BALLISTIC DESIGN OPTIMIZATION OF THREE-DIMENSIONAL GRAINS USING GENETIC ALGORITHMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

OSMAN YÜCEL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

SEPTEMBER 2012

Approval of the thesis:

BALLISTIC DESIGN OPTIMIZATION OF THREE-DIMENSIONAL GRAINS USING GENETIC ALGORITHMS

submitted by **OSMAN YÜCEL** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ____________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Süha Oral ____________________
Head of Department, **Mechanical Engineering**

Prof. Dr. M. Halûk Aksel ____________________
Supervisor, **Mechanical Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Kahraman Albayrak ____________________
Mechanical Engineering Dept., METU

Prof. Dr. M. Halûk Aksel ____________________
Mechanical Engineering Dept., METU

Prof. Dr. Zafer Dursunkaya ____________________
Mechanical Engineering Dept., METU

Prof. Dr. Abdullah Ulaş ____________________
Mechanical Engineering Dept., METU

Ö. Uğur Arkun, M.S. ____________________
Roketsan Missiles Industries Inc.

Date: 14.09.2012

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Osman YÜCEL

Signature :

ABSTRACT

BALLISTIC DESIGN OPTIMIZATION OF THREE-DIMENSIONAL GRAINS USING GENETIC ALGORITHMS

YÜCEL, Osman
M. S., Department of Mechanical Engineering
Supervisor: Prof. Dr. M. Halûk AKSEL

September 2012, 111 pages

Within the scope of this thesis study, an optimization tool for the ballistic design of three-dimensional grains in solid propellant rocket motors is developed. The modeling of grain geometry and burnback analysis is performed analytically by using basic geometries like cylinder, cone, sphere, ellipsoid, prism and torus. For the internal ballistic analysis, a quasi-steady zero-dimensional flow solver is used. Genetic algorithms have been studied and implemented to the design process as an optimization algorithm. Lastly, the developed optimization tool is validated with the predesigned rocket motors.

Key-words: Solid Rocket Motor, Grain Burnback Analysis, Internal Ballistics Design, Optimization, Genetic Algorithms

ÖZ

ÜÇ-BOYUTLU YAKIT ÇEKİRDEKLERİNİN GENETİK ALGORİTMALAR İLE BALİSTİK TASARIM OPTİMİZASYONU

YÜCEL, Osman
Yüksek Lisans, Makina Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. M. Halûk AKSEL

Eylül 2012, 111 sayfa

Bu tez çalışması kapsamında, katı yakıtlı roket motorlarında üç boyutlu yakıt çekirdeğinin balistik tasarımında kullanılmak üzere bir en iyileme aracı geliştirilmiştir. Yakıt çekirdeği geometrisinin modellenmesi ve geriye yanma analizleri; silindir, koni, küre, elipsoit, prizma ve torus gibi basit geometrik şekillerin kullanılması ile analitik olarak gerçekleştirilmiştir. İç balistik analizi için sıfır-boyutlu yarı-kararlı akış çözücü kullanılmıştır. Genetik algoritmalar üzerinde çalışılmış ve en iyileme algoritması olarak tasarım sürecine uygulanmıştır. Son olarak, daha önce tasarlanmış roket motorları kullanılarak geliştirilen en iyileme aracı doğrulanmıştır.

Anahtar Kelimeler: Katı Yakıtlı Roket Motoru, Geriye Yanma Analizi, İç Balistik Tasarım, Optimizasyon, Genetik Algoritmalar

To Burcu and my family

ACKNOWLEDGEMENT

I would like to express my deepest thanks and gratitude to Prof. Dr. M. Halûk AKSEL for his supervision, understanding and constant guidance.

I would like to thank to Uğur ARKUN, for his crucial suggestions and great support, and ROKETSAN for partially supporting this study.

I would like to express my sincere appreciation to my colleague Sevda AÇIK for supplying 0-D internal ballistics solver and her great advises during the preparation of this thesis. I also would like to thank to my colleagues Yusuf YILMAZ, Ahmet YILMAZ, Metin ACAR, Yusuf ATA, Berkan MUMCU and in particular Çağlayan DUYGU for their invaluable suggestions and supports.

I would like to thank to my friends Faruk ÇATALKAYA, Cihan CAN, M. Burak YAZGAN and Ferit TUT for their inestimable friendship and patience.

My wonderful darling Burcu DEDE is the one behind this success and I am forever grateful for her endless patience, unfailing support and encouragement during this difficult period.

Greatest love and thanks go to my entire family who supported and encouraged me throughout my whole life.

Lastly, this thesis is dedicated to my darling and my family who make the completion of this work possible.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF SYMBOLS

a	burning rate coefficient
A_b	burning surface area
A_t	nozzle throat area
c^*	characteristic exhaust velocity
c_p	specific heat under constant pressure
C_F	thrust coefficient
D	diameter
D_{out}	outer diameter
D_{port}	port diameter
f	fitness value
F	thrust force
F_{des}	calculated thrust
F_{obj}	objective (desired) thrust
g_0	gravitational acceleration at sea level
h	ratio of web increment to maximum web
I_{sp}	specific impulse
I_t	total impulse
k	ratio of station increment to web increment
K	burning surface area to nozzle throat area ratio
$\mathbf{l}$	vector of lower bounds on design variables
L	length of grain
$\dot{m}$	mass flow rate
m	mass
M	stored mass in the chamber, mach number, molecular weight
n	burning rate pressure exponent, number of design variables
N	number of star/slot points
N_b	total number of burn steps, number of bits
N_{gen}	generation number

N_p	population number
N_{tb}	total number of time data
N_x	total number of stations on x-axis
N_z	total number of stations on z-axis
p	pressure
p_{amb}	ambient pressure
p_{ref}	reference chamber pressure of burning rate
P_c	crossover probability
P_m	mutation probability
r	random number
r_1	fillet radius of star grain
r_2	cusp radius of star grain
r_b	propellant burning rate
r_{bref}	reference burning rate
R	gas constant
R_{tip}	tip radius of slotted grain
$R_{tipcenter}$	tip center radius of slotted grain
t	time
t_b	burning time
t_{inc}	time increment
T	temperature
T_{ref}	reference temperature of burning rate
$\mathbf{u}$	vector of upper bounds on design variables
V	velocity
$\forall$	volume
w	web
w_{inc}	web increment
w_p	propellant weight
$\mathbf{x}$	column vector of design variables
z	population size
x,y,z	cartesian space coordinates

γ	specific heat ratio
ε	nozzle expansion ratio
η	star point semi angle
η_{c^*}	efficiency of c^*
η_{C_F}	thrust efficiency
ξ	star angle
π_K	temperature sensitivity of pressure, %/ K
ρ	density
σ_p	temperature sensitivity of burning rate, % /K
υ	specific volume
∂	partial derivative operator

Subscripts:

ave	average
c	combustion chamber
del	delivered
e	nozzle exit
o	stagnation conditions
p	propellant
ref	reference
t	nozzle throat
theo	theoretical

CHAPTER 1

INTRODUCTION

Solid propellant rocket motor (SRM) is a class of rocket propulsion systems, which uses chemical energy from the combustion reaction of propellant chemicals. The propellant consisting of fuel and oxidizer chemicals is stored in the combustion chamber in solid phase and enables the rocket motor to operate in all environmental conditions. Solid rocket motors are widely used in military and civil applications since they have relatively simple design in comparison to other propulsion systems, are easy to integrate with a flight vehicle and require little servicing [1].

Principal components of the SRM shown in Figure 1.1 are given as follows:

1. **Grain**: The propellant having a special geometric form in motor case is called propellant grain. Grain is the chemical energy source of the propulsion system and generates high temperature and high pressure gas by ignition.
2. **Igniter:** Igniter produces the required pressure and thermal energy for the ignition of the grain.
3. **Motor Case:** The main structural part of the combustion chamber withstanding the high internal pressure resulting from the propellant combustion.
4. **Nozzle:** High pressure gas products are expanded to ambient pressure through nozzle. In this way, chemical energy of the propellant is converted to kinetic energy and thrust is obtained.

5. **Insulation:** The motor case and other structural components are protected from the high temperature of the combustion gases by insulation whose material has low thermal conductivity and high heat capacity.

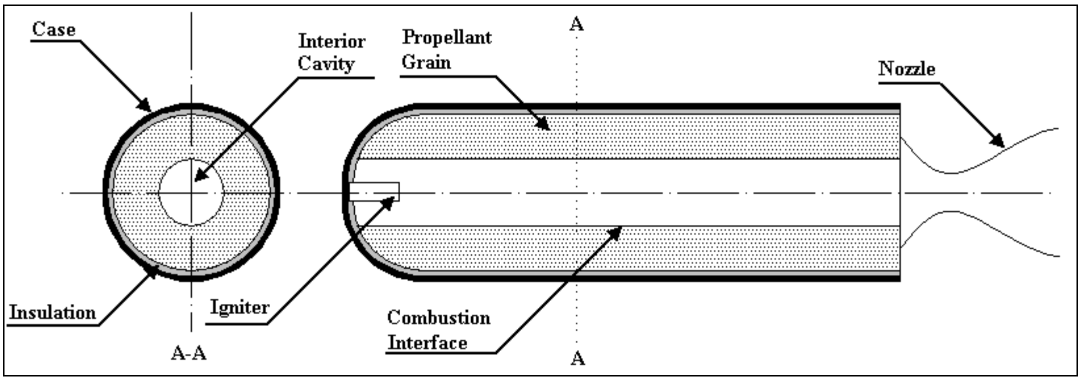


Figure 1.1 Main Parts of a Solid Propellant Rocket Motor

In the design process of solid rocket motor, after the performance requirements (total impulse, average thrust, burning time, thrust-time history, maximum thrust, maximum chamber pressure) and geometrical and weight constraints are defined, the conceptual design phase is started. In this phase, alternative design concepts meeting the design requirements are developed by selecting different structural materials, propellant type and grain configurations. Then, alternative design concepts are evaluated in terms of performance, manufacturability and cost. At the end of this phase, the best evaluated alternative is selected and the preliminary or detailed design phase is started.

After selecting the propellant type, material and grain configuration, detailed grain geometry is modeled. In this process, the grain geometry is evaluated in terms of ballistic and structural performance. After an iterative study between internal flow and structural analysis, the best geometry is chosen.

While modeling the grain geometry for the ballistic performance, generally the designers build up initial grain geometry with their knowledge and experience, and then try to find the solution in design space by changing the design variables manually by trial-and-error method. This ballistic design process, shown in Figure 1.2, ends when a grain geometry meeting the design requirements is found. Then the geometry is analyzed structurally.

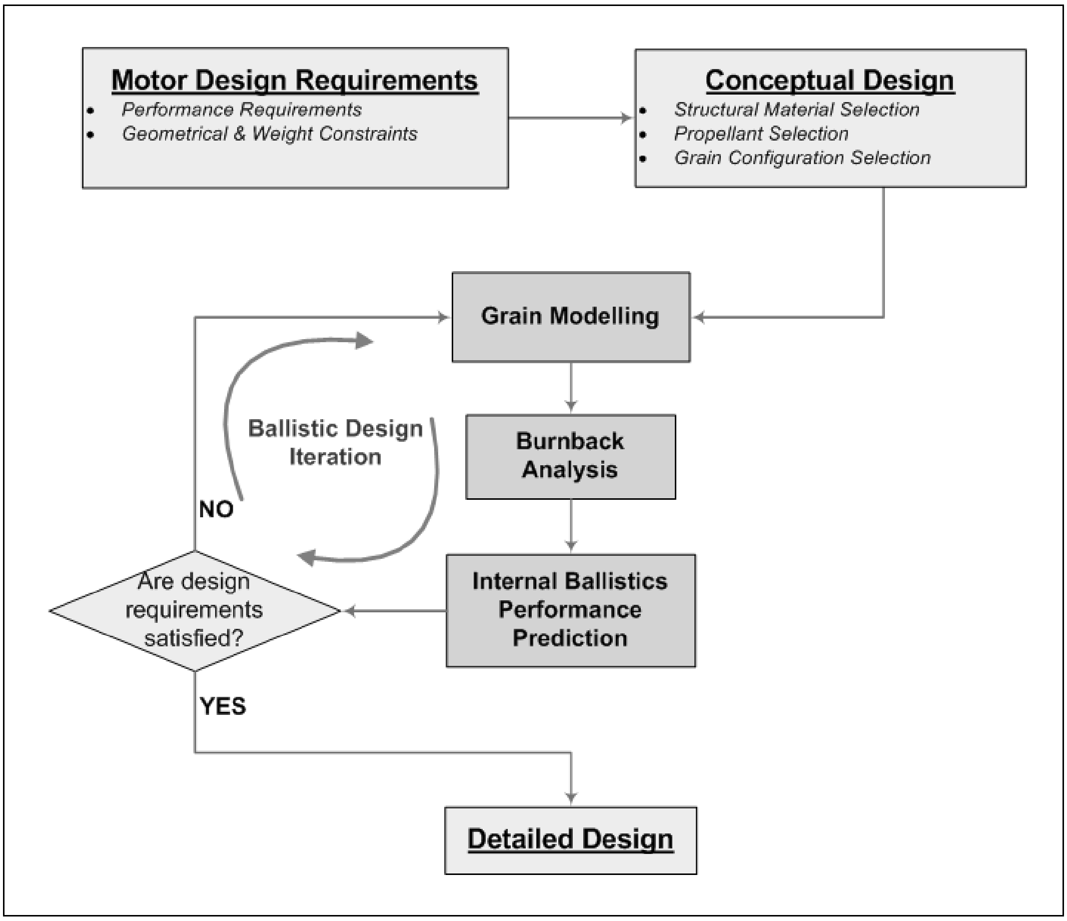

Motor Design Requirements
Performance Requirements
Geometrical & Weight Constraints
Conceptual Design
Structural Material Selection
Propellant Selection
Grain Configuration Selection
Grain Modelling
Ballistic Design Iteration
Burnback Analysis
NO
Are design requirements satisfied?
Internal Ballistics Performance Prediction
YES
Detailed Design

Figure 1.2 Ballistic Design Process of a SRM

In this thesis study, a grain design optimization tool is developed which offers an automated approach for the ballistic design process of 3-D grain configurations.

