid-body guidance based on optimization techniques was proposed by Aviles et al. [3]. According to the authors, a global error function to be minimized was defined as a weighted-sum of some local error functions, previously minimized with respect to the Cartesian coordinates of the "basic-points" of the mechanism. The so called "basic points" are the centers of the joints and the points of the links required to generate certain paths throughout the working range of the mechanism. Although the method is general in its formulation, the main disadvantage lays in the large number of variables required to define the objective function, as well as in not including the ground joint coordinates among the design parameters.

An extension of the approach of Aviles et al. to the synthesis of spatial linkages was given by Jimenez et al. [4]. However, the main drawback of an excessive number of design variables required in formulating the synthesis problem was maintained. In the example presented of synthesizing a 5S-5S suspension mechanism for only 3 prescribed positions of the wheel-carrier, an objective function of 64 variables was defined.

Suh [5] synthesized a double wishbone suspension mechanism (an RSSR-SS spatial motion generator), which can be considered a particular embodiment of the five-link suspension [1], in a mixed approach with two finite and instantaneous exact positions, using displacement matrices and constraint equations.

A combination of exact and approximate synthesis was performed by Sandor et al. [6] for the same RSSR-SS motion generating mechanism. The authors considered part of the positions imposed to the wheel-carrier as exact positions and solved the corresponding set of equations. The free choices in these equations were further considered design variables in an objective function, penalized with the conditions of avoiding branching, achieving correct sequence of prescribed positions and observing the shortest and longest links to be within prescribed limits in the remaining imposed positions.

The method detailed in this dissertation for kinematic synthesis of the 5S-5S rigid body guidance mechanism assumes the guided body (the wheel-carrier) released from its joints and moving in successive positions along the ideal trajectory. The synthesis problem thus becomes that of finding the joint disposition for which the distances between the homologous released joints (the pair joints that in the real mechanism are connected by binary links) vary as little as possible. This is a variant of the finite-position spatial theory of kinematic synthesis [7] [8] [9] the object of which is determining those points which lie on special loci: spheres, cylinders, circles, lines etc.

The same approach of considering the wheel-carrier released from its joints was also successfully applied for displacement analysis of the same mechanism. The interested reader can find this procedure directly applicable to solving the direct kinematic problem of a variety of parallel mechanisms of the Gough-Stewart type.

6.2. Synthesis Problem Formulation

The requirements upon the motion of the rear wheel that can be transposed into kinematic conditions when synthesizing the suspension mechanism are [10]:

-minimum toe angle variation during compression and rebound;

-avoid excessive outward camber thrust on corners;

-avoid excessive sideways thrust and consequent rear-end steering impulses on single wheel bump or rebounds.

-in addition, the suspension elements must ensure a minimum intrusion into the passengers and luggage compartment, a condition that can be translated into constraints imposed to the possible disposition of the ball joints on the chassis and on the wheel carrier.

The effect of the compliance of the rear wheel suspension upon the car ride behavior is important and in the final design must necessarily be considered by performing a dynamic simulation of the whole vehicle. However, in order to simplify both the kinematic synthesis and analysis procedures, it is common in the early stages of design to assume that the joints have neither clearances nor elasticities, and the vehicle chassis and suspension elements are rigid. When equipped with compliant-joints, it is to be expected that a rigid joint suspension that exhibit good kinematic characteristics, will continue to perform satisfactory (provided that the stiffness rates of the joints are properly selected).

Taking the first three above-mentioned conditions imposed to a suspension system, it can be considered that the ideal wheel movement along its operation travel must be close to a vertical translation relative to the car body. This is in accordance with Raghavan's findings [11] that for straight-line motion of the car, the motion of the wheel relative to the road should exhibit zero toe and camber change, and that track width should be maintained constant.

In order to formulate the synthesis problem, all the five links are removed from their joints (or assumed of variable length), thus allowing the wheel carrier to be displaced in successive positions along any trajectory. If the distance between the homologous joints varies very little in these successive positions, the real mechanism with the five links jointed back in place will guide the wheel very close to the imposed path.

The above considerations are the basis for formulating the synthesis of the five-link mechanism as an optimization problem, i.e. of finding the minimum of the following objective function of 30 variables (Fig. 6.1-b):

$$\mathrm{F}(\mathrm{x}_{\mathrm{A}i}, \mathrm{y}_{\mathrm{A}i}, \mathrm{z}_{\mathrm{A}i}, \mathrm{x'}_{\mathrm{B}i}, \mathrm{y'}_{\mathrm{B}i}, \mathrm{z'}_{\mathrm{B}i})_{i=1..5} = \sum_{i=1}^{5}\sum_{j=1}^{n}\left[l_i - (\mathrm{A}_i\mathrm{B}_i)_j\right]^2 \qquad (6.1)$$

with j=1..n intermediate positions of the wheel carrier evenly spaced on the prescribed trajectory.

In the followings this imposed trajectory will be a simple vertical translation of the wheel carrier i.e. x_N and y_N are kept constant for $\mathrm{z}_{\mathrm{N}j}$ varying between a lower $\mathrm{z}_{\mathrm{Nmin}}$ and an upper $\mathrm{z}_{\mathrm{Nmax}}$ limit of point N attached to the wheel carrier. One should not expect that the mechanism obtained by synthesis to exactly generate this pure vertical motion. As will be seen later, the kinematic behavior of the synthesized mechanism depends noticeably on the values chosen for

z_{Nmin} and z_{Nmax} (which should not necessarily be the upper and lower limits of wheel travel during jounce and rebound, nor even belonging to the actual motion range of the wheel-carrier).

The reference lengths of the links noted l_i (i=1,5) in relation (6.1) are determined as the distances between the joints A_i and B_i for the wheel in its initial position, corresponding to the car averagely loaded and in rest. The variable distances $(\mathrm{A}_i\mathrm{B}_i)_j$ between the five homologous joints A_i and B_i in a current position j of the wheel-carrier is given by:

$$(\mathrm{A}_i\mathrm{B}_i)_j = \sqrt{(\mathrm{x}_{\mathrm{A}ij} - \mathrm{x}_{\mathrm{B}ij})^2 + (\mathrm{y}_{\mathrm{A}ij} - \mathrm{y}_{\mathrm{B}ij})^2 + (\mathrm{z}_{\mathrm{A}ij} - \mathrm{z}_{\mathrm{B}ij})^2} \tag{6.2}$$

where the coordinates x, y and z must be specified relative to the same reference frame, preferable the fixed reference frame Oxyz. Because the disposition of the ball-joint centers B_i is given in the reference frame attached to the wheel carrier Nx'y'z', the following transformations must be applied in order to make use of equation (6.2):

$$\begin{bmatrix} \mathrm{x}_{\mathrm{B}ij} \\ \mathrm{y}_{\mathrm{B}ij} \\ \mathrm{z}_{\mathrm{B}ij} \end{bmatrix}_{\mathrm{Oxyz}} = \begin{bmatrix} \mathrm{x'}_{\mathrm{B}i} \\ \mathrm{y'}_{\mathrm{B}i} \\ \mathrm{z'}_{\mathrm{B}i} \end{bmatrix}_{\mathrm{Nx'y'z'}} + \begin{bmatrix} \mathrm{x}_{\mathrm{N}j} \\ \mathrm{y}_{\mathrm{N}j} \\ \mathrm{z}_{\mathrm{N}j} \end{bmatrix}_{\mathrm{Oxyz}} . \tag{6.3}$$

In the initial position, the reference frame Oxyz attached to the chassis and the frame Nx'y'z' attached to the wheel-carrier are considered parallel. Knowing the coordinates (x_{N0}, y_{N0}, z_{N0}) of the origin of Nx'y'z' frame relative to the chassis reference frame, the coordinates of the same point N relative to Oxyz reference frame will be (x_{N0}, y_{N0}, $z_{\mathrm{N}j}$) for a current prescribed position j, where $z_{\mathrm{N}j}=z_{\mathrm{Nmin}}+\Delta z_{\mathrm{N}j}$ with $\Delta z_{\mathrm{N}j}=j\cdot(z_{\mathrm{Nmax}}-z_{\mathrm{Nmin}})/n$.

The limitations upon the possible locations of the ball joints on the chassis and wheel-carrier can be prescribed as side constraints of the form:

$$\begin{array}{ccccc} x_{Ai\min} & \leq & x_{Ai} & \leq & x_{Ai\max} \\ y_{Ai\min} & \leq & y_{Ai} & \leq & y_{Ai\max} \quad (i=1..5) \\ z_{Ai\min} & \leq & z_{Ai} & \leq & z_{Ai\max} \end{array} \tag{6.4}$$

and:

$$\begin{array}{ccccc} x'_{Bi\min} & \leq & x'_{Bi} & \leq & x'_{Bi\max} \\ y'_{Bi\min} & \leq & y'_{Bi} & \leq & y'_{Bi\max} \quad (i=1..5). \\ z'_{Bi\min} & \leq & z'_{Bi} & \leq & z'_{Bi\max} \end{array} \tag{6.5}$$

and must necessarily be imposed in order to avoid convergence to unpractical solutions with links excessively long.

The objective function (6.1) together with the constraints (6.4) and (6.5) can be minimized using a proper optimization subroutine. Of the maximum number of design variables (30 in total - irrespective of the number of intermediate positions *n* of the wheel carrier), some of the ball-joint centers can be imposed fixed values and the number of design variables further reduced.

In theory it is possible to prescribe a trajectory to the wheel-carrier that can be exactly generated by a real mechanism (case in which the global minima of the objective function F will be zero). In practice however, there will always be a departure between the prescribed motion and the actual motion of the real mechanism. Therefore a kinematic analysis is required in order to determine the actual behavior of the suspension mechanism obtained by synthesis.

6.3. Kinematic Analysis of the Five-Link Suspension Mechanism

The analysis of the five-link suspension mechanism has been tackled by a number of researchers in the past. Lee et al. [1] derived the velocity equations of the wheel carrier and applied a step-wise linearization to solve the position problem. Mohamed and Attia [12] used the constrained equations obtained from the condition that the five connecting rods and the wheel-carrier are rigid bodies. Knapzyk and Dzierzec [13] considered a modified mechanism with two of the guiding links disassembled and solved an optimization problem describing the condition that the distances between the homologous released joints remain equal to the lengths of the disconnected members.

Following [6], Unkoo and Byeongeui [2] used 4×4 displacement and differential-displacement matrices and constraint equations to solve the position and velocity problem of five-link and strut-type multi-link suspensions. The referred authors also determined the imaginary kingpin axis of these suspensions systems using screw-axis theory and compared the results with those obtained by finite-center analysis.

The same approach of considering all the five connecting rods removed will be further considered. For successive values of the input parameter z_N, the position of the point N relative to the horizontal axis and the orientation angles of the wheel-carrier will be tuned in a searching process, until the distances between the released joints A_i and B_i become equal (within some error limits) to the lengths of the respective links A_iB_i.

6.3.1 Position Problem

The five-link suspension mechanism has 6 degrees-of-freedom, of which 5 are trivial rotations of the connecting links around their own axes. Correspondingly, the position of the wheel carrier can be specified using only one independent parameter viz the coordinate z_N of the origin of the Nx'y'z' reference frame relative to the central reference frame Oxyz. The remaining 5 parameters: coordinates x_N, y_N and angles α, β and γ that define the position and orientation of the wheel carrier can be determined by solving the following equations of constraint:

$$(x_{Ai}-x_{Bi})^2+(y_{Ai}-y_{Bi})^2+(z_{Ai}-z_{Bi})^2=l_i^2 \quad (i=1..5) \tag{6.6}$$

describing the condition of the distance between joints A_i and B_i to remain constant during the working range of the mechanism. In the above equation (6.6), the coordinates x_{Bi}, y_{Bi} and z_{Bi} are determined by applying the following transformation to the Nx'y'z' reference frame:

$$\begin{bmatrix} x_{Bi} \\ y_{Bi} \\ z_{Bi} \end{bmatrix}_{Oxyz} = \left[R_{\beta\alpha\gamma}\right] \cdot \begin{bmatrix} x'_{Bi} \\ y'_{Bi} \\ z'_{Bi} \end{bmatrix}_{Nx'y'z'} + \begin{bmatrix} x_N \\ y_N \\ z_N \end{bmatrix}_{Oxyz} . \tag{6.7}$$

where $[R_{\beta\alpha\gamma}]$ is the transformation matrix that express the successive rotation of the wheel-carrier relative to Oxyz by the pitch angle β, yaw angle α and roll angle γ [14]:

$$\left[R_{\beta\alpha\gamma}\right]=\left[R_{\gamma,x}\right]\left[R_{\alpha,z}\right]\left[R_{\beta,y}\right]= \\ \begin{bmatrix} c\alpha\cdot c\beta & -s\alpha & c\alpha\cdot s\beta \\ s\alpha\cdot c\beta\cdot c\gamma+s\beta\cdot s\gamma & c\alpha\cdot c\gamma & s\alpha\cdot s\beta\cdot c\gamma+c\beta\cdot s\gamma \\ s\alpha\cdot c\beta\cdot s\gamma+s\beta\cdot c\gamma & c\alpha\cdot s\gamma & s\alpha\cdot s\beta\cdot s\gamma+c\beta\cdot c\gamma \end{bmatrix}. \tag{6.8}$$

In the above equation $[R_{\alpha,z}]$, $[R_{\beta,y}]$ and $[R_{\gamma,x}]$ are the basic rotation matrices while cα=cosα, sα=sinα and so forth.

For a given value of the independent parameter z_N, the system of equations (6.6) in the unknowns α, β, γ, x_N and y_N can be very conveniently solved by minimizing the following objective function:

$$F_0(\alpha,\beta,\gamma,x_N,y_N)=\sum_{i=1}^{5}\left[(x_{Ai}-x_{Bi})^2+(y_{Ai}-y_{Bi})^2+(z_{Ai}-z_{Bi})^2-l_i^2\right]. \tag{6.9}$$

In order to facilitate convergence, the starting point when minimizing F_0 can be taken the position of the wheel-carrier (the same x_N, y_N and orientation angles α,β,γ) imposed during synthesis for the same z_{Nj} . Once the displacement problem of the wheel carrier is solved, the diagram of the wheel track, recessional wheel motion, camber and toe-angle alteration can be generated.

6.3.2 Linear Velocity and Acceleration Analysis

The velocities of points B$_i$ on the wheel carrier can be determined by differentiating once with respect to time the equations of constraint (6.6). The number of unknowns thus emerging is 15, and therefore 10 more equations must be added, like the time derivatives of following equations:

$$\begin{gathered}(\mathrm{x}_{\mathrm{B}j}-\mathrm{x}_{\mathrm{B}k})^2+(\mathrm{y}_{\mathrm{B}j}-\mathrm{y}_{\mathrm{B}k})^2+(\mathrm{z}_{\mathrm{B}j}-\mathrm{z}_{\mathrm{B}k})^2=\text{constant}\\ (j=1...4\ \text{ and }\ k=j+1...5)\end{gathered}\quad . \tag{6.10}$$

and:

$$(\mathrm{x}_{\mathrm{B}i}-\mathrm{x}_{\mathrm{N}})^2+(\mathrm{y}_{\mathrm{B}i}-\mathrm{y}_{\mathrm{N}})^2+(\mathrm{z}_{\mathrm{B}i}-\mathrm{z}_{\mathrm{N}})^2=\text{constant}\qquad (i=1...5)\,. \tag{6.11}$$

describing the condition that the wheel carrier is a rigid body. By differentiation these equations once with respect to time, a new independent parameter $\dot{\mathrm{z}}_{\mathrm{N}}$ will emerge, which, the same as z$_{\mathrm{N}}$ must be specified as input during the numerical analysis. The coefficients of the system of linear equations in the 17 unknowns $\dot{\mathrm{x}}_{\mathrm{B}i}, \dot{\mathrm{y}}_{\mathrm{B}i}, \dot{\mathrm{z}}_{\mathrm{B}i}$ (*i*=1..5), $\dot{\mathrm{x}}_{\mathrm{N}}$ and $\dot{\mathrm{y}}_{\mathrm{N}}$ used for velocity analysis are summarized in Table 6.1.

By differentiating with respect to time the equations used to solve the velocity problem, a second system of linear equations in the unknowns $\ddot{\mathrm{x}}_{\mathrm{B}i}, \ddot{\mathrm{y}}_{\mathrm{B}i}, \ddot{\mathrm{z}}_{\mathrm{B}i}$ (*i*=1..5), $\ddot{\mathrm{x}}_{\mathrm{N}}$ and $\ddot{\mathrm{y}}_{\mathrm{N}}$ will be further obtained, the coefficients of which are given in Table 6.2. In this case z$_{\mathrm{N}}$, $\dot{\mathrm{z}}_{\mathrm{N}}$ and $\ddot{\mathrm{z}}_{\mathrm{N}}$ will be the independent parameters that must be specified as inputs.

Table 6.1 The coefficients of the linear system of equations used to determine the linear velocity of points B$_i$ (i=1..5).

$\dot{x}_{B1}$	$\dot{y}_{B1}$	$\dot{z}_{B1}$	$\dot{x}_{B2}$	$\dot{y}_{B2}$	$\dot{z}_{B2}$	$\dot{x}_{B3}$	$\dot{y}_{B3}$	$\dot{z}_{B3}$	$\dot{x}_{B4}$	$\dot{y}_{B4}$	$\dot{z}_{B4}$	$\dot{x}_{B5}$	$\dot{y}_{B5}$	$\dot{z}_{B5}$	$\dot{x}_{N}$	$\dot{y}_{N}$	
xA_1-xB_1	yA_1-yB_1	zA_1-zB_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	xA_2-xB_2	yA_2-yB_2	zA_2-zB_2	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0		xA_3-xB_3	yA_3-yB_3	zA_3-zB_3	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	xA_4-xB_4	yA_4-yB_4	zA_4-zB_4	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	xA_5-xB_5	yA_5-yB_5	zA_5-zB_5	0	0	0
xB_1-xB_2	yB_1-yB_2	zB_1-zB_2	xB_2-xB_1	yB_2-yB_1	zB_2-zB_1	0	0	0	0	0	0	0	0	0	0	0	0
xB_1-xB_3	yB_1-yB_3	zB_1-zB_3	0	0	0	xB_3-xB_1	yB_3-yB_1	zB_3-zB_1	0	0	0	0	0	0	0	0	0
xB_1-xB_4	yB_1-yB_4	zB_1-zB_4	0	0	0	0	0	0	xB_4-xB_1	YB_4-yB_1	zB_4-zB_1	0	0	0	0	0	0
xB_1-xB_5	yB_1-yB_5	zB_1-zB_5	0	0	0	0	0	0	0	0	0	xB_5-xB_1	yB_5-yB_1	zB_5-zB_1	0	0	0
0	0	0	xB_2-xB_3	yB_2-yB_3	zB_2-zB_3	xB_3-xB_2	yB_3-yB_2	zB_3-zB_2	0	0	0	0	0	0	0	0	0
0	0	0	xB_2-xB_4	yB_2-yB_4	zB_2-zB_4	0	0	0	xB_4-xB_2	YB_4-yB_2	zB_4-zB_2	0	0	0	0	0	0
0	0	0	xB_2-xB_5	yB_2-yB_5	zB_2-zB_5	0	0	0	0	0	0	xB_5-xB_2	yB_5-yB_2	zB_5-zB_2	0	0	0
xB_1-xN	yB_1-yN	zB_1-zN	0	0	0	0	0	0	0	0	0	0	0	0	$xN-xB_1$	$yN-yB_1$	$(zB_1-zN)\,\dot{z}_N$
0	0	0	xB_2-xN	yB_2-yN	ZB_2-zN	0	0	0	0	0	0	0	0	0	$xN-xB_2$	$yN-yB_2$	$(zB_2-zN)\,\dot{z}_N$
0	0	0	0	0		xB_3-xN	yB_3-yN	zB_3-zN	0	0	0	0	0	0	$xN-xB_3$	$yN-yB_3$	$(zB_3-zN)\,\dot{z}_N$
0	0	0	0	0	0	0	0	0	xB_4-xN	YB_4-yN	zB_4-zN	0	0	0	$xN-xB_4$	$yN-yB_4$	$(zB_4-zN)\,\dot{z}_N$
0	0	0	0	0	0	0	0	0	0	0	0	xB_5-xN	yB_5-yN	zB_5-zN	$xN-xB_5$	$yN-yB_5$	$(zB_5-zN)\,\dot{z}_N$

Table 6.2[♦] The coefficients of the linear system of equations used to determine the linear accelerations of points B$_i$ (i=1..5).

$\ddot{x}_{B1}$	$\ddot{y}_{B1}$	$\ddot{z}_{B1}$	$\ddot{x}_{B2}$	$\ddot{y}_{B2}$	$\ddot{z}_{B2}$	$\ddot{x}_{B3}$	$\ddot{y}_{B3}$	$\ddot{z}_{B3}$	$\ddot{x}_{B4}$	$\ddot{y}_{B4}$	$\ddot{z}_{B4}$	$\ddot{x}_{B5}$	$\ddot{y}_{B5}$	$\ddot{z}_{B5}$	$\ddot{x}_{N}$	$\ddot{y}_{N}$	
*	*	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\dot{x}_{B1}^2+\dot{y}_{B1}^2+\dot{z}_{B1}^2$
0	0	0	*	*	*	0	0	0	0	0	0	0	0	0	0	0	$\dot{x}_{B2}^2+\dot{y}_{B2}^2+\dot{z}_{B2}^2$
0	0	0	0	0	0	*	*	*	0	0	0	0	0	0	0	0	$\dot{x}_{B3}^2+\dot{y}_{B3}^2+\dot{z}_{B3}^2$
0	0	0	0	0	0	0	0	0	*	*	*	0	0	0	0	0	$\dot{x}_{B4}^2+\dot{y}_{B4}^2+\dot{z}_{B4}^2$
0	0	0	0	0	0	0	0	0	0	0	0	*	*	*	0	0	$\dot{x}_{B5}^2+\dot{y}_{B5}^2+\dot{z}_{B5}^2$
*	*	*	*	*	*	0	0	0	0	0	0	0	0	0	0	0	$-(\dot{x}_{B1}-\dot{x}_{B2})^2-(\dot{y}_{B1}-\dot{y}_{B2})^2-(\dot{z}_{B1}-\dot{z}_{B2})^2$
*	*	*	0	0	0	*	*	*	0	0	0	0	0	0	0	0	$-(\dot{x}_{B1}-\dot{x}_{B3})^2-(\dot{y}_{B1}-\dot{y}_{B3})^2-(\dot{z}_{B1}-\dot{z}_{B3})^2$
*	*	*	0	0	0	0	0	0	*	*	*	0	0	0	0	0	$-(\dot{x}_{B1}-\dot{x}_{B4})^2-(\dot{y}_{B1}-\dot{y}_{B4})^2-(\dot{z}_{B1}-\dot{z}_{B4})^2$
*	*	*	0	0	0	0	0	0	0	0	0	*	*	*	0	0	$-(\dot{x}_{B1}-\dot{x}_{B5})^2-(\dot{y}_{B1}-\dot{y}_{B5})^2-(\dot{z}_{B1}-\dot{z}_{B5})^2$
0	0	0	*	*	*	*	*	*	0	0	0	0	0	0	0	0	$-(\dot{x}_{B2}-\dot{x}_{B3})^2-(\dot{y}_{B2}-\dot{y}_{B3})^2-(\dot{z}_{B2}-\dot{z}_{B3})^2$
0	0	0	*	*	*	0	0	0	*	*	*	0	0	0	0	0	$-(\dot{x}_{B2}-\dot{x}_{B4})^2-(\dot{y}_{B2}-\dot{y}_{B4})^2-(\dot{z}_{B2}-\dot{z}_{B4})^2$
0	0	0	*	*	*	0	0	0	0	0	0	*	*	*	0	0	$-(\dot{x}_{B2}-\dot{x}_{B5})^2-(\dot{y}_{B2}-\dot{y}_{B5})^2-(\dot{z}_{B2}-\dot{z}_{B5})^2$
*	*	*	0	0	0	0	0	0	0	0	0	0	0	0	*	*	$\ddot{z}_N(z_{B1}-z_N)-(\dot{x}_{B1}-\dot{x}_N)^2-(\dot{y}_{B1}-\dot{y}_N)^2-(\dot{z}_{B1}-\dot{z}_N)^2$
0	0	0	*	*	*	0	0	0	0	0	0	0	0	0	*	*	$\ddot{z}_N(z_{B2}-z_N)-(\dot{x}_{B2}-\dot{x}_N)^2-(\dot{y}_{B2}-\dot{y}_N)^2-(\dot{z}_{B2}-\dot{z}_N)^2$
0	0	0	0	0	0	*	*	*	0	0	0	0	0	0	*	*	$\ddot{z}_N(z_{B3}-z_N)-(\dot{x}_{B3}-\dot{x}_N)^2-(\dot{y}_{B3}-\dot{y}_N)^2-(\dot{z}_{B3}-\dot{z}_N)^2$
0	0	0	0	0	0	0	0	0	*	*	*	0	0	0	*	*	$\ddot{z}_N(z_{B4}-z_N)-(\dot{x}_{B4}-\dot{x}_N)^2-(\dot{y}_{B4}-\dot{y}_N)^2-(\dot{z}_{B4}-\dot{z}_N)^2$
0	0	0	0	0	0	0	0	0	0	0	0	*	*	*	*	*	$\ddot{z}_N(z_{B5}-z_N)-(\dot{x}_{B5}-\dot{x}_N)^2-(\dot{y}_{B5}-\dot{y}_N)^2-(\dot{z}_{B5}-\dot{z}_N)^2$

[♦] The star entries in Table 6.2 designate coefficients identical to the corresponding ones in Table 6.1.

6.3.3 Angular Velocity and Angular Acceleration Analysis

The components of the angular-velocity vector (ω_x, ω_y, ω_z) relative to the fixed reference frame Oxyz can be determined using the following matrix equation known from the rigid body kinematics:

$$\begin{bmatrix} \dot{x}_{Bi} \\ \dot{y}_{Bi} \\ \dot{z}_{Bi} \end{bmatrix} = \begin{bmatrix} \dot{x}_N \\ \dot{y}_N \\ \dot{z}_N \end{bmatrix} + \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times \begin{bmatrix} x_{Bi} - x_N \\ y_{Bi} - y_N \\ z_{Bi} - z_N \end{bmatrix} \tag{6.12}$$

written for any two different points of the wheel carrier for which the linear velocity are known. The expressions of ω_x, ω_y and ω_z as derived from relation (6.12) are given in equations (A3.1) in Appendix 3.

The components of the angular-acceleration vector (ε_x, ε_y, ε_z) can be determined writing the following equation, the same for two different points of the wheel carrier:

$$\begin{bmatrix} \ddot{x}_{Bi} \\ \ddot{y}_{Bi} \\ \ddot{z}_{Bi} \end{bmatrix} = \begin{bmatrix} \ddot{x}_N \\ \ddot{y}_N \\ \ddot{z}_N \end{bmatrix} + \begin{bmatrix} \varepsilon_x \\ \varepsilon_x \\ \varepsilon_x \end{bmatrix} \times \begin{bmatrix} x_{Bi} - x_N \\ y_{Bi} - y_N \\ z_{Bi} - z_N \end{bmatrix} + \begin{bmatrix} \omega_x \\ \omega_x \\ \omega_x \end{bmatrix} \times \left(\begin{bmatrix} \omega_x \\ \omega_x \\ \omega_x \end{bmatrix} \times \begin{bmatrix} x_{Bi} - x_N \\ y_{Bi} - y_N \\ z_{Bi} - z_N \end{bmatrix} \right). \tag{6.13}$$

The expressions of ε_x, ε_y and ε_z derived through analytical manipulations of relation (6.13) are summarized in equations (A3.2) in Appendix 3. Alternatively, the components of the angular acceleration can be determined by differentiating once with respect to time the components of the angular velocity:

$$\varepsilon_x = \dot{\omega}_x, \quad \varepsilon_y = \dot{\omega}_y \quad \text{and} \quad \varepsilon_z = \dot{\omega}_z. \tag{6.14}$$

The results of velocity and acceleration analysis will be further used in determining the location of the instantaneous screw axis of the wheel-carrier moving relative to the chassis. Position, velocity and acceleration problems are also stages required in solving the dynamic problem of the suspension system. According to Hiller [15], of the total CPU time needed to simulate the response of a five-link suspension to a road input, almost 70% was required for solving the kinematics of the system. The method described above for solving the position problem it is likely to reduce this amount of time, since requires solving a system of only 5 nonlinear equations.