1.1 LITERATURE SURVEY

Literature review is based on the subject of automating the design process of solid rocket motors and finding the optimal grain design to meet certain performance criteria. This has been the subject of search since the 1960's; however, up to the 2000's, a limited treatment of this subject appears in the literature [2]. With the advances in computing technology, the subject has recently become a hot topic in the SRM design in the last decade.

The study of Billheimer [3] in 1968 was one of the first attempts at using an automated procedure to design a SRM. The physical modeling used in this study was limited because of the computational resources available at that time; however, this paper really acknowledged the importance of automating the design process for SRMs [2].

Woltosz [4] tried to find optimal solid grain geometry by using a pattern search technique. Woltosz determined five critical design dimensions which maximize the total impulse-to-motor weight ratio while meeting a specified minimum achieved velocity for a specific vehicle. This same pattern search technique was also used by Foster and Sforzini [5] in order to find the optimum values of five primary igniter design parameters. The objective in this study was to minimize the differences between computed and desired solid rocket motor ignition characteristics.

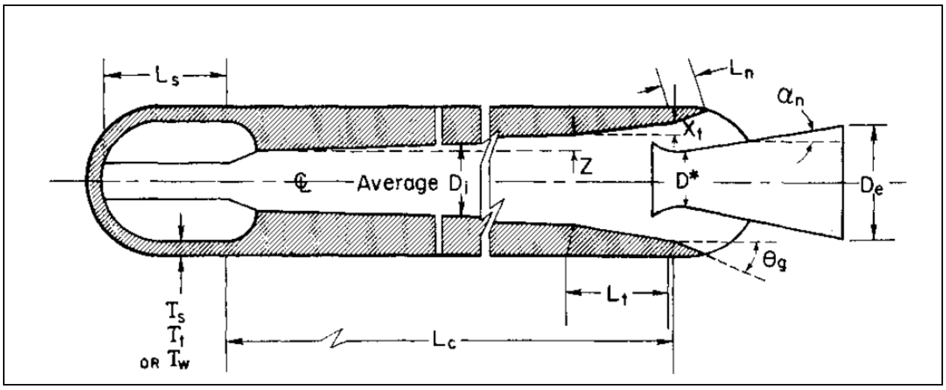


Figure 1.3 Design Variables of SRMDOP [6]

Sforzini [6] developed a computer program, called Solid Rocket Motor Design and Optimization Program (SRMDOP), by utilizing the same pattern search technique used by Woltosz for the SRM design. He modeled the grain geometry with fifteen geometric variables shown in Figure 1.3 and used analytical expressions to calculate the grain geometry during each phase of grain regression. By several modifications of the simplified ballistic analysis program described in detail in Reference [7], he performed ballistic performance evaluation of each design and tried to find the set of design parameters that will give the predicted thrust-time trace that most nearly matches a desired thrust-time trace as given in Figure 1.4.

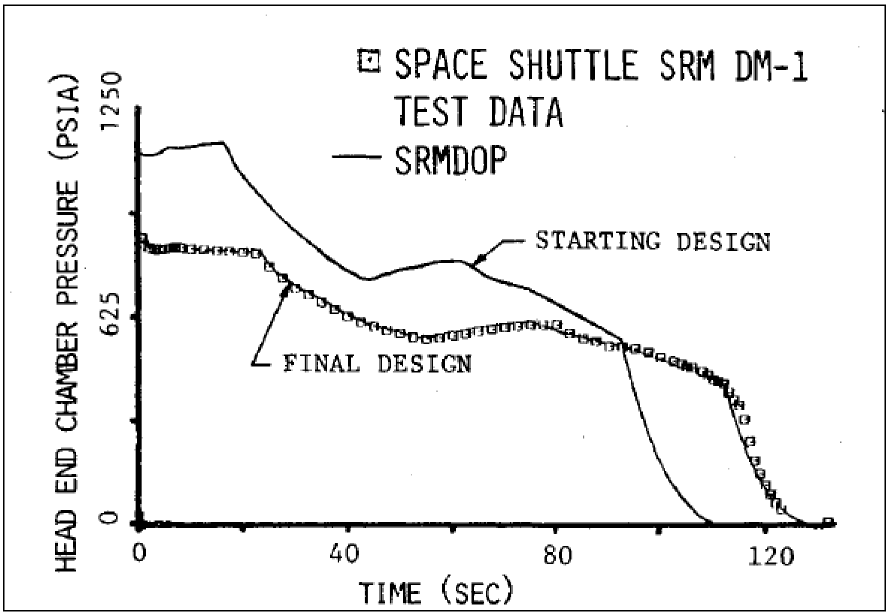


Figure 1.4 Chamber Pressure versus Time from Test Data and from SRMDOP [6]

Clegern [8] developed a program for the conceptual design and optimization of SRMs. Five main parameters; which are solid propellant, structure material, chamber pressure, outer motor diameter and nozzle half-angle, were studied as design parameters. Optimizing the design is done through an array solution algorithm comparing multiple designs based on the user entered data. Minimizing the total mass was taken as the objective of the study.

Anderson et al. [2] used genetic algorithms to design solid rocket motors as a component within an overall missile system. Totally twenty-four parameters shown in Appendix A were used as design variables; nine variables for SRM design, fourteen for external shape of the vehicle and one for launch angle verticality. In order to test the ability of genetic algorithms to work efficiently within a multidisciplinary framework; multiple goals, such as maximized range, minimized g-loading, minimized takeoff weight, and maximized fuel volume were used. Best design solution, maximizing the range and minimizing maximum g-load, is shown in Figure 1.5.

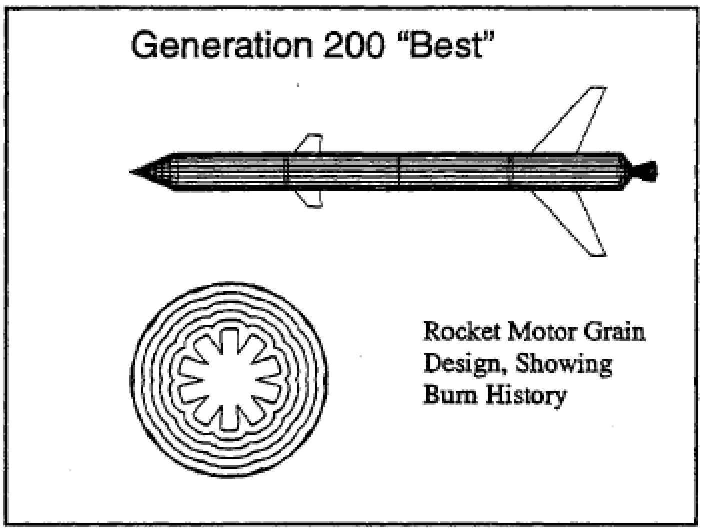


Figure 1.5 Best Design After 200 Generation [2]

Clay [9] developed a three stage optimization process for SRM design; approximation, global optimization, and high-fidelity optimization. For this optimization methodology; he employed with DOE, genetic algorithms and the BFGS first-order gradient-based algorithm. Interactive Missile Design (IMD) software developed by Lockheed Martin, Missiles and Fire Control was used geometry modeling and performance analysis. In the one of the test cases, five geometric parameters of a multi-cylinder grain configuration shown in Figure 1.6 were optimized by finding a grain design whose thrust versus time data fit best on a desired data.

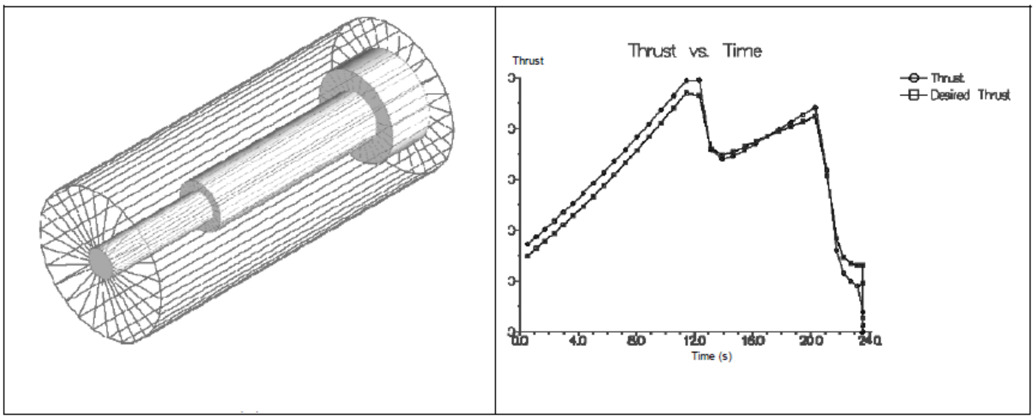


Figure 1.6 Multi-Cylinder Optimized Grain and Thrust versus Time Curves [9]

Nisar and Guozhu presented a methodology for design optimization of wagon wheel grain [10] and for design optimization of SRM finocyl grain [11]. In both studies, they utilize a hybrid optimization technique by using genetic algorithms for global convergence integrated with sequential quadratic programming for further local convergence of the solution thus attaining the final optimal solution. In the latter study, they modeled the grain geometry parametrically with seventeen basic geometries given in Appendix B and calculated the burning surface area during each phase of grain regression by using analytical method given in Reference [18]. For the ballistic performance evaluation, lumped parameter method was used and the objective of optimization was determined as minimizing the propellant mass while meeting the design constraints.

Kamran and Guozhu [12] studied on design and optimization of 3-D radial slot grain configuration by using genetic algorithm. They used CAD software for geometry modeling and surface regression. Similar to Nisar's study, lumped parameter method was utilized for the ballistic performance evaluation. Details of this study is given in Section .

Açık [13] developed a design and optimization tool for 2-D grain configurations by using complex method. In her study; star, slot, tubular and slotted-tube grains were employed. For the ballistic performance evaluation; an internal ballistic solver that uses 0-D quasi-steady model for the flow in combustion chamber and steady 1-D isentropic flow equations in the nozzle, were utilized. In Section 5.1.1, the test case in her study is used for comparison of the performance of the tool developed by this thesis study.

Kamran and Guozhu [14] applied an integrated approach using hyper-heuristic method based on genetic algorithm, simulated annealing and particle swarm method in optimization of solid rocket motor. They aimed to devise an algorithm for solving a problem that is independent of problem scenario. Geometric modeling of grain, chamber and nozzle was done by CAD software and optimization tool tried to minimize the total mass of the SRM

1.2 SCOPE OF THE THESIS

The purpose of this study is to develop a grain design optimization tool which offers an automated approach for 3-D grains in ballistic design process. For a given objective, optimization tool tries to find the optimum propellant grain geometry and nozzle geometry. This optimization process will aid the solid rocket motor design engineer in making the best initial design selections and thereby reducing the overall design cycle time of a project.

Optimization tool developed in this study consists of 3 main modules which are geometric modeling and burnback analysis of the propellant grain, ballistic performance prediction of solid rocket motor and the optimization algorithm.

3-D grains are geometrically modeled by using simple geometries like cylinder, cone, sphere, ellipsoid, prism and torus whose surface regression can be calculated easily. Then, burnback analysis is conducted analytically by enlarging or shrinking the volumes of these geometries and computing the burning surface area

For the performance prediction of a rocket motor, the internal ballistic solver developed by Açık [13] is used. The solver calculates the ballistic performance parameters by using 0-D quasi-steady flow equations in the combustion chamber and 1-D isentropic flow equations in nozzle.