6.3.4 Instantaneous Screw Axis

Considering the instantaneous motion of the five-link suspension, the wheel-carrier motion relative to the car body is a screw motion of the circle-point-surface fixed to the wheel-carrier

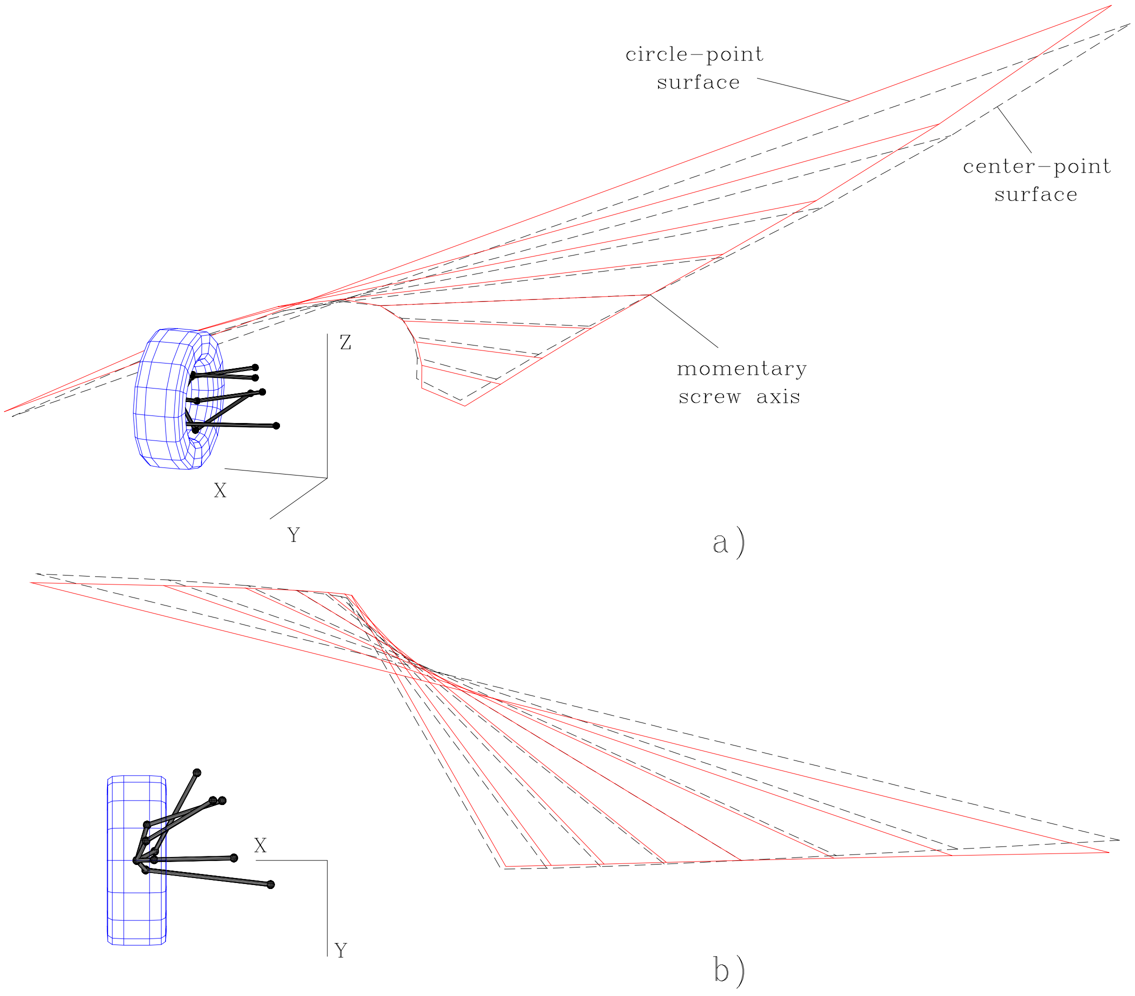


Fig. 6.2 Center-point-surface and circle-point-surface of a five-link independent suspension solution **2** in paragraph 4 in perspective view (**a**) and top view (**b**).

with respect to the center-point-surface fixed to the car body [16]. The common tangent of these two surfaces is the instantaneous screw axis of the spatial motion (see Fig. 6.2), and corresponds to the points of minimum velocity of the wheel carrier relative to the car body. Therefore, the parameters positioning the momentary screw axis can very well be determined by formulating a minimization problem.

A different approach is to solve the system of equations expressing the condition the linear velocity ($\dot{x}, \dot{y}, \dot{z}$) of a point (x,y,z) attached to the wheel carrier is parallel to the angular velocity vector (ω_x,ω_y,ω_z):

$$\dot{x}/\omega_x = \dot{y}/\omega_y = \dot{z}/\omega_z \tag{6.15}$$

Based on equation (6.12), the above equalities becomes:

$$\begin{aligned} \frac{\dot{x}_N + \omega_y(z - z_N) - \omega_z(y - y_N)}{\omega_x} &= \frac{\dot{y}_N + \omega_z(x - x_N) - \omega_x(z - z_N)}{\omega_y} \\ \frac{\dot{x}_N + \omega_y(z - z_N) - \omega_z(y - y_N)}{\omega_x} &= \frac{\dot{z}_N + \omega_x(y - y_N) - \omega_y(x - x_N)}{\omega_z} \end{aligned} \tag{6.16}$$

The resulting expressions of the parametric equation of the momentary screw axis x(y) and z(y) are given in equations (A3.6) in Appendix 3.

Determining of the screw due to steering input is of much significant importance in the case of the five-link suspension mechanism used for guiding the front wheels, which has a second DOF corresponding to the steering input. When only the steering input is active, the resulting momentary screw axis (which can be calculated following a similar approach) will be the virtual kingpin of the wheel during the steering motion [2].

6.3.5 Suspension Roll Center

Each suspension has a roll center defined as the point in the vertical plane through the wheel centers at which lateral forces may be applied to the sprung mass without producing suspension roll [17] [18]. According to Reimpell and Stoll [19] there is a direct correlation between the wheel track variation and the roll-center height h_R. According to the same authors, this correlation is also conflicting, in that a high roll center (which is desirable for a favorable car body attitude during cornering) implies a larger track alteration. The suspension roll center can be approximately determine by finite-center analysis as the intersection between the normal to the trajectory of the path center point S projected on the vertical plane Oxz and the car's longitudinal plane Oyz (Fig. 6.3). The following formula:

$$h_R(z_{Nj}) = 0.5\frac{x_S^2(z_{Nj}) - x_S^2(z_{Nj+1}) + z_S^2(z_{Nj}) - z_{Sj}^2(z_{Nj+1})}{z_S(z_{Nj}) - z_{Sj}(z_{Nj})} \tag{6.17}$$

has been derived for calculating the roll-center height relative to the chassis reference frame. The height of the roll-center measured from the ground will be:

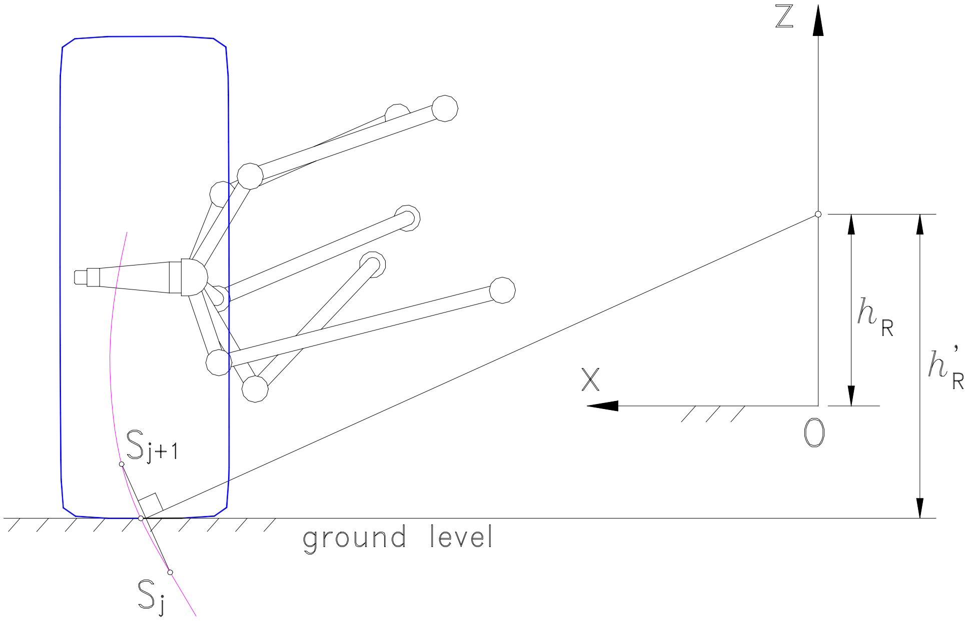


Fig. 6.3 Schematic for calculating the roll-center height of the rear axle.

$$h_R(z_{Nj}) = h'_R(z_{Nj}) - x_S^2(z_{Nj}) \qquad (6.18)$$

In the above equations z_{Nj} and z_{Nj+1} are two successive positions of the wheel center, sufficiently close one to the other to allow a tangent-chord approximation along the trajectory of the path center point.

6.4. Numerical Results

Based on the procedure described above, the synthesis of a five-link rear wheel independent suspension system was performed. The numerical data corresponding to the Mercedes-190 multi-link suspension available [13] was used in defining the allowable positions of the ball-joint centers (Table 6.3).

The origin of the Nx'y'z' coordinate system in the reference position was x_{N0}=705mm, y_{N0}=0 and z_{N0}=302mm, while the wheel radius was R=314mm.

Two intervals of the wheel-carrier vertical travel have been considered in the objective function F. The first numerical solution recorded (Table 6.4) was obtained for the reference frame Nx'y'z' translating verticaly between z_{Nmin}=-50mm and z_{Nmax}=100mm. The second

Table 6.3 Side constraints of the design variables x_{Ai}, y_{Ai}, z_{Ai}, x_{Bi}, y_{Bi}, z_{Bi} (i=1..5).

$190 \le x_{A1} \le 220$	$87 \le y_{A1} \le 117$	$216 \le z_{A1} \le 246$
$481 \le x_{A2} \le 511$	$-336 \le y_{A2} \le -306$	$236 \le z_{A2} \le 266$
$389 \le x_{A3} \le 419$	$-224 \le y_{A3} \le -194$	$281 \le z_{A3} \le 311$
$422 \le x_{A4} \le 452$	$-224 \le y_{A4} \le -194$	$387 \le z_{A4} \le 417$
$341 \le x_{A5} \le 371$	$-10 \le y_{A5} \le 20$	$401 \le z_{A5} \le 431$
$-53 \le x'_{B1} \le -33$	$33 \le y'_{B1} \le 53$	$-104 \le z'_{B1} \le -84$
$-83 \le x'_{B2} \le -63$	$-54 \le y'_{B2} \le -34$	$-149 \le z'_{B2} \le -129$
$-49 \le x'_{B3} \le -29$	$-151 \le y'_{B3} \le -131$	$-43 \le z'_{B3} \le -23$
$-53 \le x'_{B4} \le -33$	$-88 \le y'_{B4} \le -68$	$87 \le z'_{B4} \le 105$
$-83 \le x'_{B5} \le -63$	$-5 \le y'_{B5} \le 15$	$115 \le z'_{B5} \le 135$

Table 6.4 Solution obtained for -50mm≤z_N≤100mm in the objective function F_0 (variant **1**).

	i=1	*i*=2	*i*=3	*i*=4	*i*=5
x_{Ai}	190.436	482.605	401.068	422.000	344.310
y_{Ai}	87.591	-317.292	-210.635	-198.545	-3.447
z_{Ai}	238.816	236.036	289.298	410.077	430.258
x'_{Bi}	-33.737	-63.000	-31.195	-46.577	-67.136
y'_{Bi}	43.949	-36.344	-135.524	-78.843	-3.353
z'_{Bi}	-90.997	-129.034	-43.000	87.000	115.002
l_i	483.584	329.115	284.509	265.835	293.853

solution (Table 6.5) was obtained for z_{Nmin}=3000mm and z_{Nmax}=3100mm. This unusual domain of z_N facilitated obtaining a kinematic solution that ensures a higher location of the suspension roll center relative to the ground. According to [19], both rear and front suspension-roll centers should be as high as possible and at approximately the same height. However, limitations imposed to the wheel track alteration (mostly for the front wheels) restrict choosing a rear suspension that ensures a roll center located to high.