As an optimization algorithm, genetic algorithms which are derivative-free, global search methods are utilized. Different genetic operators are studied in order to find proper optimization parameters giving the best solution for the grain design problem.

The developed optimization tool is validated with the results of previously designed rocket motor data.

1.3 CONTENTS OF THE THESIS REPORT

In Chapter 2, details of the grain burnback analysis are explained. Firstly, commonly used 2-D and 3-D grain configurations are introduced. Secondly, burnback analysis methods in literature are given in three groups; analytical, numerical and drafting methods. Then, 3-D analytical burnback code (BB3D) which is developed within the scope of this thesis study is explained. Lastly, the validation of the BB3D code is presented.

Chapter 3 contains the detailed description of the internal ballistic solver. At first, the fundamental ballistic parameters of a SRM are introduced. Then, the main assumptions and governing equations of the solver are presented in this chapter.

In Chapter 4, genetic algorithms are discussed in detail. Optimization methods in literature are summarized; then, the theory of genetic algorithms and developed genetic optimizer code (GENOP) is presented. Lastly, the optimization code is validated with the cases whose solutions are known.

The details of the developed grain design optimization tool and its validation are presented in Chapter 5.

CHAPTER 2

GRAIN BURNBACK ANALYSIS

2.1 INTRODUCTION

As the propellant burns, the burning surface of the grain moves in a direction normal to the surface. This regression is called burnback. Because of the grain burnback, burning surface area of the grain and control volume of the flow change during rocket motor operation. Since these parameters directly affect chamber pressure, initial geometry of the grain and its change during action time are the crucial properties for the performance of the rocket motor. For this reason, grain designers perform the grain burnback analysis in order to determine the change of the grain geometry as the propellant burns.

Grain burnback analysis is a pure geometrical analysis which is the offsetting the burning surfaces through their normal direction, modifying the surfaces according to their intersection/interference and generating the new geometry. The offset distance is actually the burnt propellant thickness which is called “web”. After the burnback analysis, the obtained data like “web versus burn area, perimeter and port volume” are given to performance prediction analysis.

In this chapter, firstly, commonly used two-dimensional (2-D) and three-dimensional (3-D) grain configurations are introduced. Secondly, burnback analysis methods in literature are given as classified in three groups; analytical, numerical and drafting methods. Then, 3-D analytical burnback code (BB3D) which is developed within the scope of this thesis study is explained. Lastly, the validation of the code is presented.

2.2 GRAIN CONFIGURATIONS

Each grain configuration gives different pressure and thrust profile such as neutral, progressive, regressive or boost-sustain (Figure 2.1). According to motor requirements, designer should choose the appropriate configuration. In this section, commonly used grain geometries will be introduced in two groups as 2-D and 3-D grains.

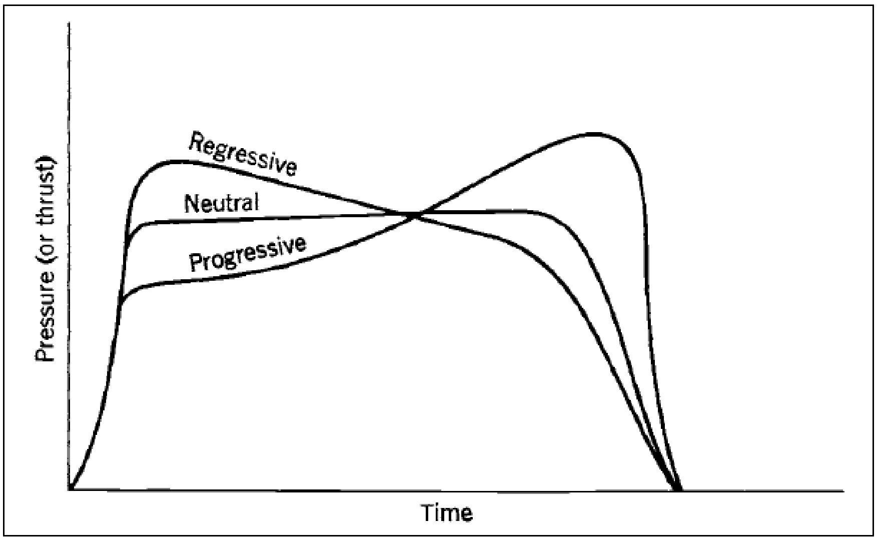


Figure 2.1 Classification of Grains According To Their Pressure-Time Characteristics [1]

2-D grain configurations are the grains that burn only longitudinally or only radially. On the other hand, 3-D grain configurations are commonly combination of 2-D grains, which combine both radial and longitudinal burning [1].

2.2.1 2-D Configurations

End burning, internal burning tube, star and wagon wheel grains are given as commonly used 2-D grains in the following sections.

2.2.1.1 End Burning Grain

End burning grain, given in Figure 2.2, is the simplest grain configuration which is defined by two variables: length *L* and diameter *D*.

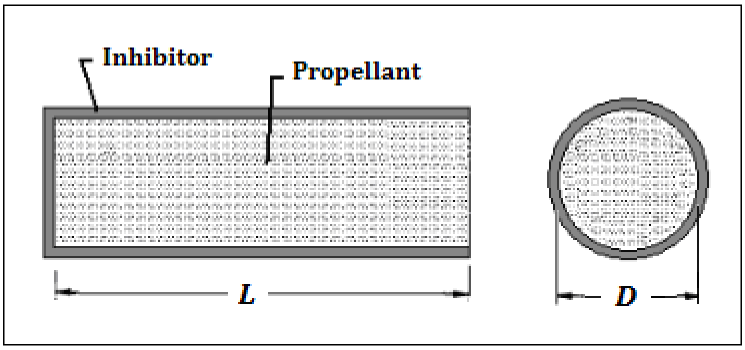


Figure 2.2 End Burning Grain

The propellant can only burn from the aft end of the grain and the burning surface merely recedes in the longitudinal (axial) direction. Thus, the burning area is defined as follows:

$$A_b = \frac{\pi.D^2}{4} \tag{2.1}$$

Since the burning area, calculated using Equation (2.1), is constant during the burn time, it provides neutral thrust-time curve. However, in larger motors (over 600 mm diameter) burning rate near propellant/liner interface becomes higher and the burning surface forms a conical shape. For this reason, these end burners in large motors show a progressive thrust curve [1].

The amount of propellant that can be placed in a given cylindrical motor case is the highest when the end burning grains are used. Therefore, the high volumetric loading (0.90-0.98) is the main advantage of these grains. On the other hand, the main drawback is low thrust level when compared to the size of the motor. Hence, end burners typically are applicable to missions requiring relatively long durations and low thrust level [15].

2.2.1.2 Internal Burning Tube Grain

The internal burning tube grain, given in Figure 2.3, is widely used grain configuration which is defined by a length *L* and two diameters D_{out} and D_{port}.

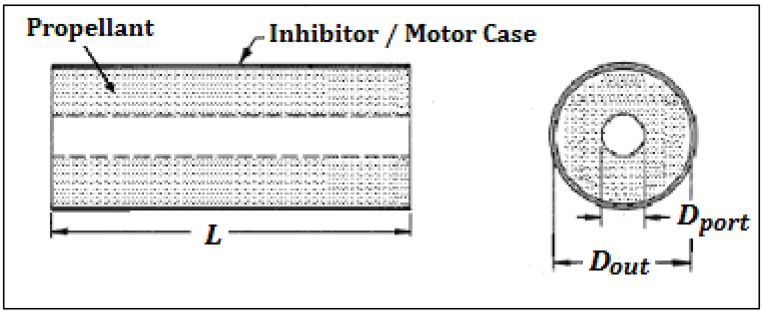


Figure 2.3 Internal Burning Tube Grain

When the head end and aft end of the grain is restricted to burn, it burns only in radial direction and becomes a progressive burning grain. But it is possible to get neutral burning with unrestricted ends. The burn area of the grain whose both ends are unrestricted can be calculated analytically as follows:

$$A_b = \pi(D_{port} + 2w)(L - 2w) + \frac{\pi}{2}\left[D_{out}^{\ 2} - (D_{port} + 2w)^2\right] \tag{2.2}$$

where D_{port} is the port diameter of the grain, D_{out} is the outer diameter of the grain, *L* is the length of the grain and *w* is the web.

By using above equation, it can be determined that internal burning tube grain gives neutral burning with unrestricted ends of L/D_{out} ~2. But it becomes significantly progressive for $L/D_{out} > 2$ [15].

2.2.1.3 Star Grain

Star grain, given in Figure 2.5, is radially burning grain configuration which has 7 independent geometric parameters: D_{out}, r_1, r_2, w, η, ξ and N. By changing these parameters, it is possible to get neutral, progressive or regressive thrust profile from a star grain. Because of this design flexibility, star grain is widely used in rocket industry.

The outside round radius, r_2 can be replaced by inside radius or flat tip. Star grain with outside radius is named **convex star grain**, while the one with inside radius is named **concave star grain** (Figure 2.5). There is also a special form of concave star grain, called **slotted grain** or **truncated star grain** [16]. In slotted grains, shown in Figure 2.4, the star grain angles of η and ξ are equal to π/N; therefore, slotted grains have 5 independent geometric parameters; D_{out}, D_{port}, R_{tip}, $R_{tipcenter}$ and N.

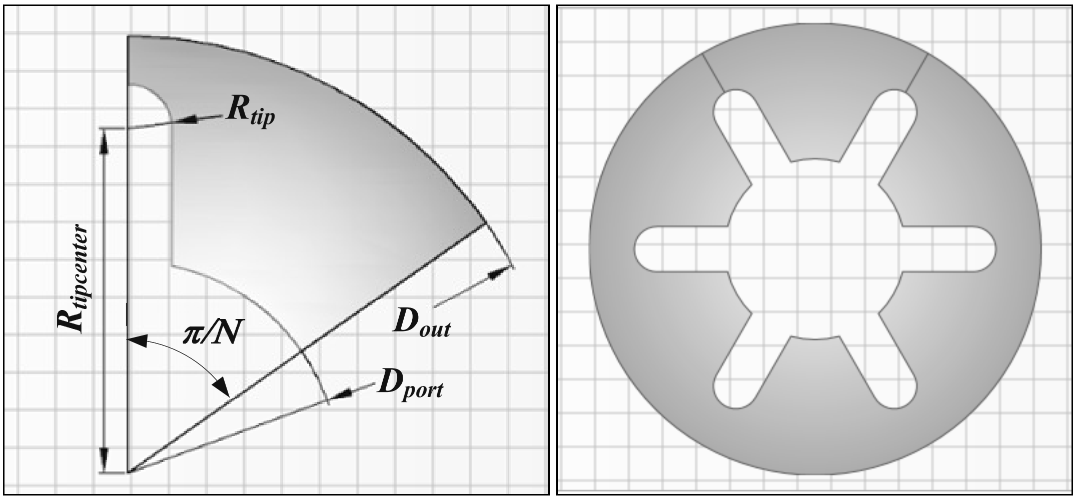
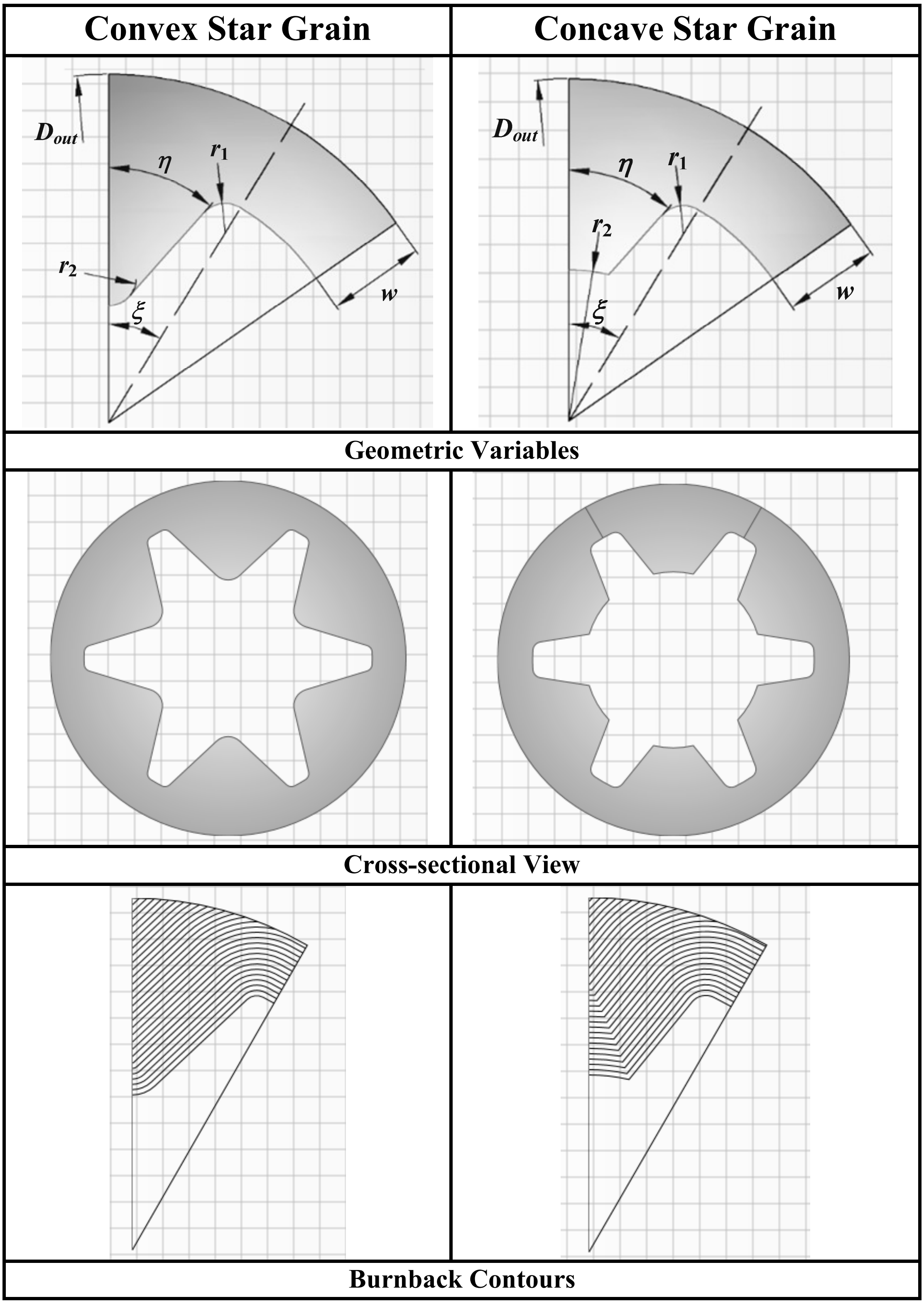


Figure 2.4 Slotted Grain (Truncated Star Grain)

The star configuration is neutral for web fractions ($2w/D_{out}$) of 0.3 to 0.4; and it is progressive above 0.4 [1]. However, sliver (remaining unburned propellant after motor operation) is an inherent characteristic of the star grain, the amount depending on the specific design. The effects of the geometric parameters on neutral burning and sliver can be found in reference [15].



Convex Star Grain

Concave Star Grain

Geometric Variables

Cross-sectional View

Burnback Contours

Figure 2.5 Convex and Concave Star Grains

2.2.1.4 Wagon Wheel Grain

Wagon wheel grain is the extension of star grain configuration having 7 independent parameters of star grain and 3 additional variables (*β*, L_a, r_3). Additional variables break the flat edges of the slot parts of star grain and increase the burning area.

The wagon wheel is used when web fractions of approximately 0.15 to 0.25 are required. Although the wagon wheel configuration has lower volumetric loading, 0.70, it is structurally superior to the star shape around web fraction of 0.3 and used for high thrust and short burn time requirements [1], [15].

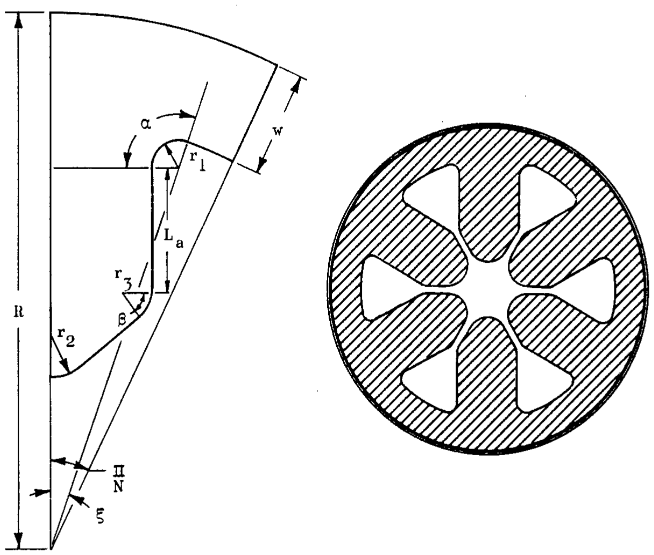


Figure 2.6 Wagon Wheel Grain [15]

2.2.2 3-D Grain Configurations

Conocyl (acronym for "cone in cylinder") and finocyl (acronym for "fin in cylinder"), given in Figure 2.7, are 3-D grain configurations.

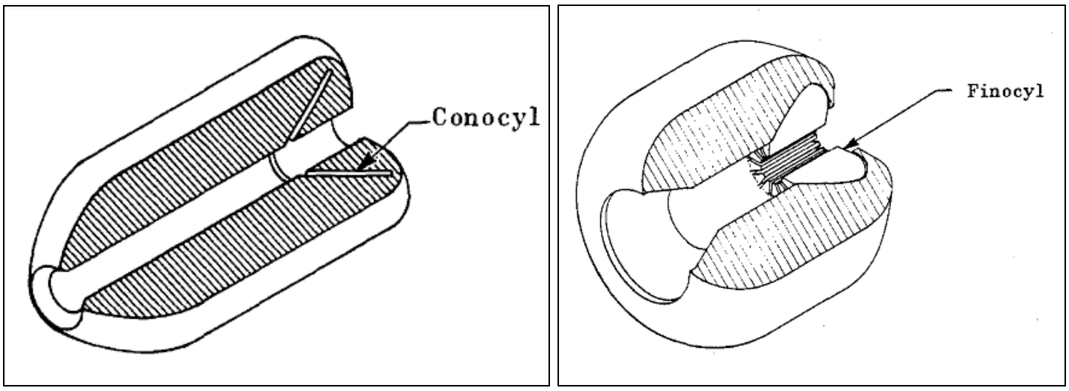


Figure 2.7 Conocyl and Finocyl Grains [15]

Conocyl grain utilizes the progressive characteristic of an internal burning cylinder and the regressive feature of an external-burning cone. Interaction of these two elements provides a ballistically acceptable grain configuration in terms of burning neutrally for a range value of L/D_{out} with an upper limit of approximately 4 [15].

Finocly grain is the combination of an internal-burning tube and a star grain. Main difference from conocyl grain is using axial slots rather than radial. Like cone parts of the conocyl, slotted portion provides regressive profile offsetting the progressive feature of internal-burning tube.

2.3 BURNBACK ANALYSIS

The methods in literature, used for burnback analysis, are explained in three groups as analytical, numerical and drafting methods.

2.3.1 Analytical Methods

The analytical approach has fallen out of favor in recent decades; however, for some classes of grains, the analytical methods are much more efficient than grid-based numerical techniques. For a grain design optimization process, in which large numbers of grain configurations are to be considered, generating grids for each candidate design is often prohibitive. For such optimization processes, analytical developments of burn perimeter and port area are critically important [16].

In analytical methods, the interface of grain and internal cavity is constructed analytically. In 2-D grain geometries, burnback analysis is done on the radial cross-sectional view of the grain with the interface elements; lines and arcs. At each burn step, lines and arcs are shifted with the offset amount of web and the perimeter of the burning interface is calculated. The difficulty is finding the intersection of the burning elements and generating new geometry by trimming or deleting the elements. For this reason, grain cross-section is divided into several zones as shown in Figure 2.8. At each zone, burning perimeters and port areas can be evaluated by means of the same exact formulas.

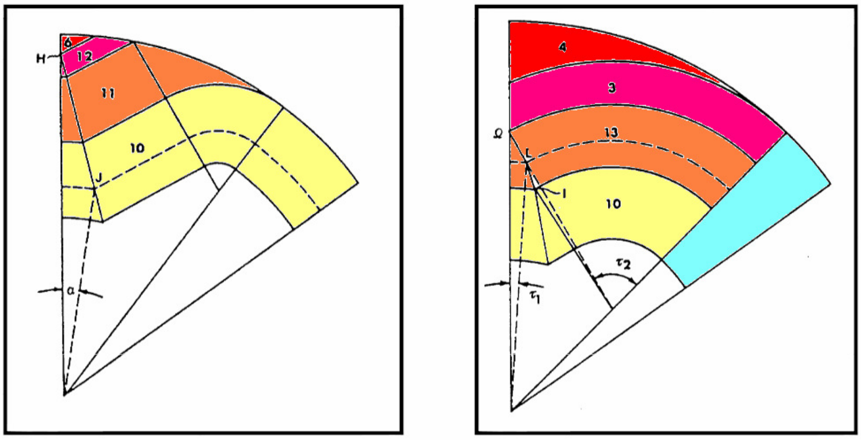


Figure 2.8 Star Grain Samples Divided into Several Zones [17]

In literature, Ricciardi's study [17] is one of the examples in which 16 different configurations were defined for analytical burnback evaluation of the star grain. In Reference [16], the equations for the star, long spoke wagon wheel, and dendrite grains were summarized and the development of the burnback equations for the short spoke wagon wheel and slotted grain configurations were included.

In the analytical methods used for the evaluation of the burnback of 3-D grain configurations, the geometry is divided into simple figures like cube, sphere, torus or pyramid whose surface regression is calculated easily (Figure 2.9). In 1967, Peterson et al. [18] developed a computer program for generalized 3-D grain design. Burnback simulation in the program was accomplished using 4 basic figures (right circular cylinder, right circular cone, right triangular prism and sphere) in various combinations to describe the initial grain geometry.

Another program called SPP (Solid Performance Program) [19], which was first released in 1975, uses a Grain Design Module based on reference [18]. The geometry calculations are based on the computation of volumes and changes in volumes of 5 basic shapes (right circular cone, right triangular prism, right circular cylinder, sphere and torus).

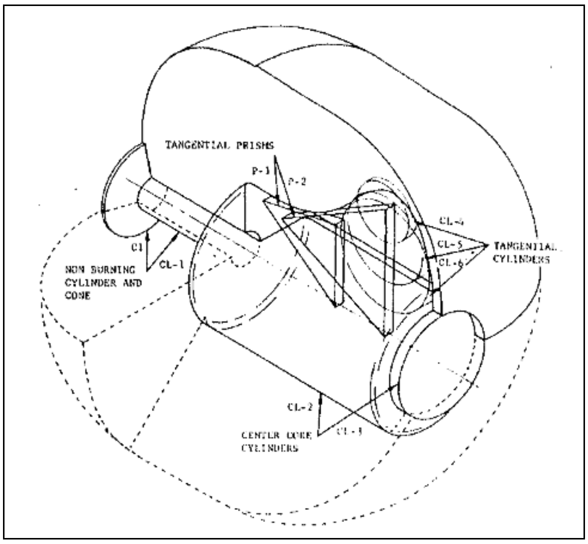


Figure 2.9 Simulation of 3-D Grain Void Using Basic Figures [18]

2.3.2 Numerical Methods

Since analytical methods are used for limited classes of grains due to its solution method, numerical algorithms are developed which are geometrically versatile. Instead of dividing the grain geometry into several zones or figures, these algorithms commonly use grid-based techniques which enable to model complex grain geometries and flow domain. Then, in order to evaluate the propellant grain surface regression, solution techniques for moving boundary problems are used. Main disadvantages of such methods are numerical errors involved and high computation time required for analysis. The details of these methods can be found in References [20] and [21].

Toker [20] and Yıldırım [21] applied Level-Set Method to the grain burnback problem. In the validation section of this chapter, the code developed by Toker [20] is used in order to compare the performance of analytical and numerical methods in 3-D grain burnback analysis. Therefore, Toker's study is summarized as follows:

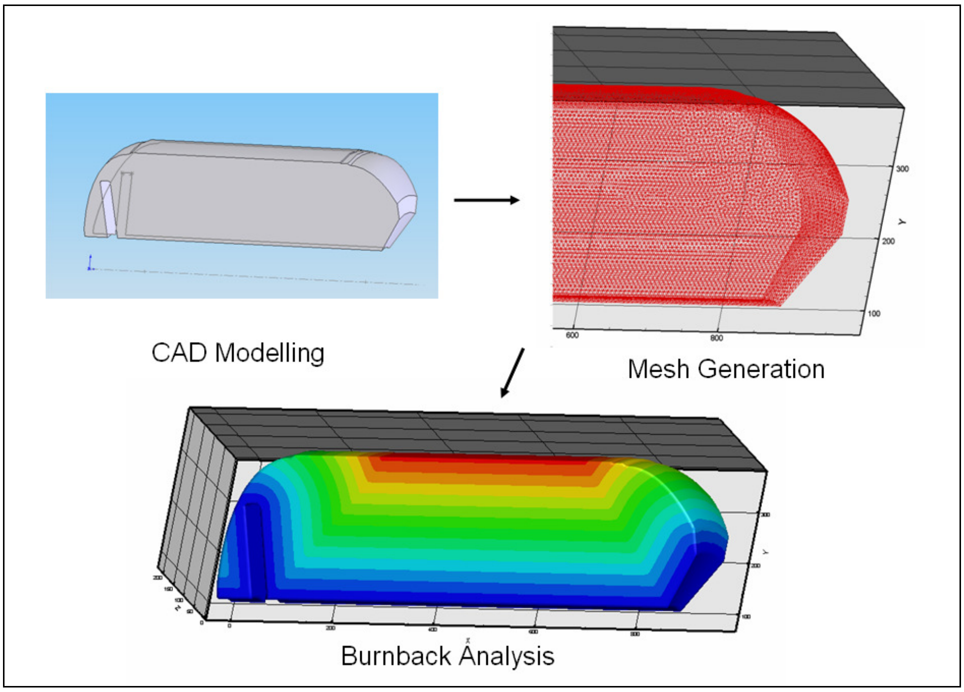


Figure 2.10 Procedure of Numerical Burnback Analysis

Firstly, 3-D grain is modeled using a CAD software program and then tetrahedron meshes are generated in the grain model by using a commercial mesh generator as shown in Figure 2.10. The mesh and boundary data are given to the burnback code, called FMM3 which uses Fast Marching Method to calculate the time at which the interface arrives at the each node. Then, burning surface area is calculated by using Cut-Cell Methodology.