The two solutions obtained by synthesis noted **1** and **2**, were compared with an existing solution noted **0** also available in [13]. The wheel track variation:

$$\Delta S(z_N) = x_S(z_{N0}) - x_S(z_N) \tag{6.19}$$

Table 6.5 Solution obtained for 3000mm≤z_N≤3100mm in the objective function F_0 (variant **2**).

	i=1	*i*=2	*i*=3	*i*=4	*i*=5
x_{Ai}	203.760	493.038	390.597	422.066	348.449
y_{Ai}	111.186	-330.546	-197.256	-211.100	20.000
z_{Ai}	243.559	261.996	308.938	392.424	426.300
x'_{Bi}	-33.000	-69.814	-29.608	-35.783	-72.934
y'_{Bi}	37.798	-34.000	-133.477	-82.386	15.000
z'_{Bi}	-100.101	-148.828	-30.041	87.057	115.226
l_i	475.783	346.393	294.182	278.679	283.807

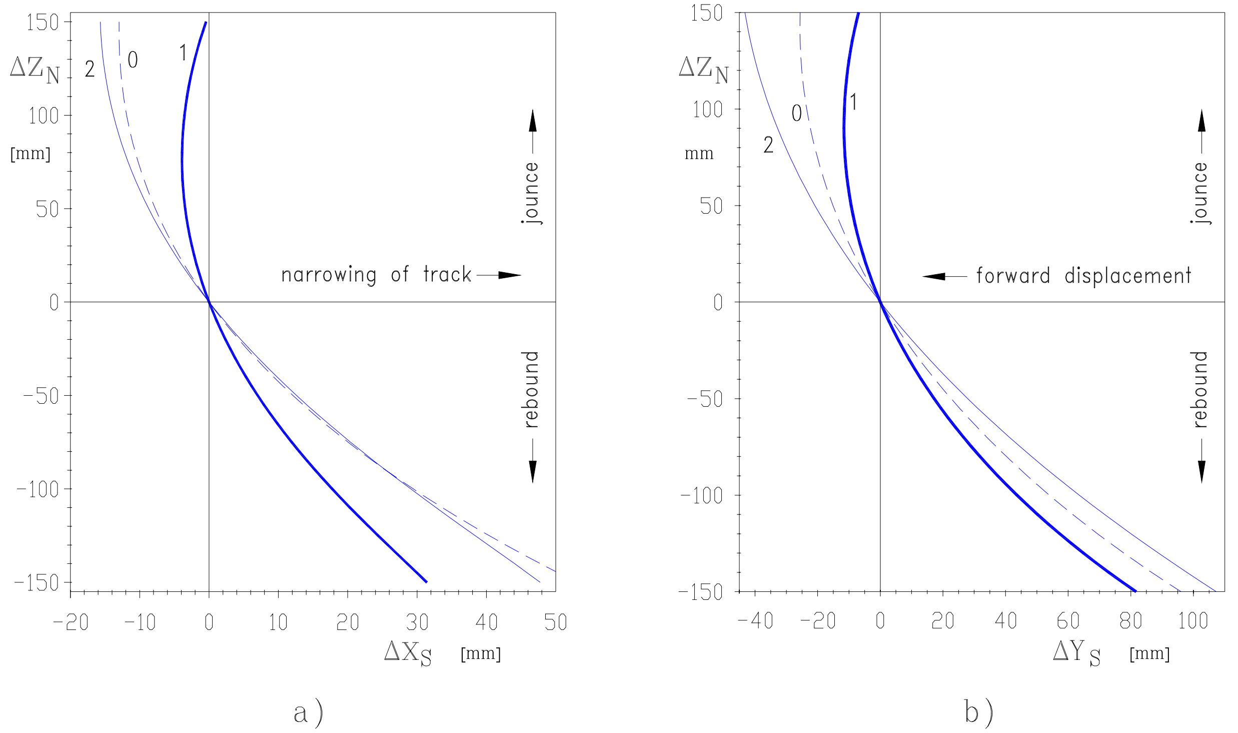


Fig.6.4 Wheel track alteration (**a**) and recessional wheel motion (**b**) during jounce and rebound for an initial solution **0**, and the two solutions obtained by synthesis, **1** and **2**.

is visibly improved in case of solution **1**. Also improved is the camber angle alteration Δδ as compared to the existing solution **0** (Fig. 6.4).

In Fig. 6.5 is given the diagram of the recessional motion of the wheel as the variation of y_S coordinate of the center path S.

$$\Delta Y(z_N) = y_S(z_{N0}) - y_S(z_N) \tag{6.20}$$

This parameter describes the fore/aft motion of the wheel during jounce and rebound. However, since it occurs along the direction of car travel, is has a smaller effect upon the car dynamics than the wheel track alteration.

The camber angle variation Δδ was determined as the projection of the angle between the axes Oz and Nz' on the vertical transverse plane (Fig. 6.6-a).

Similarly, the toe angle alteration Δφ shown in Fig. 6.7-b was determined as the angle between the axes Ox and Nx' projected on the horizontal plane. In this case,

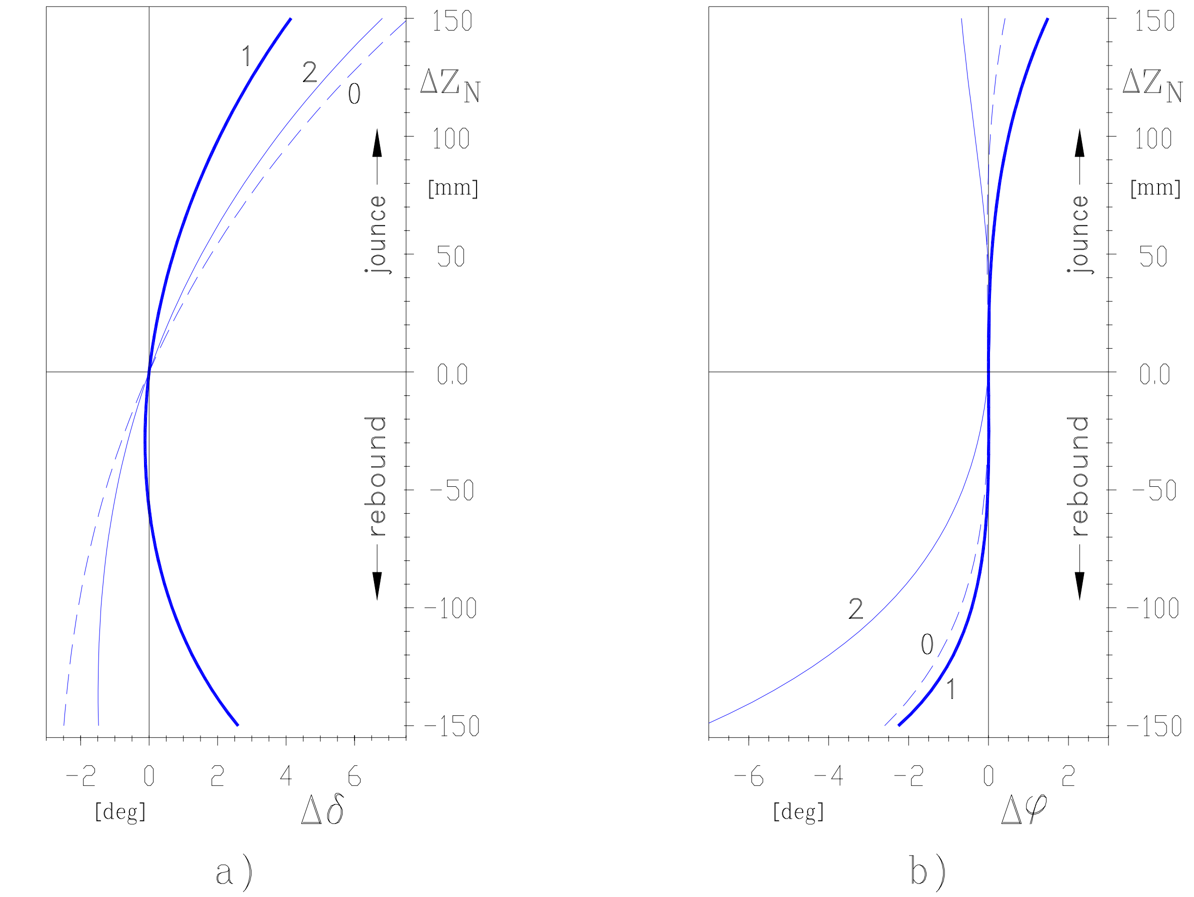


Fig. 6.5 Camber alteration $\Delta\delta(\Delta Z_N)$ (**a**) and toe angle alteration $\Delta\varphi(\Delta Z_N)$ (**b**) of the wheel relative to the chassis during jounce and rebound, for the same numerical variants in Fig. 6.4.

for $-150\text{mm} \le \Delta z_N \le 150\text{mm}$ the toe angle of solution **1** is slightly larger than that of the existing solution **0**, being however compensated by the under-steer effect of track widening during jounce.

For illustrative purposes, the diagrams of the magnitude of the angular velocity ω and angular acceleration ε of the wheel carrier have been plotted (Fig. 6.6) for $\dot{z}_N = 1.0\text{m/s}$ and $\ddot{z}_N = 0$ using equations (A3.1) and (A3.2) in Appendix 3.

The results of the kinematic analysis have been used in the 3D visualization and animation of the mechanism and also to check the possible collisions between the mobile elements. For this purpose an AutoLISP application named M3D.LSP (see Appendix 4) was developed for

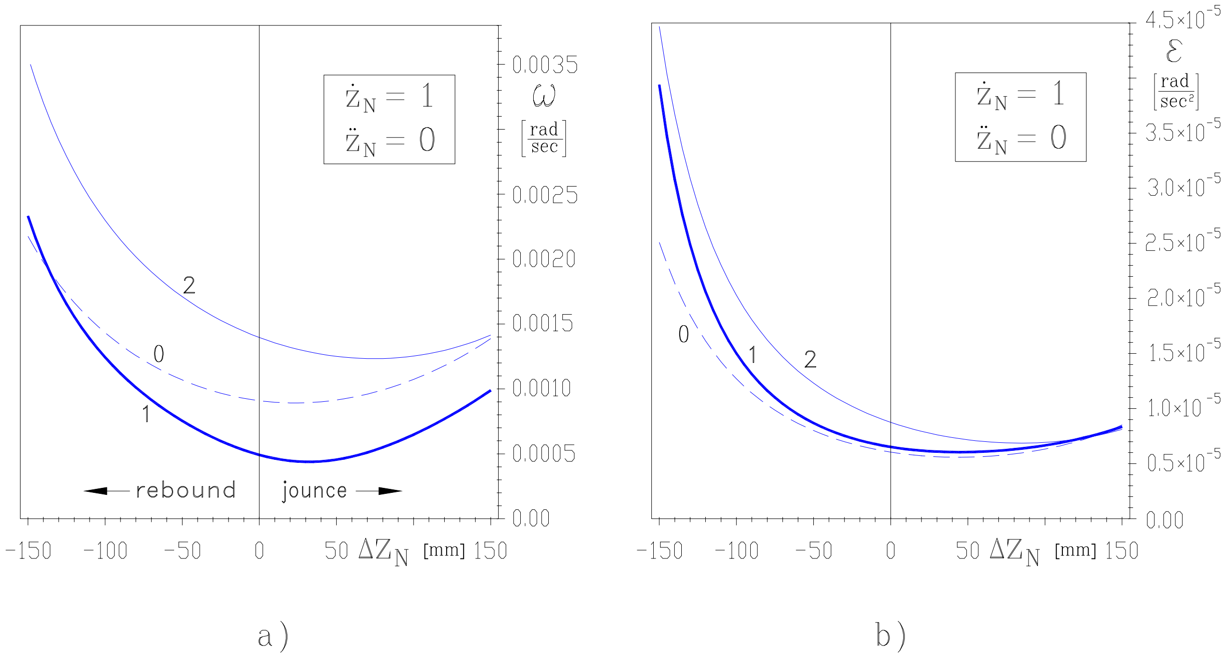


Fig. 6.6 Variation of the angular velocity ω (**a**) and angular acceleration ε (**b**) for $\dot{z}_N$ =1.0m/s and $\ddot{z}_N = 0$ for the same variants in Fig. 6.4.

automatically generating and animation inside AutoCAD^{TM} environment of 3D entities (e.g. cylinders, spheres, cones, tori) with their dimensions, positions and orientations read from an ASCII file.

Fig. 6.7, shows superimposed positions of the suspension mechanism solution **1**, corresponding to z_{N0} and z_{N0}±150mm, viewed from the rear (a) and from above (b) generated using M3D.LSP program. The validity of the kinematic analysis results was checked using MSC.visualNastran 4D commercial multibody simulation software; some results are available for comparison in Appendix 5.

The circle-point-surface and the center-point-surface in Fig. 6.3 were produced for solution **2**. They were generated as ruled surfaces of the momentary screw axis relative to the chassis (the circle-point-surface) and to the wheel carrier (the center-point-surface). The inclined position of the screw axis relative to car's longitudinal axis is due to the wheel-carrier

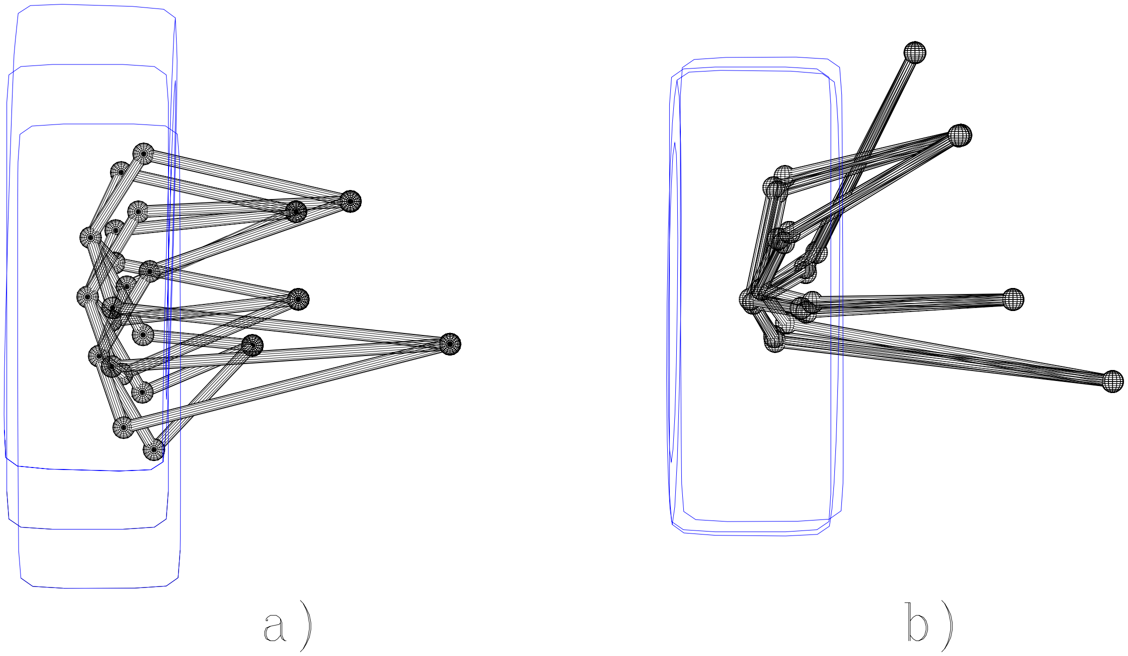


Fig. 6.7 Superimposed positions of the suspension mechanism solution **1**, corresponding to Δz_N=0 and Δz_N=±150mm, viewed from the rear (**a**) and from above (**b**).

rotation around its own axis, which for solution **2** corresponds to a maximum angle γ of 16.2° occurring for maximum rebound.

Finally, the plot in Fig. 6.8 of the alteration of the roll center height with Δz_N were produced. As compared to the existing solution, both variant **1** and **2** have a favorable smaller drop of the roll center under load. According to [19], in case of the real vehicle with compliant suspension, the roll center will be higher than for the simplified mechanism with rigid joints.