2.3.3 Drafting Methods

In drafting techniques, before computer methods have become widespread, a detailed scale drawing of the grain configuration is prepared and the burning surfaces are shifted on the sketch by drafting tools. Perimeter and port area are measured and calculated by simple devices such as scales, map measurers and planimeters [15].

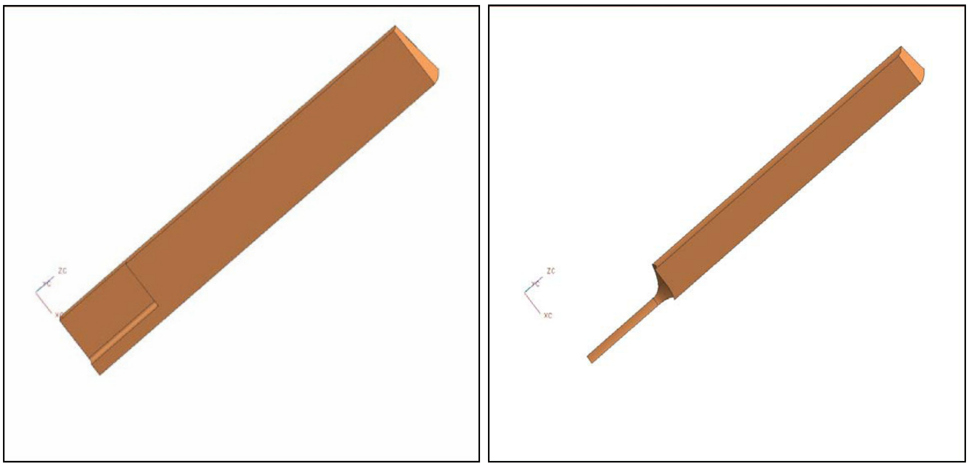


Figure 2.11 Solid Model of Grain at Different Burnback Steps [22]

Nowadays, CAD software programs are used for modeling and drafting the grain geometry. The geometry is modeled parametrically and the parameters which change during the burnback process are modified for each burn step [22].

2.4 3-D ANALYTICAL BURNBACK CODE (BB3D)

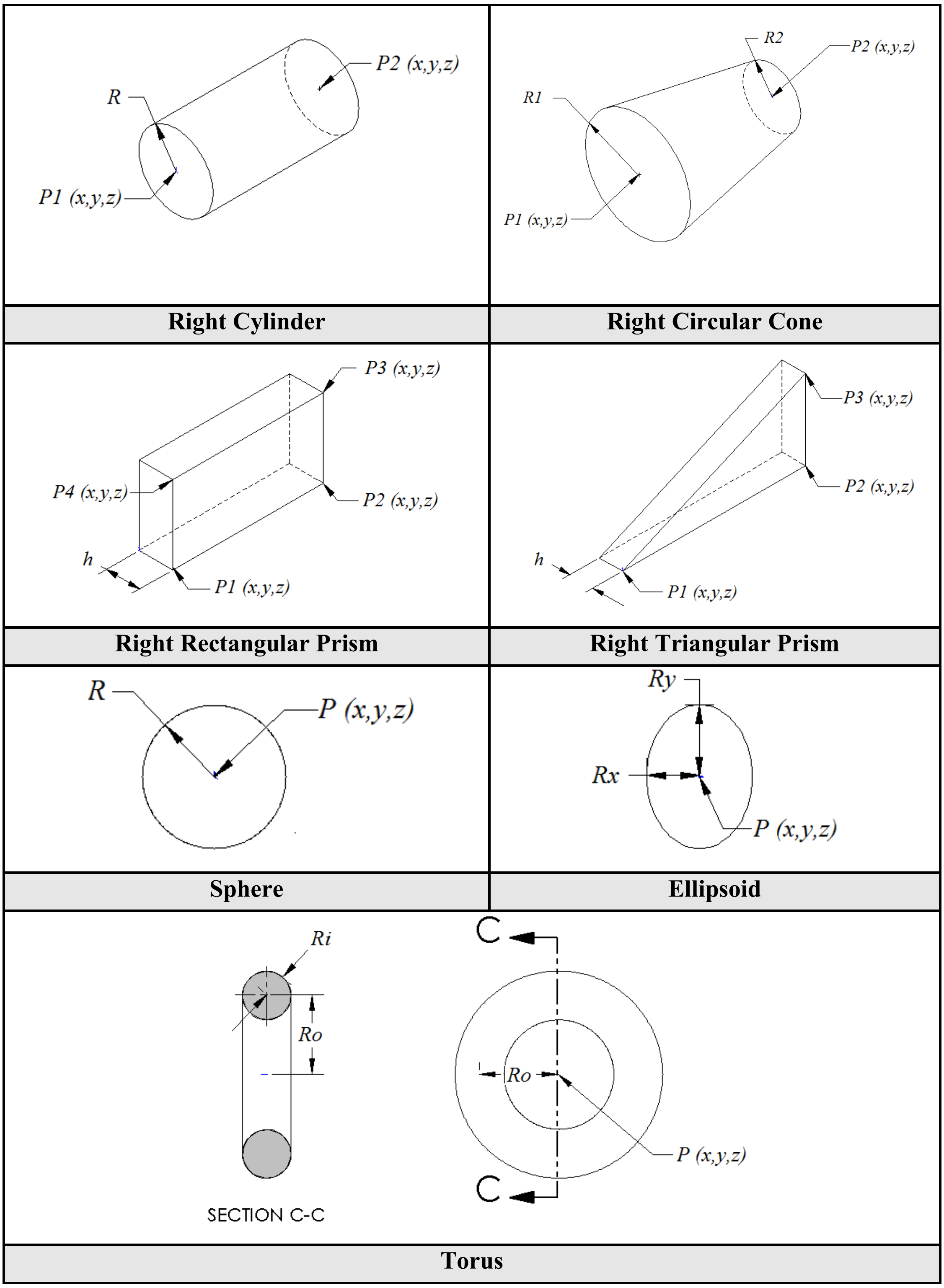
In this thesis, 3-D analytical burnback code called BB3D is developed for the grain design optimization process due to the following reasons:

- Since large numbers of grain alternatives are considered in the optimization process, computational time of burnback analysis becomes a highly critical parameter.
- Grid based numerical techniques and drafting techniques need user intervention during geometry modeling and mesh generation, which intervenes the optimization process.
- Burnback results of numerical methods involve numerical errors.

When the reasons presented above are considered, analytical burnback analysis distinguishes itself than other methods.

The solution method of the BB3D is based on the study of Peterson et al. [18]. The geometric modeling of the grain is accomplished using various combinations of some basic figures which are right cylinder, right circular cone, right rectangular prism, right triangular prism, sphere, ellipsoid and torus. The geometric variables defining these figures are given in Figure 2.12.

During geometric modeling, the case is assumed to be full of propellant initially and it is defined by using the above figures. It is also possible to define the outer boundary of the grain in terms of the radius versus the axial position. Then, inner void of the grain is defined by locating the basic figures in any orientation anywhere in space. It is not important that the figures may overlap or protrude out of the case. When the figures forming the voids are taken out from the case which is full of propellant, the grain is obtained. In this way, grain is modeled by using only geometric variables of basic figures.



Right Cylinder

Right Circular Cone

Right Rectangular Prism

Right Triangular Prism

Sphere

Ellipsoid

Torus

Figure 2.12 Basic Figures Used by BB3D

Mostly, grains are symmetric about the motor axis. Only modeling the symmetric section saves work and computation time required for the analysis. For example, instead of modeling the whole star grain with 8 star points, modeling the 1/16th of the grain and multiplying the results of the analysis by the symmetry factor of 16 provides faster solution. Therefore, symmetry factor is given as an input to the code and symmetry plane is created by the code.

BB3D calculates the burning surface area (A_b) by using the following equation:

$$A_b = \frac{\Delta\forall}{\Delta w} \tag{2.3}$$

where $\forall$ is the volume of the grain and w is the web.

Equation (2.3) states that the change in propellant volume with respect to the burn distance gives burning surface area. A_b lies somewhere between w and $w+\Delta w$ and the code assumes that A_b lies at the middle point ($w+\Delta w/2$).

In order to calculate the grain volume, firstly, geometry is divided into several stations along motor axis (x-axis) as given in Figure 2.13. The aim is to find the area, which is full of propellant, at each station (in y-z plane). For this reason, the station is scanned by drawing lines parallel to y axis, as shown in Figure 2.14.

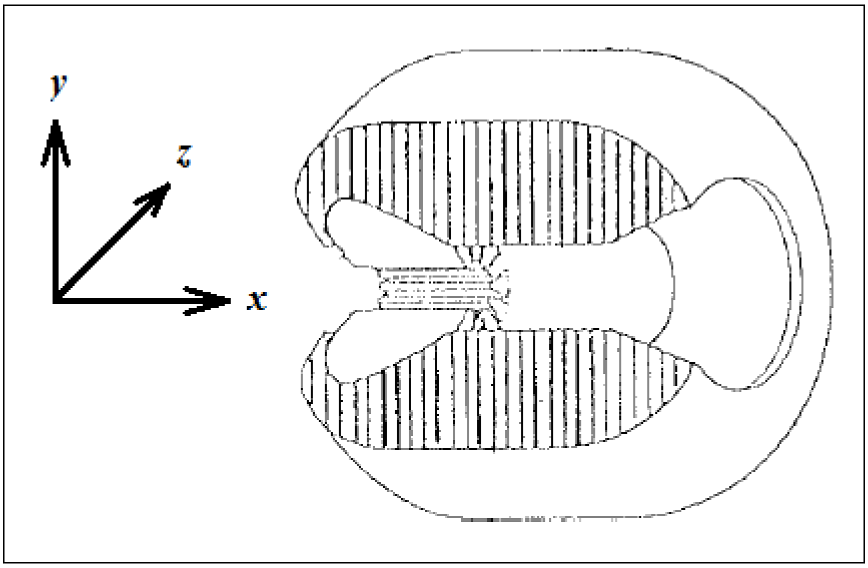


Figure 2.13 Stations Along Motor Axis (x axis)

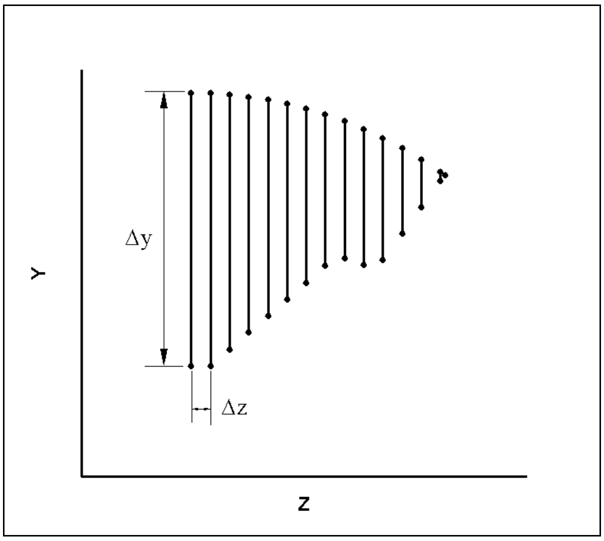


Figure 2.14 Lines on Radial Cross-Section (y-z plane)

The intersection points of each line with the basic figures located at that station are found by using the geometric equations of figures on y-z plane. The length of the each line (Δy) which is in the propellant zone is calculated. The average length of two successive lines is multiplied with the distance (Δz) between two lines in order to calculate the propellant area. The total propellant area, A_i, at that station (i) is found by using Equation (2.5).

$$A_i = \sum_{j=1}^{Nz} \frac{\left(\Delta y_{j+1} + \Delta y_j\right)}{2} \cdot \left(z_{j+1} - z_j\right) \tag{2.4}$$

The above procedure is applied for each station and the volume of the propellant grain, $\forall_n$, at burn step, *n*, is computed by Equation (2.5).

$$\forall_n = \sum_{i=1}^{Nx} \frac{\left(A_{i+1} + A_i\right)}{2} \cdot \left(x_{i+1} - x_i\right) \tag{2.5}$$

At the following burn step, the burning surface of figures defined as voids are offset with the amount of web thickness. During this operation, the figures are expanded by using the geometric formulas defining the basic figures. While the figures forming the case do not change, ones defined as propellant is shrunk. In this way, figures are redefined at the new burn step and the grain volume is recomputed, as described above. Lastly, the burning surface area is calculated according to Equation (2.3).

With this calculation method, it is also possible to find the mass and mass center of the grain. Therefore, the change of mass and mass center of the grain with respect to web thickness is easily calculated by the code.

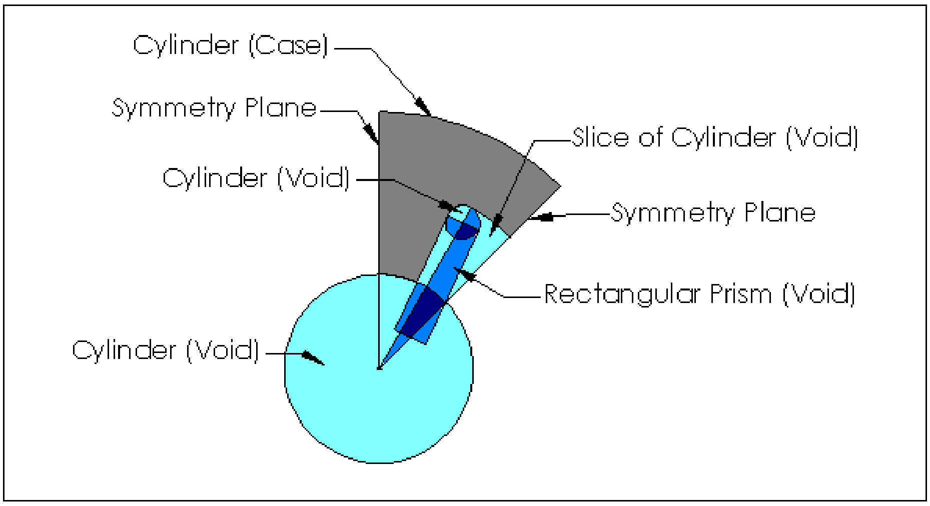


Figure 2.15 Figures Forming Star Grain Module

As presented in Figure 2.15, four figures (two whole cylinders, one slice of cylinder and one rectangular prism) are used to obtain the inner void of a convex star grain. Instead of defining four figures every time when star grain is analyzed, a star module is developed, which uses the geometric variables given in Section 2.2.1.3. Thus, user can define the geometry only with variables given in literature by using geometric modules and the basic figures forming the defined grain are automatically created. Convex star, concave star and slotted grain modules were developed in the scope of this thesis.

2.5 VALIDATION

In this section, in order to validate the 3-D analytical burnback code, the results of BB3D are compared with the results of other analytical, numerical and drafting methods used in literature. In addition, the effect of the station and web increments on the solution accuracy and computation time is investigated. During the study, a computer with Intel Core i5 2.67 GHz CPU and 4 GB RAM is used.

Two test cases are developed for the validation. For the first test case, a concave star grain, which is a simple and 2-D grain configuration, is analyzed. Then, a finocyl grain having slots at both ends, which is more complex and a 3-D grain configuration, is chosen as the second test case. The output data (web versus burning area) of different methods are compared according to the following equation:

$$Difference_{ave}(\%) = \frac{\sum_{n=1}^{N_b}\left(\sqrt{\left[\left(A_{b\,ref_n} - A_{b_{\;n}}\right)w_{inc}\right]^2}\right)}{\sum_{n=1}^{N_b}\left(A_{b\,ref_n} w_{inc}\right)} 100 \qquad (2.6)$$

where $A_{b\,ref_n}$ is the reference burning area at burn step n, $A_{b_{\;n}}$ is the calculated burning area at area at burn step n, w_{inc} is the web increment and N_b is the total number of burn step

2.5.1 Test Case-1

The geometric parameters of the concave star grain modeled for the burnback analysis are given in Table 2.1. Since the grain is 5 points star grain, 1/10 of the geometry is modeled for analysis.

Table 2.1 Geometric Parameters of Star Grain

D_{out}	180 mm	L_{grain}	400 mm	w	25 mm
η	49°	r_1	6 mm	N	5
ξ	33°	r_2	33 mm		

The burnback analysis is done by using 2-D analytical, 3-D analytical (BB3D), numerical (FMM) and CAD modeling methods. The code called STAR [13] which is based on Reference [17] is used for 2-D analytical burnback analysis. As a numerical method, FMM3 code is used, which is described in Section 2.3.2. For the drafting method, the solid model of the grain is modeled by CATIA software and the burning area is calculated by offsetting the burning surfaces. 2-D and 3-D burnback contours of BB3D solution are presented in Figure 2.16.

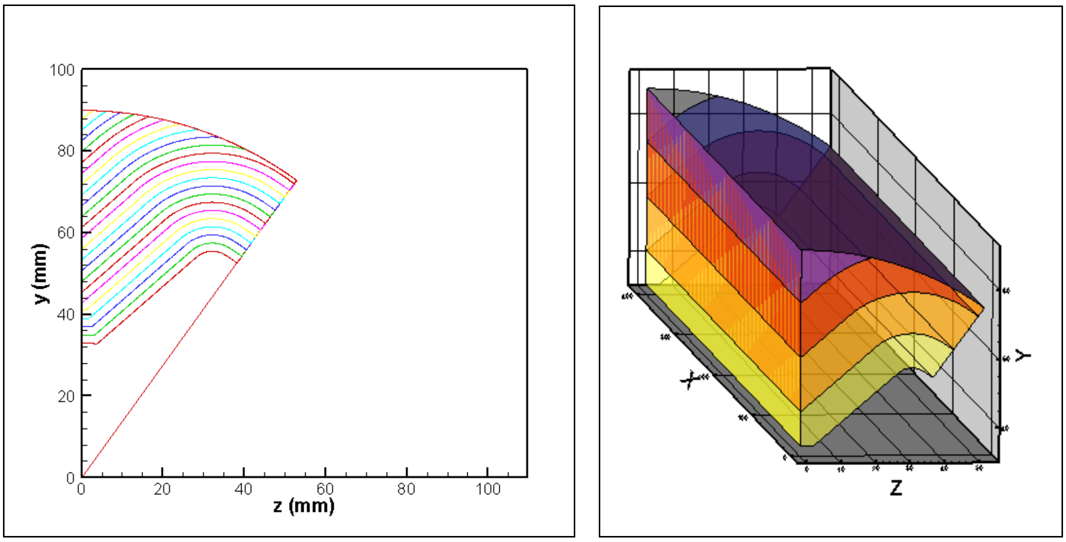


Figure 2.16 Burnback Contours of BB3D Solution

In order to analyze the effect of the axial and radial station increments used for the volume calculation in BB3D, burnback analysis is performed with different station increments. Instead of using different constant station increments in the study, the ratio of station increment to web increment (*k*) is used to make the solution independent from the grain size. The same increments are taken for the axial and radial stations, and the web-burn area data is calculated for 46 burn steps with a web increment of 1 mm by using different ratios of station increment to web increment such as 0.1, 0.2 0.5, 1.0 and 2.0. The results are compared with the solution of STAR code since the code gives the exact solution by using analytic equations. The error and computation time of the burnback analyses are given in Table 2.2 and Figure 2.17.

Table 2.2 Error and Computation Time for Different *k* Values

k	**Error (%)**	**Comp. Time (s)**
0.1	0.023	~ 150
0.2	0.027	~ 40
0.5	0.049	~ 6
1.0	0.232	~ 2
2.0	0.736	~ 0.5

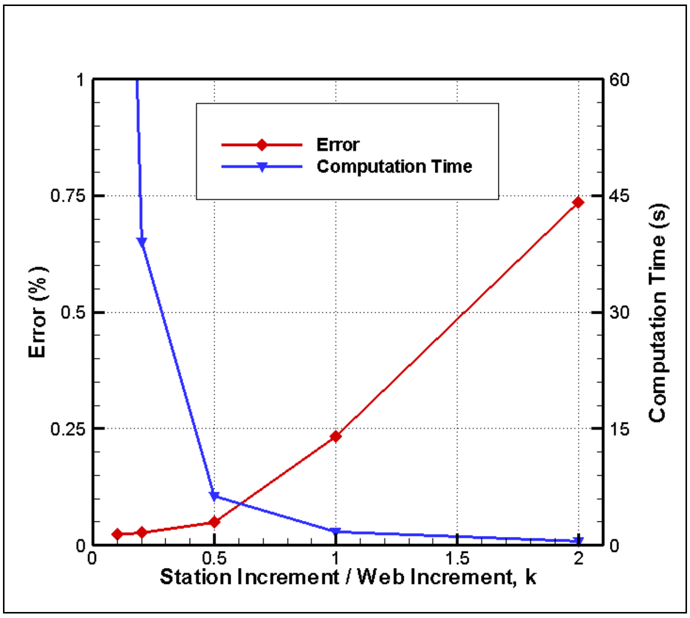


Figure 2.17 Error and Computation Time for Different *k* Values

As seen from Figure 2.17, the error decreases when smaller station increments are used. However, computation time increases rapidly when *k* is smaller than 0.5. Since the computation time is a significant parameter for the optimization process, the station increment can be increased up to web increment to reduce the computation time. If the station increment is taken larger than web increment, the calculation of the grain volume is done inaccurately as the void figures are enlarged with the amount of web thickness. This causes pulsation effect on the burn area calculation as seen from Figure 2.18 and increases the error rapidly. Therefore, *k* value can be taken in the range of 0.2 - 1.0 for the burnback analysis used in the optimization of 2-D grain geometries.

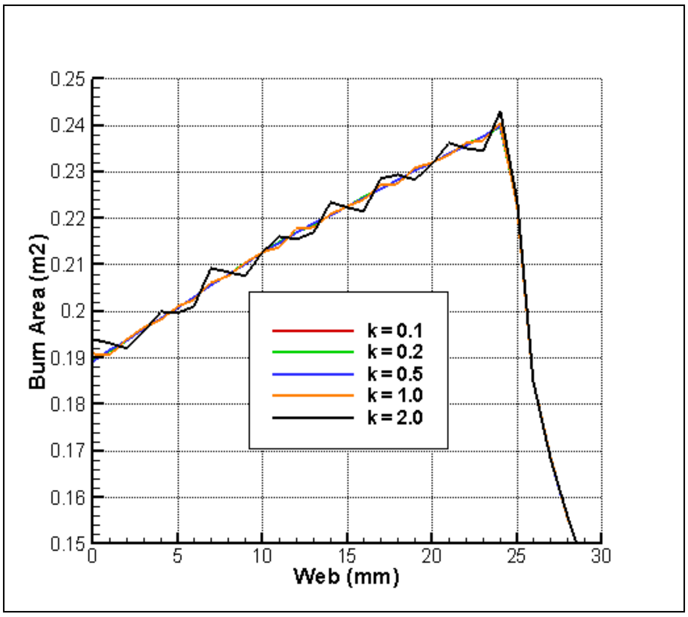


Figure 2.18 Web versus Burn Area for Different *k* Values

As the station increment, web increment is an important parameter affecting the solution accuracy and computation time of the burnback analysis. When the web increment is taken larger, the points where the burn area changes suddenly could not be captured and the error increases. Since using smaller web increments increase the computation time, a proper increment should be found for the burnback analysis used in optimization process. Therefore, burnback analysis is performed with different web increments. Instead of using different constant web increments in the study, the ratio of web increment to maximum web (*h*) is used to make the solution independent from the grain size. The plots of web versus burn area for the *h* values of 0.005, 0.01, 0.02, 0.05 and 0.1 are shown in Figure 2.19 and their computation time is given in Table 2.3.