6.5. Conclusions

A complex multibody synthesis problem was presented, that of designing a five-link independent rear suspension system under the condition of ensuring a proper motion of the wheel-carrier. Also given were complete kinematic analysis equations that allow determining the wheel recession, wheel-track, toe angle, camber angle and roll-center height variation together with the linear and angular velocities and accelerations of the wheel-carrier of a given

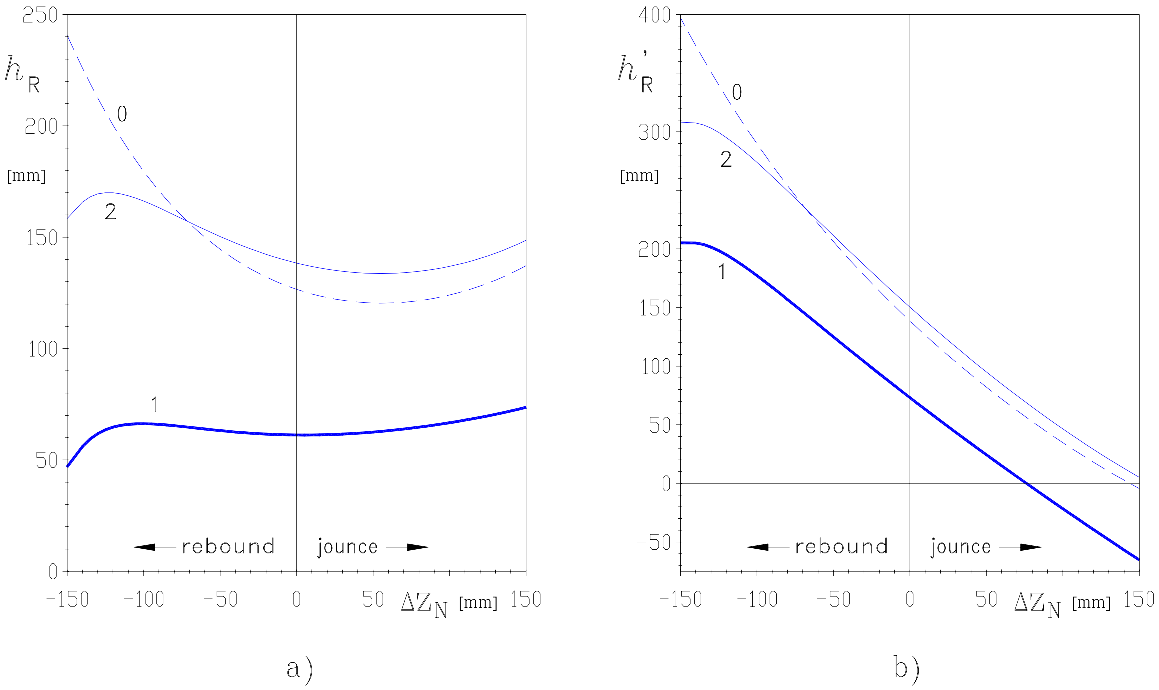


Fig. 6.8 Variation of the suspension roll-center height, measured relative to the car reference frame (a) and relative to the ground (b). In the reference position (Δz_N=0), h_R=138.6mm for variant **0**, h_R=73.2mm for variant **1** and h_R=150.4mm for variant **2**.

five-link suspension system. Two variants obtained by synthesis were analyzed and compared to an existing solution of a Mercedes 190 suspension system. Though the characteristics of the same mechanisms equipped with compliant joints will differ, the good behavior of the rigid joint mechanisms obtain by synthesis are likely to be preserved.

Both the synthesis and the analysis procedures advanced in this chapter can be extended to designing and simulating other suspension systems. For example the RSSR-SS double-wishbone suspension can be synthesized in the same manner. Multi-link suspensions used for front wheels of passenger cars (that have a second DOF needed for wheel steering) can also be synthesized following a similar approach, as well as the 5S-5S mechanisms used in guiding rigid axles. In this case however, the effect of joint elasticities must necessary be assessed using an advanced dynamic simulation software, since they have an essential contribution to the combined translation-rotation motion of the real axle.

6.6 References

[1] Lee, D.M.A. Pascoe, D.M. and El Maragy, W.H. (1993) “An Analysis of Multilink Independent Suspension Systems,” Intl J. of Vehicle Design, Vol. 14, p 44-58.

[2] Unkoo, L. and Byeongeui, A. (1993) “Method to Analyze the Imaginary Kingpin Axis in Multi-link Type Suspension Systems,” SAE Paper 930262, p. 11-26.

[3] Aviles, R. Ajuria, M.B. and Garcia de Jalon, J. (1985) “A Fairly General Method for Optimum Synthesis of Planar Mechanisms,” Mechanism and Machine Theory, Vol. 20, p. 321-328.

[4] Jimenez, J.M. Alvarez, G., Cardenal, J. and Cuadrado, J. (1997) “A Simple and General Method for Kinematic Synthesis of Spatial Mechanisms,” Mechanism and Machine Theory, Vol. 32, p. 323-341.

[5] Suh, C.H. (1989) “Synthesis and Analysis of Suspension Mechanisms with Use of Displacement Matrices,” SAE Paper 890098, p. 189-200.

[6] Sandor, G.N., Xu, L.J. and Yang, S.P. (1986) “Computer Aided Synthesis of Two-Closed Loop RSSR-SS Spatial Motion Generators with Branching and Sequence Constraints,” Mechanism and Machine Theory, Vol. 21, p. 345-350.

[7] Roth, B. (1967) “The Kinematics of Motion Through Finitely Separated Positions,” J. of Applied Mechanics, Vol. 34, p. 591-598.

[8] Roth, B. (1967) “Finite-Position Theory Applied to Mechanism Synthesis,” J. of Applied Mechanics, Vol. 34, p. 599-605.

[9] Sarkisyan, Y.L., Gupta, K.C. and Roth, B. (1973) “Spatial Least Square Approximation of Motion,” IFToMM Intl. Symposium on Linkages and CAD Methods, Bucharest, Vol. B, p. 512-521.

[10] Bastow, D. (1976) “Aspects of Car Rear Suspension,” Proc Inst Mech Engr, Vol. 190, p. 611-626.

[11] Raghavan, M. (1991) “Suspension Kinematic Structure for Passive Control of Vehicle Attitude,” Intl. J. of Vehicle Design, Vol. 12, p. 525-547.

[12] Mohamed, M.G, Attia, H.A. (1995) “Numerical Kinematic Analysis of the Motor-Vehicle Multilink Suspension System,” ASME DETC, Vol. 82, p. 883-889.

[13] Knapczyk, J. and Dzierzek, S. (1995) “Displacement and Force Analysis of Five-Rod Suspension With Flexible Joints,” J. of Mechanical Design, Vol. 117, p. 532-538.

[14] Suh, C.H. and Radcliffe, C.W. (1978) *Kinematics and Mechanisms Design*, Wiley.

[15] Hiller, M. (1993) “Five-Link Suspension” in *Multibody Computer Codes in Vehicle System Dynamics*, (W. Kortüm and R.S. Sharp, Eds.), Swets & Zeitlinger, Amsterdam, p. 254-262.

[16] Hiller, M. (1995) “Five-point Wheel Suspension,” in *Kinematics and Dynamics of Multi-Body Systems*, (Angeles J. and Kecskemethy, A., Eds.), Springer, p. 177-188.

[17] Gillespie, T.D. (1992) *Fundamentals of Vehicle Dynamics*, SAE, Warrendale, PA.

[18] Dixon, J.C. (1996) *Tires, Suspension and Handling*, SAE, Warrendale, PA.

[19] Reimpell J. Stoll, H. (1996) *Automotive Chassis. Engineering Principles*, SAE, Warrendale, PA.

CHAPTER 7. CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH

In this dissertation new optimum design tools were proposed and tested for solving several multibody design problems of engineering importance.

Firstly the constrained optimization problem solving capabilities of two Estimation of Distribution Algorithms (EDA) i.e. the Univariate Marginal Distribution Algorithm (UMDA) and Population Based Incremental Learning Algorithm (PBIL) were tested and some improvements proposed. It was found that forcing the standard deviation values to remain relatively large for an extended period of time during the search avoids premature convergence of these algorithms and increases their global optimum finding capabilities. It was also proposed that, within EDAs, the norm of the standard deviation vector can be used as stopping criteria, similarly to the gradient vector in derivative-based searching algorithms. Further investigations should be performed in testing the above concepts using additional benchmark optimization problems from literature, and also comparisons should be made with other Evolutionary Computation algorithms.

Secondly a technique of projecting hypersurfaces down to 3D and 2D space was proposed, which is particularly useful in inspecting the design space of objective functions of more than two variables. This technique requires repeated partial-global minimizations and/or maximizations of the function with respect to all but one or two of the design variables. Therefore the availability of powerful global optimization techniques are necessary requirements for its implementation. In addition to hyperobject visualization and objective function study, the use of this technique in solving multiple-objective optimization problems is also worth investigating - in such problems the designer must make a selection from a set of best solutions and therefore an overview of this set through graphical representations is always desirable.

Examples of applying the proposed visualization technique are presented in Chapters 4 and 5: In Chapter 4 the properties of the slider-crank and oscillating-slide actuators (widely used to convert the input motion of a linear actuator into the rotary motion of an output-member rocker) have been extensively investigated and new properties revealed. Further practical applications of the slider-crank mechanisms with large displacements of the output member, reported in this dissertation, should be sought.

Another example where the visualization of the design space of the optimization problem proved fruitful was the gear teeth number synthesis of an automatic planetary transmission of the Ravigneaux type. This highly constrained optimization problem in integer and discrete design variables was solved using an elitist Population Based Incremental Learning algorithm (PBIL) modified as proposed in Chapter 2. The same algorithm was used in performing the repeated searches needed for visualizing the design space of the problem through partial global minima plots. Such plots allow the designer to select the final numerical solution based on additional criteria, like using unified tools during the manufacturing process or limiting the overall size of the transmission. Further promising research could be the integration in the design problem of additional constraints and requirements, like maximizing the mechanical efficiency of the transmission and using the engine and torque converter characteristics for fuel consumption and dynamic output improvement of the vehicle.

In Chapter 6 a complex multibody-design problem was solved, that of synthesizing a five-link independent rear suspension system from the condition of ensuring minimum wheel recession, wheel-track, toe angle and camber variation during jounce and rebound of the wheel. Simplified analysis procedures were also proposed for verifying the above-listed parameters together with roll-center height and linear and angular velocities and accelerations of the wheel-carrier. The synthesis and analysis methods proposed in Chapter 6 can be applied to the design and simulation of double-wishbone suspensions and front wheel multi-link suspensions. Other similar parallel mechanisms used in materials handling applications, machine tools and robot-manipulators can also be designed and simulated using the proposed approach.

APPENDICES

Appendix 1. Computer program for generating the values required for plotting the "silhouettes" in Figs. 3.3-c and d and of the corresponding upper/lower-bound paths.

```
#include <stdio.h>
#include <math.h>
double MatlabPeaks(X1,X2) double X1,X2;{
  double T1,T2,T3;
  T1=pow(1-X1,2)*exp(-pow(X1,2)-pow(X2-1,2));
  T2=(X1/5-pow(X1,3)-pow(X2,5))
  *exp(-pow(X1,2)-pow(X2,2));
  T3=exp(-pow(X1+1,2)-pow(X2,2));
  return(3*T1-10*T2-1/3*T3);
}
void main(void){
   FILE *ASCIIfile;
   int     i1,n1,i2,n2;
   double  X1,X1min,X1max,X2,X2min,X2max;
   double  Z,Zmin,Zmax, X2Zmin,X2Zmax;
   ASCIIfile = fopen("MatlabPk.DAT", "w+");
   fprintf(ASCIIfile,"    X1       Zmin    X2(Zmin)");
   fprintf(ASCIIfile,"   Zmax   X2(Zmax)\n");
   n1=61;   X1min=-3.0; X1max=3.0;
   n2=1000; X2min=-3.5; X2max=3.5;

   for (i1= 1; i1 <= n1; i1++) {
    printf("Loop no. %d\n",i1);
    X1=X1min+(X1max-X1min)/(n1-1)*(i1-1);
    Zmin=1.0E100;  Zmax=-1.0E100;
    for (i2=1; i2 <= n2; i2++) {
      X2=X2min+(X2max-X2min)/(n2-1)*(i2-1);
      Z=MatlabPeaks(X1,X2);
      if (Zmin > Z) { Zmin=Z; X2Zmin=X2; }
      if (Zmax < Z) { Zmax=Z; X2Zmax=X2; }
    }
    fprintf(ASCIIfile,"%f %f %f %f %f\n"
    ,X1,Zmin,X2Zmin,Zmax,X2Zmax);
  }
  fclose(ASCIIfile);
```

Appendix 2. Summary of the optimum synthesis problem of a 3+1 Ravigneaux transmission.

Find integers $N_{1..6}$, number of planets p and module m_1 and m_3 that minimize the function:

$$f_1(...) = max\left| i_k - i_{0k} \right| \quad (k = \{1,2,\mathrm{R}\}) \tag{A2.1}$$

where:

$$i_1 = N_6/N_4 \quad \text{and} \quad i_{01} = 3.11 \tag{A2.2}$$

$$i_2 = \frac{N_6(N_1N_3 + N_2N_4)}{N_1N_3(N_6 - N_4)} \quad \text{and} \quad i_{02} = 1.84 \tag{A2.3}$$

$$i_\mathrm{R} = -(N_2N_6)/(N_1N_4) \quad \text{and} \quad i_{0\mathrm{R}} = -3.22 \tag{A2.4}$$

subject to the following constrains:

$$\begin{aligned} &N_{1,4} \geq N_{\mathrm{S\,min}} = 17 \\ &N_{2,3,5} \geq N_{\mathrm{P\,min}} = 14 \end{aligned} \tag{A2.5}$$

$$m_3(N_6 + 2.5) \leq D_{\max} \tag{A2.6}$$

$$m_1(N_1 + N_2) + m_1(N_2 + 2) \leq D_{\max} \tag{A2.7}$$

$$\left| m_1(N_1 + N_2) - m_3(N_6 - N_3) \right| \leq m_1 + m_3 \tag{A2.8}$$

$$(N_1 + N_2) \cdot \sin(\pi/p) - N_2 - 2 - \delta_{22} \geq 0 \tag{A2.9}$$

$$(N_6 - N_3) \cdot \sin(\pi/p) - N_3 - 2 - \delta_{33} \geq 0 \tag{A2.10}$$

$$(N_4 + N_5) \cdot \sin(\pi/p) - N_5 - 2 - \delta_{55} \geq 0 \tag{A2.11}$$

$$(\mathrm{N}_6-\mathrm{N}_3)^2+(\mathrm{N}_4+\mathrm{N}_5)^2-2(\mathrm{N}_6-\mathrm{N}_3)(\mathrm{N}_4+\mathrm{N}_5)\cos\left(\frac{2\pi}{p}-\beta\right)$$

$$\geq(\mathrm{N}_3+\mathrm{N}_5+2+\delta_{35})^2 \qquad \text{(A2.12)}$$

with

$$\beta=\arccos\frac{(\mathrm{N}_6-\mathrm{N}_3)^2+(\mathrm{N}_4+\mathrm{N}_5)^2-(\mathrm{N}_3+\mathrm{N}_5)^2}{2\cdot(\mathrm{N}_6-\mathrm{N}_3)(\mathrm{N}_4+\mathrm{N}_5)}$$

$$\mathrm{N}_6-2\mathrm{N}_3-\mathrm{N}_4-4-2\delta_{34}\geq 0 \qquad \text{(A2.13)}$$

$$\mathrm{N}_6-\mathrm{N}_4-2\mathrm{N}_5-4-2\delta_{56}\geq 0 \qquad \text{(A2.14)}$$

and equality constrains:

$$(\mathrm{N}_6-\mathrm{N}_4)/p=integer \qquad \text{(A2.15)}$$

and

$$Frac\left(\frac{1}{p}\left|\frac{\mathrm{N}_1}{\mathrm{N}_2}+\frac{\mathrm{N}_6}{\mathrm{N}_3}\right|\right)=\left|\frac{\mathrm{A}}{\mathrm{N}_2}\pm\frac{\mathrm{B}}{\mathrm{N}_3}\right| \qquad \text{(A2.16)}$$

where *Frac*(..) is the factional part of the expression in parentheses, A and B are integers within:

$$0\leq \mathrm{A}<\mathrm{N}_2/p \text{ and } 0\leq \mathrm{B}<\mathrm{N}_3/p \qquad \text{(A2.17)}$$

The lower limit of N_6 determined from equation (A13) and (A14) is:

$$\mathrm{N}_6\geq \mathrm{N}_{\mathrm{S}\min}+2\mathrm{N}_{\mathrm{P}\min}+2+2\cdot min(\delta_{34},\delta_{56}) \qquad \text{(A2.18)}$$

Additional upper side constraints can be obtained as follows:

From inequality (A6):

$$\mathrm{N}_6\leq \mathrm{D}_{\max}/m_{3\min}-2.5 \qquad \text{(A2.19)}$$

From inequality (A7):

$$\mathrm{N}_1\leq \mathrm{D}_{\max}/m_{1\min}-2\mathrm{N}_{\mathrm{P}\min}-2 \qquad \text{(A2.20)}$$

$$\mathrm{N}_2\leq(\mathrm{D}_{\max}/m_{1\min}-\mathrm{N}_{\mathrm{S}\min}-2)/2\,. \qquad \text{(A2.21)}$$

From inequalities (A13) and (A19):

$$N_3 \leq \left(D_{max}/m_{3\min} - N_{S\min} - 6.5 - 2\delta_{34}\right)/2 \tag{A2.22}$$

From inequality (A14) and (A19):

$$N_4 \leq D_{max}/m_{3\min} - 2N_{P\min} - 6.5 - 2\delta_{56} \tag{A2.23}$$

$$N_5 \leq \left(D_{max}/m_{3\min} - N_{S\min} - 6.5 - 2\delta_{56}\right)/2\,. \tag{A2.24}$$

In the above equations the maximum admissible outer diameter is D_{max}=220 mm, the number of identical planets p can be 3, 4 or 5, while module m_1 and m_3 can have the following discrete values: 1.75, 2.0, 2.25, 2.5, 2.75, or 3.0 mm. The relative clearance between adjacent wheels δ_{22}, δ_{33}, δ_{55}, δ_{35}, δ_{34} and δ_{56} were considered all equal 0.5.

Appendix 3. Velocity and acceleration analysis equations of the wheel carrier of a five-link suspension.

Considering two distinct ball-joint centers B$_j$ and B$_k$ (Fig. 6.1), the components of the angular velocity of the wheel carrier can be calculated with the equations:

$$
\begin{aligned}
\omega_{\mathrm{x}} &= \frac{\mathrm{P1}\cdot\Delta \mathrm{x}_j\cdot\Delta \mathrm{x}_k+\mathrm{P2}\cdot\Delta \mathrm{y}_j\cdot\Delta \mathrm{x}_k+\mathrm{P3}\cdot\Delta \mathrm{x}_j\cdot\Delta \mathrm{z}_j}{\Delta \mathrm{x}_j\cdot\Delta \mathrm{z}_j\cdot\Delta \mathrm{y}_k-\Delta \mathrm{y}_j\cdot\Delta \mathrm{z}_j\cdot\Delta \mathrm{x}_k}\\
\omega_{\mathrm{y}} &= (\Delta \mathrm{y}_k\cdot\omega_x-\mathrm{P3})/\Delta \mathrm{x}_k\\
\omega_{\mathrm{z}} &= (\Delta \mathrm{z}_j\cdot\omega_{\mathrm{y}}-\mathrm{P1})/\Delta \mathrm{y}_j
\end{aligned}
\tag{A3.1}
$$

The components of the angular acceleration of the wheel carrier are given by:

$$
\begin{aligned}
\varepsilon_{\mathrm{x}} &= \frac{\mathrm{Q1}\cdot\Delta \mathrm{x}_j\cdot\Delta \mathrm{x}_k+\mathrm{Q2}\cdot\Delta \mathrm{y}_j\cdot\Delta \mathrm{x}_k+\mathrm{Q3}\cdot\Delta \mathrm{x}_j\cdot\Delta \mathrm{z}_j+}{\Delta \mathrm{x}_j\cdot\Delta \mathrm{z}_j\cdot\Delta \mathrm{y}_k-\Delta \mathrm{y}_j\cdot\Delta \mathrm{z}_j\cdot\Delta \mathrm{x}_k}\\
\varepsilon_{\mathrm{y}} &= (\Delta \mathrm{y}_k\cdot\varepsilon_{\mathrm{x}}-\mathrm{Q3})/\Delta \mathrm{x}_k\\
\varepsilon_{\mathrm{z}} &= (\Delta \mathrm{z}_j\cdot\varepsilon_{\mathrm{y}}-\mathrm{Q1})/\Delta \mathrm{y}_j
\end{aligned}
\tag{A3.2}
$$

with

$$
\begin{aligned}
\Delta_{\mathrm{x}j} &= \mathrm{x}_{\mathrm{B}j}-\mathrm{x}_{\mathrm{N}}\,, \quad \Delta_{\mathrm{x}k}=\mathrm{x}_{\mathrm{B}k}-\mathrm{x}_{\mathrm{N}}\\
\Delta_{\mathrm{y}j} &= \mathrm{y}_{\mathrm{B}j}-\mathrm{y}_{\mathrm{N}}\,, \quad \Delta_{\mathrm{y}k}=\mathrm{y}_{\mathrm{B}k}-\mathrm{y}_{\mathrm{N}}\\
\Delta_{\mathrm{z}j} &= \mathrm{z}_{\mathrm{B}j}-\mathrm{z}_{\mathrm{N}}\,, \quad \Delta_{\mathrm{z}k}=\mathrm{z}_{\mathrm{B}k}-\mathrm{z}_{\mathrm{N}}
\end{aligned}
\tag{A3.3}
$$

$$
\mathrm{P1}=\dot{\mathrm{x}}_{\mathrm{B}j}-\dot{\mathrm{x}}_{\mathrm{N}}\,, \quad \mathrm{P2}=\dot{\mathrm{y}}_{\mathrm{B}j}-\dot{\mathrm{y}}_{\mathrm{N}}\,, \quad \mathrm{P3}=\dot{\mathrm{z}}_{\mathrm{B}k}-\dot{\mathrm{z}}_{\mathrm{N}}\,,
\tag{A3.4}
$$

$$
\begin{aligned}
\mathrm{Q1} &= \left(\omega_{\mathrm{y}}^2+\omega_{\mathrm{z}}^2\right)\cdot\Delta_{\mathrm{x}j}-\omega_{\mathrm{x}}\left(\omega_{\mathrm{y}}\cdot\Delta_{\mathrm{y}j}+\omega_{\mathrm{z}}\cdot\Delta_{\mathrm{z}j}\right)+\ddot{\mathrm{x}}_{\mathrm{B}j}-\ddot{\mathrm{x}}_{\mathrm{N}}\\
\mathrm{Q2} &= \left(\omega_{\mathrm{x}}^2+\omega_{\mathrm{z}}^2\right)\cdot\Delta_{\mathrm{y}j}-\omega_{\mathrm{y}}\left(\omega_{\mathrm{x}}\cdot\Delta_{\mathrm{x}j}+\omega_{\mathrm{z}}\cdot\Delta_{\mathrm{z}j}\right)+\ddot{\mathrm{y}}_{\mathrm{B}j}-\ddot{\mathrm{y}}_{\mathrm{N}}\ .\\
\mathrm{Q3} &= \left(\omega_{\mathrm{x}}^2+\omega_{\mathrm{y}}^2\right)\cdot\Delta_{\mathrm{z}k}-\omega_{\mathrm{z}}\left(\omega_{\mathrm{x}}\cdot\Delta_{\mathrm{x}k}+\omega_{\mathrm{y}}\cdot\Delta_{\mathrm{y}k}\right)+\ddot{\mathrm{z}}_{\mathrm{B}k}-\ddot{\mathrm{z}}_{\mathrm{N}}
\end{aligned}
\tag{A3.5}
$$

The parametric equations of the screw axis of the wheel carrier can be determined with the following equations:

$$
\begin{aligned}
x(y) &= \left[T_1 \cdot \omega_y \cdot \omega_z + T_2(\omega_x^2 + \omega_y^2)\right] \Big/ \left(\omega_x \cdot \omega_y \cdot \omega^2\right) \\
z(y) &= \left(-T_1 \cdot \omega_y + T_2 \cdot \omega_z\right) \Big/ \omega_y \cdot \omega^2
\end{aligned}
\tag{A3.6}
$$

where:

$$
\begin{aligned}
T_1 &= \omega_y \cdot \dot{x}_N - \omega_x \cdot \dot{y}_N + \omega_x \cdot \omega_z \cdot x_N + \omega_y \cdot \omega_z (y_N - y_C) - z_N (\omega_x^2 + \omega_y^2) \\
T_2 &= \omega_x \cdot \dot{z}_N - \omega_z \cdot \dot{x}_N + \omega_y \cdot \omega_z \cdot z_N + \omega_x \cdot \omega_y \cdot x_N + (y_C - y_N)(\omega_x^2 + \omega_z^2)
\end{aligned}
\tag{A3.7}
$$