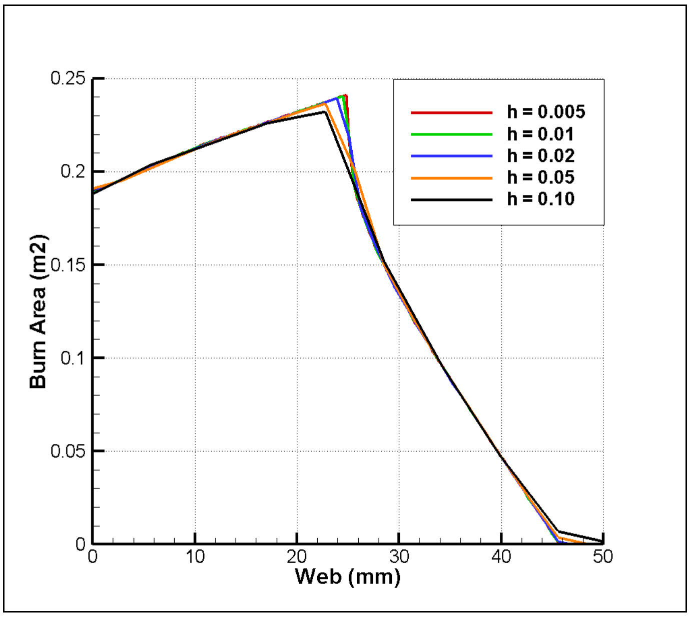


Figure 2.19 Web versus Burn Area for Different *h* Values

Table 2.3 Computation Time for Different *h* Values

h	Comp. Time (s)
0.10	~ 0.2
0.05	~ 0.5
0.02	~ 5
0.01	~ 30
0.005	~ 200

Analyzing Figure 2.19 and Table 2.3, it is seen that when the web increments which are 5 % and 10 % of the maximum web are used, the points of sudden burn area change could not be captured. For the high accuracy, 0.5 % of maximum web can be used; however, computation time increases significantly in this case. Therefore, *h* value can be taken in the range of 0.01 - 0.02 for the burnback analysis used in the optimization process.

After finding proper station and web increments for BB3D, the performance of BB3D is compared with numerical and CAD solutions. For the burnback analysis, the web increment giving the solution in sufficient accuracy is selected. As stated above, the web increment can be taken as 1.0 - 2.0 % of the maximum web for the analysis of BB3D. This web increment gives also accurate solution for the analysis performed by CAD software. In the thesis study of Püskülcü [22], the proposed web increment for the detailed analysis of grain geometry by CAD software is 1 % or 2 % of the web thickness. Therefore, the burnback analysis is performed with the web increment of 1 mm, which is the 1.75 % of the maximum web (h = 0.0175) for the comparison of solutions.

In BB3D, k is taken as 0.02. For the numerical analysis (FMM), the grain geometry is modeled with 1386428 elements and 251254 nodes. Since the number of elements affects the solution accuracy, possible smaller mesh size is used. In the CAD solution, the burn area is calculated directly from the surfaces of solid model of the grain and burning surfaces are shifted manually for each burn step. To measure the burn area, exact calculation mode is used in CATIA. The error is calculated according to Equation (2.6) by taking the solution of STAR code as a reference. The graphs of web versus burn area data are shown in Figure 2.20.

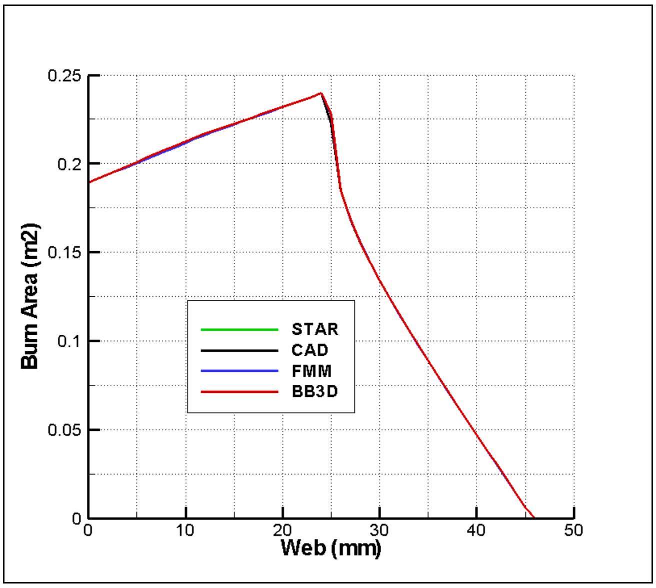


Figure 2.20 Web versus Burn Area

Table 2.4 Comparison of Results with 2-D Analytical Method (STAR)

	Error (%)	**Comp. Time**
BB3D	0.027	~ 40 s
FMM	0.229	~ 26 min
CAD	1.5x10^{-6}	-

As seen from Table 2.4, BB3D finds the solution in a shorter time, 39 seconds, with an error less than 0.02 %. The error of the numerical solution is less than 0.3 %. Considering the error, the numerical method may be used for the burnback analysis; however, its computation is too high, about half an hour, which is not acceptable for an optimization process. The error of the CAD solution is so small since the surface area calculation is performed by the numeric solution of surface equations. The drawback of burnback analysis with CAD software can be user intervention during offsetting the surfaces. Even if this can be done automatically by the exterior commands of a written code, user must intervene in the software to model the grain configuration.

2.5.2 Test Case-2

Finocyl grain which is a 3-D grain configuration is chosen as the second case. 1/16 of the geometry is modeled for analysis, since the fore end of the grain has 8 axial slots. Moreover, a cone and a radial slot are located at the aft end. The drawing of the grain is given in Appendix C.

For BB3D, the case is defined by 3 basic geometries: two ellipsoids and one cylinder. In order to obtain the inner void, totally 8 basic geometries (4 cylinders, 3 tori and one cone) are used. It is assumed that the void geometries are connected to central port; thus, the additional void geometries are not created in order to join the above geometries used at the slotted section with the central cylinder void. While scanning propellant area with lines on an *x*-station, the lines are trimmed with the outer intersection points of void figures. In this way, the number of void figures required for obtaining the inner void is decreased. 2-D burnback contours on radial cross-section at x = 200 mm and axial cross-section are shown in Figure 2.21 and Figure 2.22.

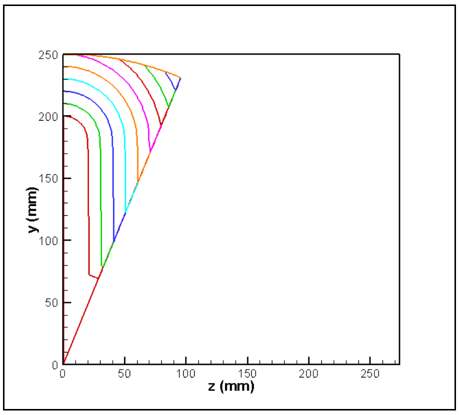


Figure 2.21 Burnback Contours on Radial Cross-Section at x=200mm

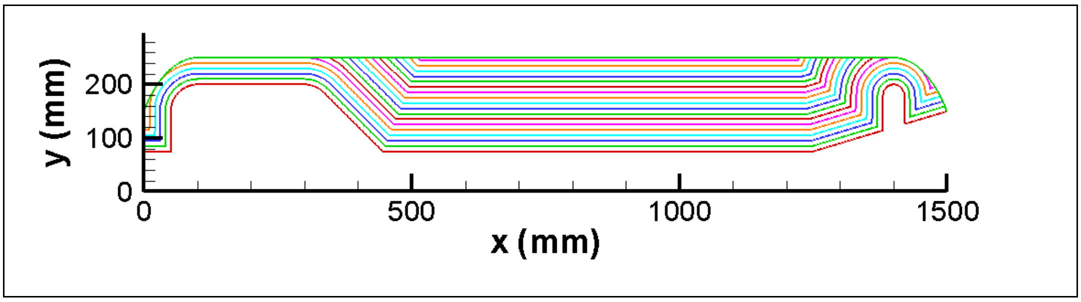


Figure 2.22 Burnback Contours on Axial Cross-Section

3-D burnback contours of the solution are presented in Figure 2.23.

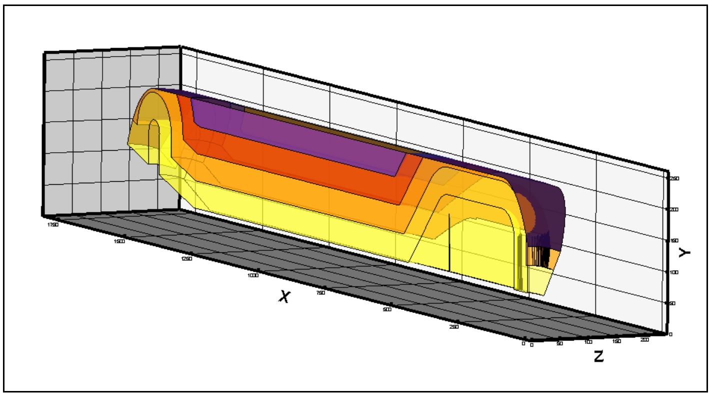


Figure 2.23 3-D Burnback Contours

In order to analyze the effect of the axial and radial station increments in BB3D for 3-D grain geometries, burnback analysis is performed with different *k* values, which are 0.1, 0.2 0.5, 1.0 and 2.0. The same increments are taken for the axial and radial stations, and the web-burn area data is calculated for 90 burn steps with a web increment of 2 mm ($h \approx 0.01$). The CAD solution is taken as the reference solution; because in the previous test case, it is seen that the results of exact solution (2-D analytical) and CAD solution are almost same. Therefore, results of the BB3D are compared with the solution of CAD modeling according to Equation (2.6). Computation time of the burnback analyses and differences between BB3D and CAD solution are given in Table 2.5 and Figure 2.24.

Table 2.5 Error and Computation Time for Different *k* Values

k	Difference (%)	Comp. Time (s)
0.1	0.097	~ 750
0.2	0.120	~ 190
0.5	0.181	~ 30
1.0	0.583	~ 8
2.0	9.476	~ 2

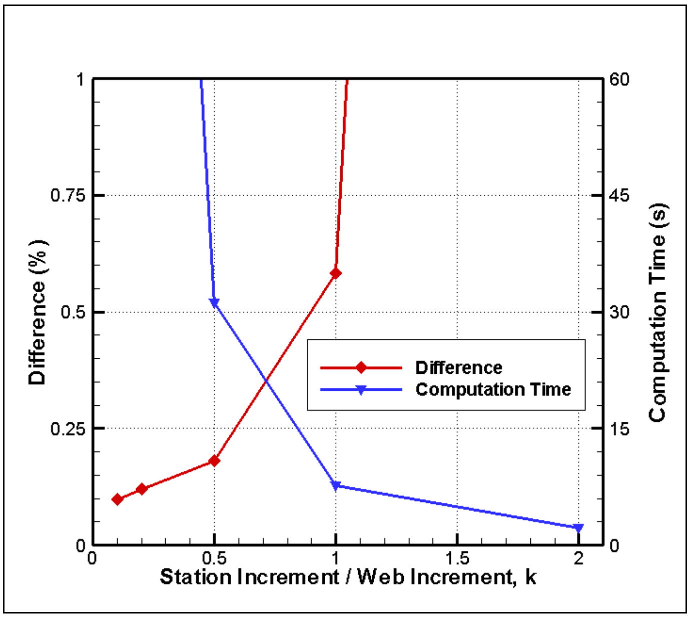


Figure 2.24 Error and Computation Time for Different *k* Values

As seen from the results, the effect of station increments on the solution accuracy and computation time of the burnback analysis of 3-D grain geometries is larger than analysis of 2-D geometries. Computation time increases rapidly when *k* is smaller than 0.5. In addition, due to the inaccurate volume calculation for the *k* values larger than web increment, the pulsation of the burn area increases significantly as seen from Figure 2.25. Therefore, *k* value can be taken in the range of 0.5 - 1.0 for the burnback analysis used in the optimization of 3-D grain geometries.

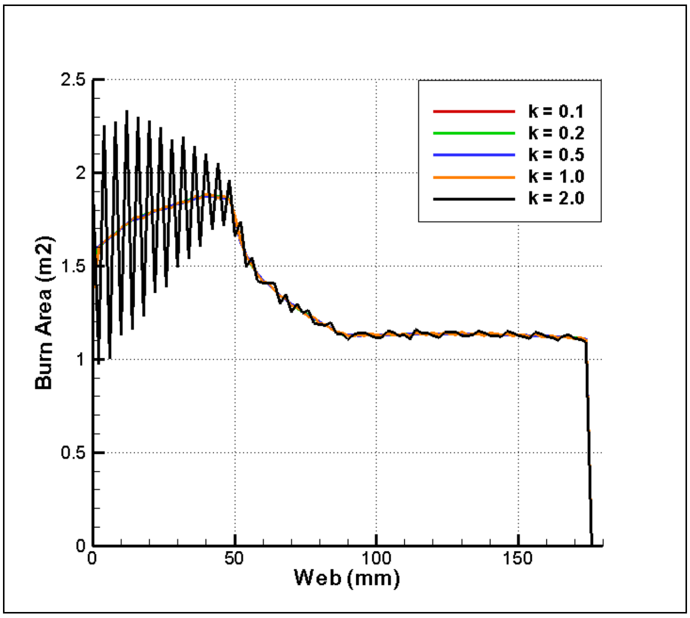


Figure 2.25 Web versus Burn Area for Different *k* Values

In order to analyze the effect of web increment on the solution, the burnback analysis is also performed with different web increments The plots of web versus burn area for the *h* values of 0.005, 0.01, 0.02, 0.05 and 0.1 are shown in Figure 2.26 and their computation time is given in Table 2.6.