Appendix 4. AutoLisp program for plotting and animating inside AutoCAD lines, cylinders, spheres, tori, cones, spirals and AutoCAD blocks using data read from ASCII files.

```
;-------------------------------------------------------------------------
; M3d.LSP
;
; AutoLisp program for plotting and animating lines, cylinders, spheres,
; tori, cones, spirals and AutoCAD blocks using data read from ASCII files.
: This program is provided "as is" without express or implied warranty.
; All implied warranties of fitness for any particular purpose and of
; merchantability are hereby disclaimed.   (c) P.A. Simionescu  2004
;
;-------------------------------------------------------------------------
; Input .M3D file line for creating a new layer and making it current:
;
; (New_Layer) comment
;
;-------------------------------------------------------------------------
;Input .M3D file line for setting current color:
;
; (CL "RED"    )  comment
; (CL "YELLOW" )  comment
; (CL "GREEN"  )  comment
; (CL "CYAN"   )  comment
; (CL "BLUE"   )  comment
; (CL "MAGENTA")  comment
; (CL "WHITE"  )  comment
;
;-------------------------------------------------------------------------
;Input .M3D file line for drawing a line from #1 to #2:
;
; (x1  y1  z1  x2  y2  z2) comment
;with:
;  x1  y1  z1 : WCS coordinates of end #1
;  x2  y2  z2 : WCS coordinates of end #2
;
;-------------------------------------------------------------------------
;Input .M3D file line for drawing a cylinder with a sphere at each end:
;
; (x1  y1  z1  x2  y2  z2  r12  rS1  rS2) comment
;with:
;  x1  y1  z1 : WCS coordinates of end #1
;  x2  y2  z2 : WCS coordinates of end #2
;  r12        : cylinder radius
;  rS1        : sphere radius at end #1
;  rS2        : sphere radius at end #2
;NOTE:
; if (r12 = 0) draws a line from #1 to #2 only
; if (rS1 = nil) draws no sphere at #1 and the end is transparent
; if (rS2 = nil) draws no sphere at #2 and the end is transparent
; if (rS1 =   0) draws no sphere at #1 and the end is opaque
; if (rS2 =   0) draws no sphere at #2 and the end is opaque
;
;-------------------------------------------------------------------------
;Input .M3D file line for drawing a cone [fulcrum] with a sphere at each end:
;
; (CO x1  y1  z1  x2  y2  z2  r1  r2  rS1  rS2) comment
;with:
;  x1  y1  z1 : WCS coordinates of end #1
;  x2  y2  z2 : WCS coordinates of end #2
;  r1         : base radius at #1 (cannot be zero)
;  r2         : base radius at #2 (can be zero)
;  rS1        : sphere radius at end #1
;  rS2        : sphere radius at end #2
```

```
;NOTE:
; if (rS1 = nil) draws no sphere at #1 and the end is transparent
; if (rS2 = nil) draws no sphere at #2 and the end is transparent
; if (rS1 =   0) draws no sphere at #1 and the end is opaque
; if (rS2 =   0) draws no sphere at #2 and the end is opaque
;
;--------------------------------------------------------------------------
;Input .M3D file line for drawing a torus:
;
; (TO x1  y1  z1  x2  y2  z2  r0  rC) comment
;with:
;  x1  y1  z1 : WCS coordinates of the center of the torus
;  x2  y2  z2 : WCS coordinates of a 2nd point on the axis of the torus
;  r0         : centroidal radius (cannot be zero)
;  rC         : radius of the path circle (cannot be zero)
;
;--------------------------------------------------------------------------
;Input .M3D file line for drawing a sphere:
;
; (SP x  y  z  r) comment
;with:
;  x  y  z : WCS coordinates of the center of the sphere
;  r       : radius (cannot be zero)
;
;--------------------------------------------------------------------------
;Input .M3D file line for drawing a cylindrical helix:
;
; (CS x1 y1 z1 x2 y2 z2 r n)  comment
;with:
;  x1  y1  z1 : WCS coordinates of end #1
;  x2  y2  z2 : WCS coordinates of end #2
;  r          : helix radius
;  n          : number of threads
;
;--------------------------------------------------------------------------
;Input .M3D file line for inserting an existing block named "BlkNme":
;
; (BK "BlkNme" x1 y1 z1 x2 y2 z2 x3 y3 z3) comment
;with:
;  BlkNme      : BlkNme must exist in the database of the current .DWG file
;  x1  y1  z1  : WCS origin of the reference frame (LRF) attached to the block
;  x2  y2  z2  : WCS coordinates of a point on the positive OX axis of the LRF
;  x3  y3  z3  : WCS coordinates of a point on the positive OY axis of the LRF
;
;--------------------------------------------------------------------------
;Input .M3D file line for inserting a text:
;
; (TX "MyText" x1 y1 z1 Hgt Rot) comment
;with:
;  MyText     : the text to be printed
;  x1  y1  z1 : WCS insertion point of the text
;  Hgt        : text height
;  Rot        : orientation angle of the text
;
;==========================================================================
;==========================================================================

(defun My_Torus (P1 P2 r1 r2 / e1 e2)
   (command "_.LINE" P1 P2 "")
   (setq e1 (entlast))
   (command "_.UCS" "_ZA" P1 P2)
   (command "_.UCS" "_Y" "90")
   (command "_.CIRCLE" (list 0.0 r1 0.0) r2); generating circle
   (setq e2 (entlast))
   (command "_.revsurf" e2 e1 "" "")
   (entdel e1); delete axis
   (entdel e2); delete generating cyrcle
   (command "_.UCS" "_P" )
   (command "_.UCS" "_P" )
); end My_Torus

;--------------------------------------------------------------------------
```

```
(defun My_cone (P1 P2 h r1 r2 rS1 rS2 / oldelev e1 e2)
   (setq oldelev (getvar "ELEVATION"))
   (setvar "ELEVATION" 0)

   (if (=  r2 0.0) (command "_.POINT" P2)); top point
   (setq e2 (entlast))

   (command "_.UCS" "_ZA" P1 P2)
   (command "_.CIRCLE" "0,0" r1); base circle
   (setq e1 (entlast))
   (setvar "ELEVATION" h)
   (if (/= r2 0.0)
      (progn
         (command "_.CIRCLE" "0,0" r2); top circle
         (setq e2 (entlast))
      ); end progn
   ); end if

   (command "_.RULESURF" (list e1 P1) (list e2 P2)); draw cone
   (setvar "ELEVATION" oldelev)
   (command "_.UCS" "_PREV")

   (if (= rS1 nil) (entdel e1)); delete base circle
   (if (= rS2 nil) (entdel e2)); delete top circle
); end My_cone

;--------------------------------------------------------------------------
(defun My_cylinder (P1 P2 h r12 rS1 rS2 / oldelev e1 e2)
   (My_cone P1 P2 h r12 r12 rS1 rS2)
); end My_cylinder

;--------------------------------------------------------------------------
(defun My_sphere (Pt rs / e1 e2 ax ax1)
   (setq ax (list (car Pt) (+ (cadr Pt) rs) (caddr Pt)))
   (setq ax1 (list (car Pt) (- (cadr Pt) rs) (caddr Pt)))
   (command "_.LINE" ax ax1 "");Draw axis of revolution
   (setq e1 (entlast))
   (command "_.ARC" ax "_e" ax1 "_a" "180.0"); draw path curve
   (setq e2 (entlast))
   (command "_.REVSURF" (list e2 ax) (list e1 Pt) "" ""); draw sphere
   (entdel e1)
   (entdel e2)
); end My_sphere

;--------------------------------------------------------------------------
(defun 3Dspiral (P1 P2 r0 nC)
   (setq nLC 30); number of segments on one helix

   (setq h (distance P1 P2))
   (command "_.UCS" "_ZA" P1 P2)
   (if (= h 0.0) (command "_.CIRCLE" P1 r0 ""))
   (if (/= h 0.0)
      (progn
         (setq Ainc (/ (* pi 2) nLC))
         (setq Hinc (/ (/ h nC) nLC))
         (setq Ang 0.0)
         (setq j 0.0)
         (command "_.3DPOLY")
         (setq Pt0 (list  0.0  0.0  0.0))  (command Pt0)
         (setq Pt  (list   r0  0.0  0.0))  (command Pt)
         (repeat nC
            (repeat nLC
               (setq j (+ j 1))
               (setq Pt (polar Pt0 (setq Ang (+ Ang Ainc)) r0))
               (setq Pt (list (car Pt) (cadr Pt) (* Hinc j)))
               (command Pt)
            ); end repeat nC
         ); end repeat nLC
         (setq Pt (list 0.0  0.0  (caddr Pt)))  (command Pt "")
      ); end progn
   ); end if
   (command "_.UCS" "_PREV")
); end 3Dspiral
```

```
;==========================================================================
;==========================================================================
(defun C:M3D()
  (setq OldCMDECHO  (getvar "CMDECHO" ))
  (setq OldBLIPMODE (getvar "BLIPMODE"))
  (setvar "CMDECHO"   0)
  (setvar "BLIPMODE"  0)
  (setvar "SURFTAB1" 16)
  (setvar "SURFTAB2" 16)
  (command "_.UCSICON" "_OFF" "")
  ;- - - - - - - - - - - - - - - - - - - - - - - - - - -
  (setq Path "C:/a/TpFILES/")
  (setq M3DfileNm (getfiled "Select a M3D input file:" Path "M3D" 8)); accept the
name of .M3D input-file
  (setq M3Dfile (open M3DfileNm "r")); open for input an M3D file

  (setq BK "BLOCK"     )
  (setq CL "COLOR"     )
  (setq CO "CONE"      )
  (setq CS "CoilSpring")
  (setq LN "LINE"      )
  (setq SP "SPHERE"    )
  (setq TO "TORUS"     )
  (setq TX "OutTEXT"   )
  (setq ROW (read-line M3Dfile))
  (while (/=  ROW nil)
    (setq ROWlst (read ROW))
    ;- - - - - - - - - - - - - - - - - - - - - - - - - - -
    ; Create new layer and make it current
    ;- - - - - - - - - - - - - - - - - - - - - - - - - - -
    (if (= (length ROWlst) 1)
      (progn
        (setq LayNr (nth 0 ROWlst))
        (command "_.LAYER" "_MAKE" LayNr "")
      ); end progn
    ); end if new layer
    ;- - - - - - - - - - - - - - - - - - - - - - - - - - -
    ; Set current color
    ;- - - - - - - - - - - - - - - - - - - - - - - - - - -
    (if (AND (= (eval (nth 0 ROWlst)) "COLOR") (= (length ROWlst) 2))
      (command "_.COLOR" (nth 1 ROWlst) "")
    ); end if color

    ;- - - - - - - - - - - - - - - - - - - - - - - - - - -
    ; Write text
    ;- - - - - - - - - - - - - - - - - - - - - - - - - - -
    (if (AND (= (eval (nth 0 ROWlst)) "OutTEXT") (= (length ROWlst) 7))
      (progn
        (setq Txt (nth 1 ROWlst)
               x1  (nth 2 ROWlst)
               y1  (nth 3 ROWlst)
               z1  (nth 4 ROWlst)
               Hgt (nth 5 ROWlst)
               Rot (nth 6 ROWlst)
               P1  (list x1 y1 z1)
        );end setq
        (command "TEXT" P1 Hgt Rot Txt "")
      ); end progn
    ); end if text
    ;- - - - - - - - - - - - - - - - - - - - - - - - - - -
    ; Draw cylinder [with spheres at ends]
    ;- - - - - - - - - - - - - - - - - - - - - - - - - - -
    (if (AND (/= (eval (nth 0 ROWlst)) "BLOCK")
      (/= (eval (nth 0 ROWlst)) "CONE")
      (/= (eval (nth 0 ROWlst)) "LINE")
      (/= (eval (nth 0 ROWlst)) "TORUS")
      (/= (eval (nth 0 ROWlst)) "CoilSpring")
      (> (length ROWlst) 5) (< (length ROWlst) 10))
      (progn
        (setq x1  (nth 0 ROWlst)
              y1  (nth 1 ROWlst)
              z1  (nth 2 ROWlst)
              x2  (nth 3 ROWlst)
```

```
          y2  (nth 4 ROWlst)
          z2  (nth 5 ROWlst)
          r12 (nth 6 ROWlst);cylinder radius
          rS1 (nth 7 ROWlst);sphere radius at #1
          rS2 (nth 8 ROWlst);sphere radius at #2
          P1  (list x1 y1 z1)
          P2  (list x2 y2 z2)
    ); end setq
    (if (= r12 nil) (setq r12 0.0))
    (if (= r12 0.0) (command "_.LINE" P1 P2 ""))

    (if (/= r12 0.0)
      (progn
        (setq h (distance P1 P2))
        (if (/= h 0.0)
          (progn
            (setq r12 (abs (float r12)))
            (My_cylinder P1 P2 h r12 rS1 rS2)
          ); end progn
        ); end if
      ); end progn
    ); end if
    (if (= rS1 nil) (setq rS1 0.0))
    (if (= rS2 nil) (setq rS2 0.0))
    (setq rS1 (float rS1) rS2 (float rS2))
    (if (/= rS1 0.0) (My_sphere P1 rS1))
    (if (/= rS2 0.0) (My_sphere P2 rS2))
  ); end progn
); end if cylinder

;- - - - - - - - - - - - - - - - - - - - - - - - - - -
; Draw helix
;- - - - - - - - - - - - - - - - - - - - - - - - - - -
(if (AND (= (eval (nth 0 ROWlst)) "CoilSpring") (= (length ROWlst) 9))
  (progn
    (setq x1  (nth  1 ROWlst)
          y1  (nth  2 ROWlst)
          z1  (nth  3 ROWlst)
          x2  (nth  4 ROWlst)
          y2  (nth  5 ROWlst)
          z2  (nth  6 ROWlst)
          r0  (abs (nth  7 ROWlst)) ;coil radius
          nC  (fix (nth  8 ROWlst)) ;number of coils
          P1  (list x1 y1 z1)
          P2  (list x2 y2 z2)
    );end setq
    (if (=  r0 0.0) (command "_.LINE" P1 P2 ""));zero diameter arch
    (if (/= r0 0.0) (3Dspiral P1 P2 r0 nC))
  ); end progn
); end if helix

;- - - - - - - - - - - - - - - - - - - - - - - - - - -
; Draw torus
;- - - - - - - - - - - - - - - - - - - - - - - - - - -
(if (AND (= (eval (nth 0 ROWlst)) "TORUS") (= (length ROWlst) 9))
  (progn
    (setq x1 (nth  1 ROWlst)
          y1 (nth  2 ROWlst)
          z1 (nth  3 ROWlst)
          x2 (nth  4 ROWlst)
          y2 (nth  5 ROWlst)
          z2 (nth  6 ROWlst)
          r0 (abs (nth 7 ROWlst))
          rC (abs (nth 8 ROWlst))
          P1 (list x1 y1 z1)
          P2 (list x2 y2 z2)
    ); end setq
    (if (/= (distance P1 P2) 0.0) (My_Torus P1 P2 r0 rC))
  ); end progn
); end if torus

;- - - - - - - - - - - - - - - - - - - - - - - - - - -
; Draw cone or cone fulcrum [with spheres at ends]
```

```
;- - - - - - - - - - - - - - - - - - - - - - - - - - -
(if (AND (= (eval (nth 0 ROWlst)) "CONE")
  (> (length ROWlst) 8)
  (< (length ROWlst) 12)
  ); end AND
  (progn
    (setq x1  (nth  1 ROWlst)
          y1  (nth  2 ROWlst)
          z1  (nth  3 ROWlst)
          x2  (nth  4 ROWlst)
          y2  (nth  5 ROWlst)
          z2  (nth  6 ROWlst)
          r1  (nth  7 ROWlst);radius of the circle at #1
          r2  (nth  8 ROWlst);radius of the circle at #2
          rS1 (nth  9 ROWlst);radius of the sphere at #1
          rS2 (nth 10 ROWlst);radius of the sphere at #2
          P1  (list x1 y1 z1)
          P2  (list x2 y2 z2)
    ); end setq
    (if (= r1 0.0) (command "_.LINE" P1 P2 "")); base circle cannot be 0 radius
    (if (/= r1 0.0)
      (progn
        (setq h (distance P1 P2))
        (if (/= h 0.0)
          (progn
            (setq r1 (abs (float r1)))
            (setq r2 (abs (float r2)))
            (My_cone P1 P2 h r1 r2 rS1 rS2)
          ); end progn
        ); end if
      ); progn
    ); end if

    (if (= rS1 nil) (setq rS1 0.0))
    (if (= rS2 nil) (setq rS2 0.0))
    (setq rS1 (float rS1) rS2 (float rS2))
    (if (/= rS1 0.0) (My_sphere P1 rS1))
    (if (/= rS2 0.0) (My_sphere P2 rS2))
  ); end progn
); end if cone

;- - - - - - - - - - - - - - - - - - - - - - - - - - -
; Draw sphere
;- - - - - - - - - - - - - - - - - - - - - - - - - - -
(if (AND (= (eval (nth 0 ROWlst)) "SPHERE") (= (length ROWlst) 5))
  (progn
    (setq x (nth  1 ROWlst)
          y (nth  2 ROWlst)
          z (nth  3 ROWlst)
          r (abs (nth 4 ROWlst))
          P1 (list x y z)
    ); end setq
    (if (/= r 0.0) (My_sphere P1 r))
  ); end progn
); end if sphere

;- - - - - - - - - - - - - - - - - - - - - - - - - - -
; Insert block
;- - - - - - - - - - - - - - - - - - - - - - - - - - -
(if (AND (= (eval (nth 0 ROWlst)) "BLOCK") (= (length ROWlst) 11))
  (progn
    (setq BlockName (nth 1 ROWlst)
          x1 (nth  2 ROWlst)
          y1 (nth  3 ROWlst)
          z1 (nth  4 ROWlst)
          x2 (nth  5 ROWlst)
          y2 (nth  6 ROWlst)
          z2 (nth  7 ROWlst)
          x3 (nth  8 ROWlst)
          y3 (nth  9 ROWlst)
          z3 (nth 10 ROWlst)
          P1 (list x1 y1 z1)
          P2 (list x2 y2 z2)
```

```
              P3 (list x3 y3 z3)
        );end setq
        (command "_.UCS" "_3" P1 P2 P3)
        (command "_.INSERT" BlockName "0,0,0" "" "" "")
        (command "_.UCS" "_PREV")
      ); end progn
    ); end if block
    ;- - - - - - - - - - - - - - - - - - - - - - - - - - -

    (setq ROW (read-line M3Dfile))
  ); end while
  (close M3Dfile)
  (command "_.COLOR" "WHITE" "")
  (command "_.LAYER" "_SET" "0" "")
  (command "_.VPOINT" "1,1,1" "")
  (command "_.ZOOM" "_EXTENTS" "")

  (alert "Choose a proper vewpoint\n\nand then type 'motion'!")

); end defun M3D

;=================================================================

(defun C:MOTION( )

  (setq SCRfileNm (getfiled "Output file:" "" "SCR" 1)); accept the name of the
output .SCR file
  (setq SCRfile (open SCRfileNm "w")); open .SCR file for output

  (setq S SCRfileNm)

  (setq S (substr S (- (strlen S) 12) 9)); extract (roughly) the name of .SCR file

  ; if the name of the .SCR file is shorter than 8 char it must be cleaned off:
  ; converted S to a list
  (setq S (list
            (substr S  1 1) (substr S  2 1) (substr S  3 1)
            (substr S  4 1) (substr S  5 1) (substr S  6 1)
            (substr S  7 1) (substr S  8 1) (substr S  9 1)
          ); end list
  ): end setq

  (setq AuxS S)
  (while (/= AuxS nil); delete from front until "\\" encountered
    (setq S AuxS)
    (while (AND (/= AuxS nil)(/= (CAR AuxS) "\\"))
      (setq AuxS (CDR AuxS)); delete 1st char of S until "\" encountered
    ); end while
    (setq AuxS (CDR AuxS)); delete the second "\"
  ); end while
  (setq AuxS S)

  (setq Path (substr SCRfileNm 1 (- (strlen SCRfileNm) (length S) 4))); update path

  (setq S "")
  (setq i 0)
  (while (<= i 4); i less equal 5 means that no more than 999 frames can be
generated
    (if (/= (nth i AuxS) nil) (setq S (strcat S (nth i AuxS))))
    (setq i (+ i 1))
  ); while

;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
;extract the total number of layers LayNr named 1,2,3...999
  (setq LayNr (atoi (cdr (assoc '8 (entget (entlast))))))

  (setq i 1); prepare layer for hiding
  (command "_.LAYER")
  (while (<= i LayNr)
    (command "_OFF" i "_FREEZE" i)
    (setq i (+ i 1))
  ); end while
  (if (> i  999) (setq i 999)); ignore all frames over 999
  (command "" "_.REGEN" "")
  ;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  (setq i 1)
  (while (<= i LayNr)
    (command "_.LAYER" "_ON" i "_THAW" i "")
    (command "_.HIDE" "")

    ; generate ith .BMP file names as NNNNN001.BMP, NNNNN002.BMP ... NNNNNi.BMP
    ; generate ith .SLD file names as NNNNN001.SLD, NNNNN002.SLD ... NNNNNi.SLD
    ; with NNN the first 5 characters in the name of the output .SCR file

    (setq SLDfileNm S)
    (setq BMPfileNm S)

    (if (< i 10) (setq SLDfileNm (strcat SLDfileNm  "00") BMPfileNm (strcat
BMPfileNm  "00")))
    (if (AND (> i  9) (< i  100)) (setq SLDfileNm (strcat SLDfileNm "0") BMPfileNm
(strcat BMPfileNm "0")))

    (setq SLDfileNm (strcat SLDfileNm (itoa i)))
    (setq BMPfileNm (strcat BMPfileNm (itoa i)))

    (write-line (strcat "vslide *" SLDfileNm) SCRfile); write a preload command line
to .SCR file
    (write-line "delay 1" SCRfile); write delay command line to .SCR file
    (write-line "vslide" SCRfile); write line to .SCR file

    (setq SLDfileNm (strcat Path SLDfileNm))
    (setq BMPfileNm (strcat Path BMPfileNm))

    (command "_.MSLIDE" SLDfileNm ""); output SLD file # i
    (command "_.BMPOUT" BMPfileNm "_ALL" ""); generate BMP file # i

    (command "_.LAYER" "_OFF" i "_FREEZE" i "")
    (setq i (+ i 1))
  ); end while

  (write-line "rscript" SCRfile); write the last line to .SCR file
  (close SCRfile)

;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  (setq i 1); restore old settings
  (command "_.LAYER")
  (while (<= i LayNr)
    (command "_ON" i "_THAW" i)
    (setq i (+ i 1))
  ); end while
  (command "" "_.REGEN" "")
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

  (alert (strcat "Frame files created in the following directory:\n\n"Path))
  (setvar "CMDECHO"  OldCMDECHO ); back to old CMDECHO settings
  (setvar "BLIPMODE" OldBLIPMODE); back to old BLIPMODE settings
); end MOTION

;==========================================================================
; Print info message once loaded

(princ "\nFile loaded. Type M3D to run…")
(princ)
```

Appendix 5. Simulation results of the optimized five-link suspension (variant **1** in Chapter 6) performed using MCS.visualNastran 4D multibody simulation software.

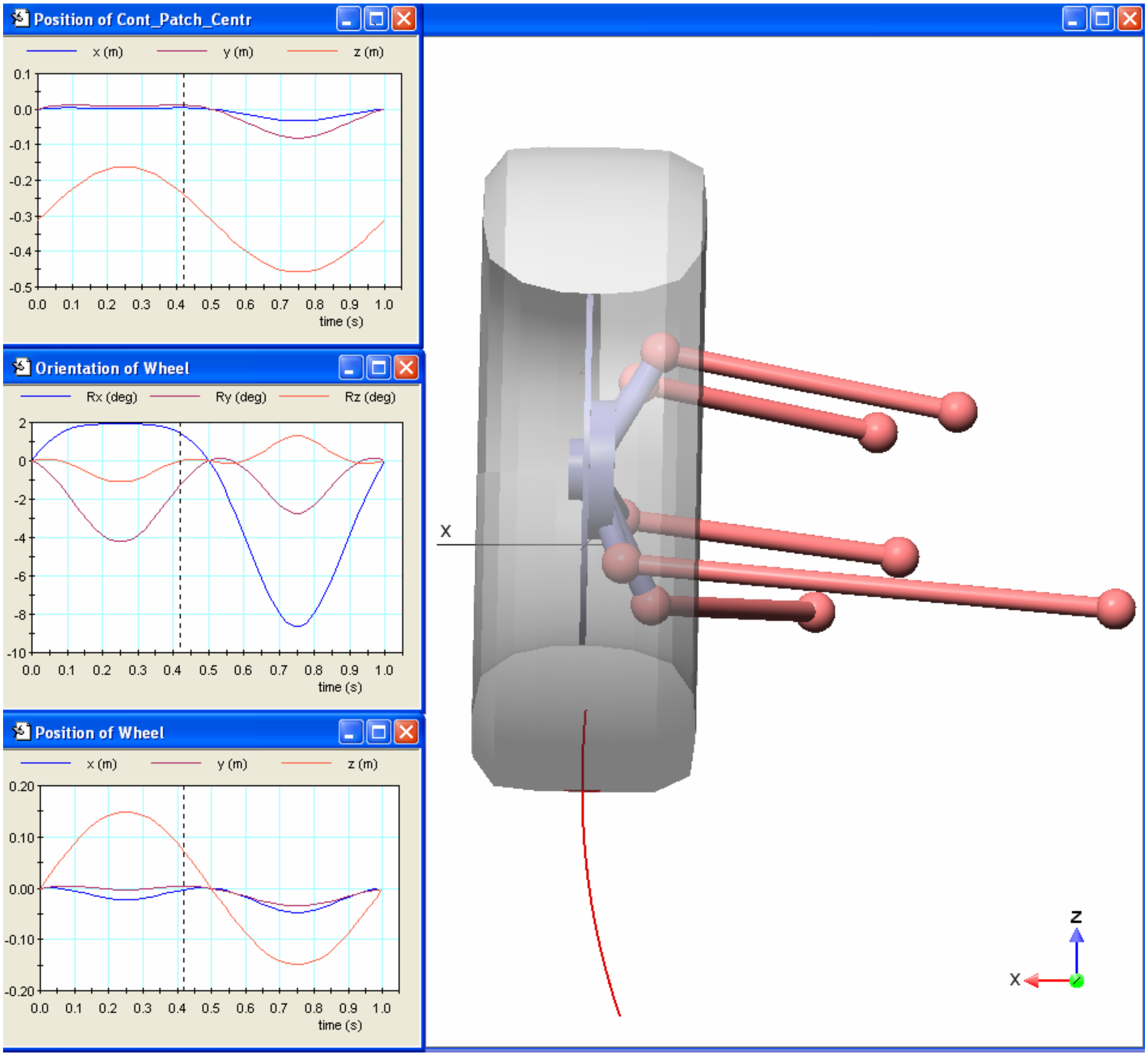


Fig. A.1 MSC.visualNastran 4D simulation screenshot

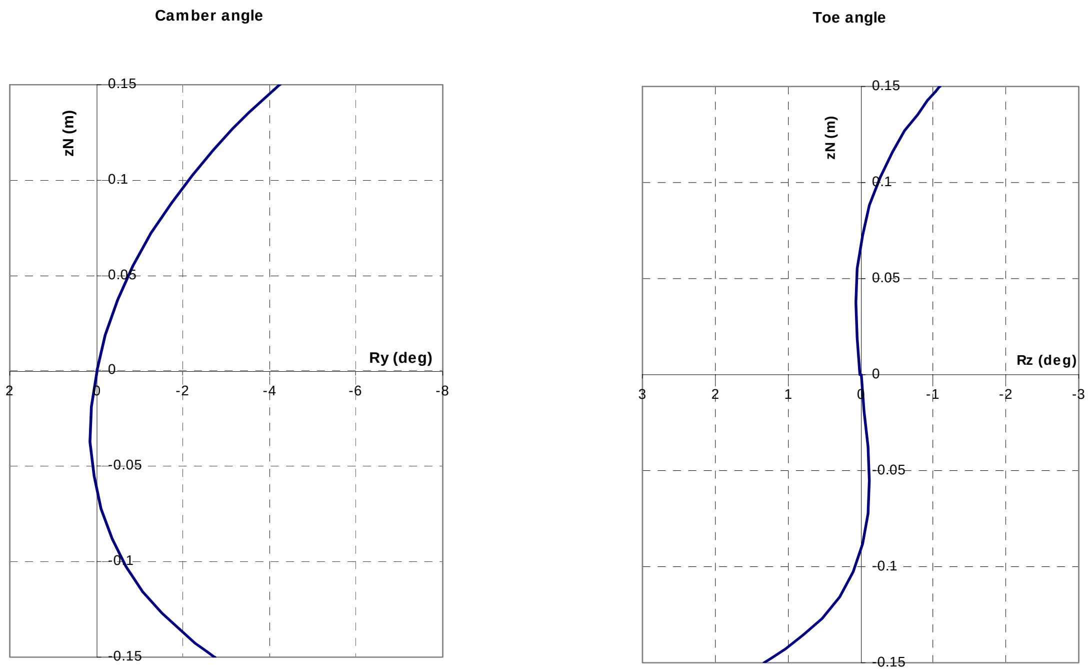


Fig. A.2 Camber angle and tow angle variations during jounce and rebound for the suspension variant **1** in Chapter 6 obtained with MSC.visualNastran.

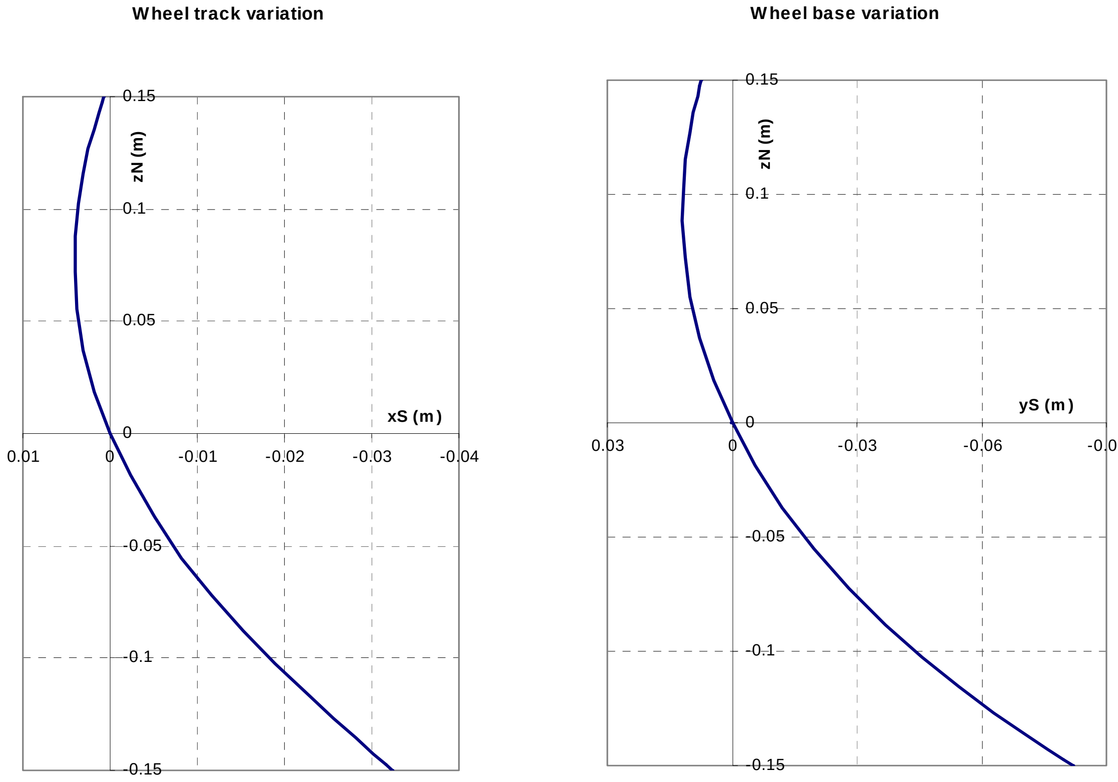


Fig. A.3 Wheel track and wheel base variation during jounce and rebound for the suspension variant **1** in Chapter 6 obtained with MSC.visualNastran.