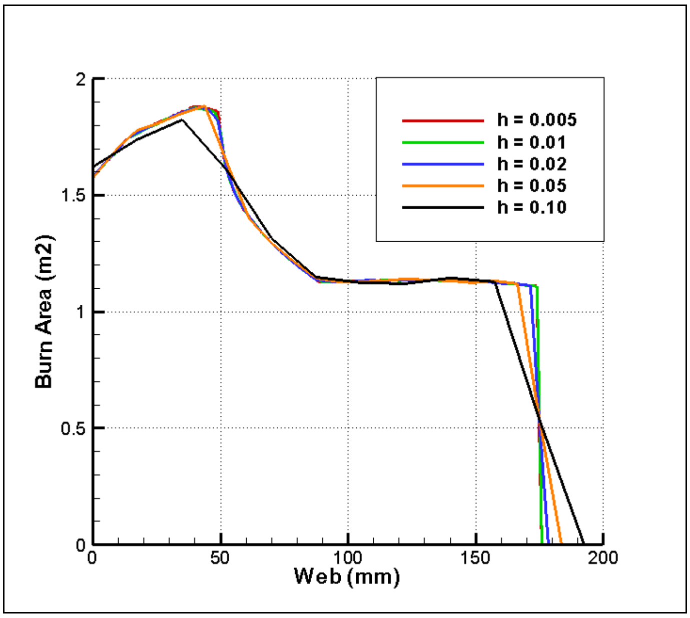


Figure 2.26 Web versus Burn Area for Different *h* Values

Table 2.6 Computation Time for Different *h* Values

h	Comp. Time (s)
0.10	~ 0.2
0.05	~ 0.7
0.02	~ 6
0.01	~ 30
0.005	~ 300

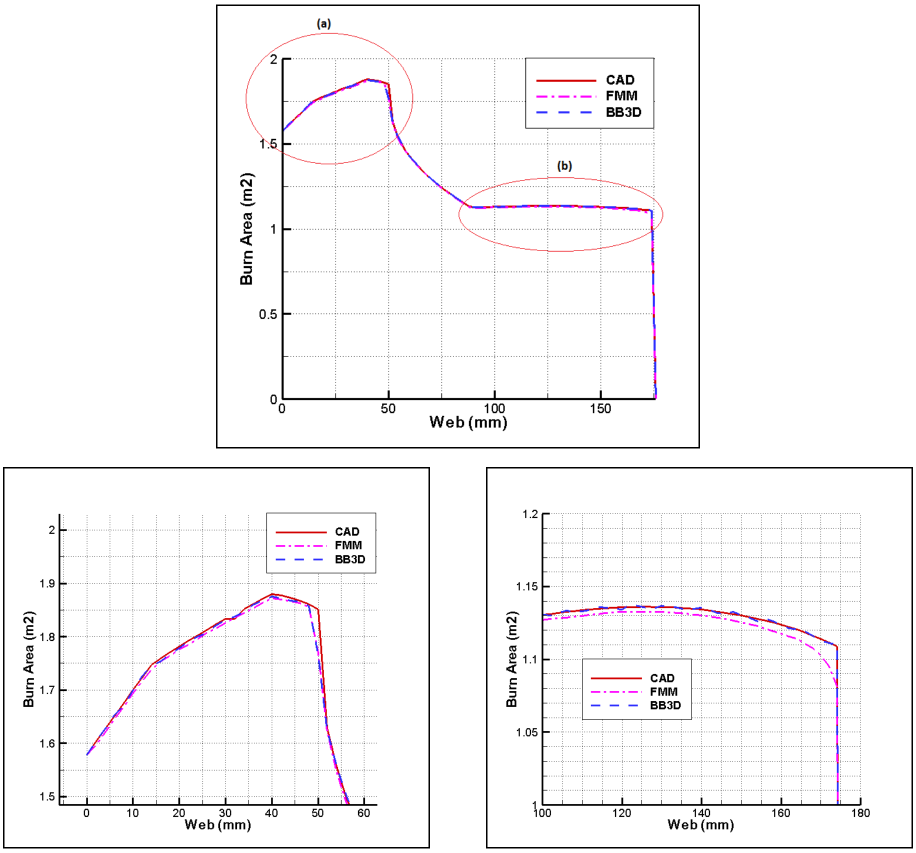
Analyzing Figure 2.26 and Table 2.6, the same conclusion with the previous case is done. The points of sudden burn area change could not be captured when h = 0.05 and h = 0.10. For the high accuracy, h = 0.005 can be used; however, computation time increases significantly in this case. Therefore, h value can be taken in the range of 0.01 - 0.02 for the burnback analysis used in the optimization process.

After finding proper station and web increments for BB3D in the study of 3-D grain geometry, the differences of BB3D and FMM from CAD solution are calculated according to Equation (2.6). CAD solution is taken as the reference solution, since CAD software finds the nearly exact solution when the web increment is properly chosen. The burnback analysis performed by using CAD software is assumed to be independent from web increment when h = 0.01, that is the web increment is equal to 1 % of the maximum web [22]. Therefore, burnback analysis is done with the web increment of 2 mm, which is about 1 % of the maximum web (h = 0.011) in this study.

In BB3D, k is taken as 0.05. For the numerical analysis (FMM), the grain geometry is modeled with 1256839 elements and 235300 nodes. Since the number of elements affects the solution accuracy, possible smaller mesh size is used. Results are given in Table 2.7. The graphs of web versus burn area data are also shown in Figure 2.27.

Table 2.7 Comparison of Results with CAD Solution

	Difference (%)	Comp. Time
BB3D	0.181	~ 30 s
FMM	0.441	~ 31 min



(a) (b)

Figure 2.27 Web versus Burn Area

As seen from Table 2.7, BB3D finds the solution in a shorter time, about 30 seconds, with a difference less than 0.2 %, which is applicable for the grain design optimization tool. The difference of the numerical solution is also small; however, its computation is too high, about half an hour, which is not acceptable for an optimization process

CHAPTER 3

INTERNAL BALLISTICS SOLVER

3.1 INTRODUCTION

Internal ballistics is the branch of applied science describing the combustion characteristics of the propellant, its burning rate, burning surface and grain geometry [1]. When the parameters specifying the ballistic performance of the rocket motor are considered, nozzle design can also be discussed in the scope of this science. In this chapter, firstly, the basic ballistic parameters of solid propellant rocket motors are introduced.

The performance characteristics of the rocket motor are obtained by the internal flow solution. For the ballistic performance simulation; 0-D quasi-steady, 1-D quasi-steady, 1-D unsteady and 3-D unsteady flow models are used commonly in literature [19]-[24]. When the computation time of flow solvers is concerned, lumped (0-D) models are generally used in optimization applications in order to explore the design space effectively. For this reason, a 0-D quasi-steady flow solver developed by Açık [13] is utilized in this thesis study. The main assumptions and the governing equations of the solver are explained in the following sections.

3.2 BALLISTIC PARAMETERS

3.2.1 Propellant Properties

3.2.1.1 Burning Rate

During the rocket motor operation, the burning surface of a propellant grain recedes in a direction essentially perpendicular to the surface. Burning rate is the linear regression rate of the flame edge, measured at a specific time and a specific distance on the propellant burning surface. The steady-state burning rate of a propellant (excluding the ignition phase and thrust tail-off) is defined by the ratio of minimum web to be burned (minimum distance traveled by the flame edge from the start of combustion to the time when the flame reaches the outside contour of the grain) versus steady-state burning time [15].

Burning rate of the propellant depends on [1]:

1. Propellant composition
2. Combustion chamber pressure
3. Initial temperature of the solid propellant
4. Combustion gas temperature
5. Velocity of the gas flow parallel to the burning surface
6. Motor motion (acceleration and spin)

The burning rate is a function of the chamber pressure. The relation between burning rate and the chamber pressure is governed by the following empirical equation, also known as Saint Robert's or Vielle’s burn rate law:

$$r_b = a p_c{}^n \tag{3.1}$$

where $a = r_{bref} \dfrac{e^{\sigma_p (T - Tref)}}{p_{ref}{}^n}$.

The values a (pre-exponential factor or temperature coefficient) and n (burning rate pressure exponent) are usually derived from strand burner tests or small subscale test motor firings at different operating pressures. r_{bref} is the reference burning rate value at a specific pressure, p_{ref}, and at a specific temperature, T_{ref}.

The sensitivity of burning rate to propellant temperature can be expressed in the form of temperature coefficients:

$$\sigma_p = \left[\frac{\partial \ln r_b}{\partial T}\right]_p = \frac{1}{r_b}\left[\frac{\partial r_b}{\partial T}\right]_p \tag{3.2}$$

$$\pi_K = \left[\frac{\partial \ln p}{\partial \ln T}\right]_K = \frac{1}{p}\left[\frac{\partial p}{\partial T}\right]_K \tag{3.3}$$

where σ_p is known as the temperature sensitivity of burning rate expressed as percent change of burning rate per degree change in propellant temperature at a particular value of the chamber pressure. The second one π_K is known as the temperature sensitivity of pressure expressed as percent change of chamber pressure per degree change in propellant temperature at a particular value of K which is the ratio of the burning surface area to nozzle throat area [1].

3.2.1.2 Specific Impulse

Specific impulse, I_{sp} is a measure of the impulse or momentum change that can be produced per unit weight of the propellant consumed. It is an important ballistic parameter of the propellant affecting the performance of a rocket propulsion system, similar in concept to the miles per gallon parameter used with automobiles [1].

Specific impulse is defined as:

$$I_{sp} = \frac{\int_0^{tb} F dt}{g_0 \int_0^{tb} \dot{m}_p dt} = \frac{I_t}{w_p} = \frac{F}{g_0 m_p} = \frac{c^* C_F}{g_0} \tag{3.4}$$

where F is the thrust force, t_b is the burning time, $\dot{m}_p$ is the propellant mass flow rate exhausted from the nozzle, g_0 is the gravitational acceleration at sea level, I_t is the total impulse, w_p is the weight of the propellant, m_p is the mass of the propellant, c^* is the characteristic exhaust velocity and C_F is the thrust coefficient.

3.2.1.3 Characteristic Exhaust Velocity

Characteristic exhaust velocity, $c*$, is a function of the propellant combustion process. As seen in Equation (3.7) it is proportional to $\sqrt{T_c / M}$, where T_c is the propellant flame temperature and M is the average molecular weight of the gas. Therefore, it has a slight dependence on chamber pressure; however, it is independent of nozzle characteristics [15]. The $c*$ is used in comparing the relative performance of different chemical rocket propulsion system designs and propellants; it is easily determined from measured data of propellant mass flow rate ($\dot{m}$), chamber pressure (p_c) and nozzle throat area (A_t). The delivered $c*$ can be formulated as:

$$c^*_{del} = \frac{p_c A_t}{\dot{m}} \tag{3.5}$$

The critical mass flow rate through chocked nozzle can be written as [25]:

$$\dot{m} = \frac{A_t V_t}{\upsilon_t} = A_t p_c \frac{\gamma \sqrt{\left(\frac{2}{\gamma + 1} \right)^{\frac{\gamma+1}{\gamma-1}}}}{\sqrt{\gamma R T_c}} \tag{3.6}$$

where V_t is the gas velocity at the nozzle throat, υ_t is the specific volume at the nozzle throat, T_c is the chamber temperature, γ is the specific heat ratio and R is the gas constant.

Substituting Equation (3.6) into Equation (3.5), theoretical $c*$ is found as:

$$c^*_{theo} = \frac{\sqrt{\gamma R T_c}}{\gamma\sqrt{\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{\gamma-1}}}} \tag{3.7}$$

Delivered $c*$ in the motor are less than theoretical values by a significant amount. The reduction in values is the results of [15]:

1. fluid flow losses including two-phase flow in which particles fail to achieve kinetic and thermal equilibrium,
2. heat losses to motor hardware and
3. combustion inefficiency.

Those losses occurring upstream of the nozzle throat plane affects the delivered $c*$. These losses can be indicated with $c*$ efficiency, η_{c*} as follows:

$$\eta_{c*} = \frac{c^*_{del}}{c^*_{theo}} \tag{3.8}$$

3.2.2 Chamber Pressure and MEOP

Chamber pressure is the static pressure measured at the head end of the internal gas flow; in other words, it is the pressure at the forward end of the combustion chamber [26]. It is the main parameter affecting the thrust of the rocket motor. By increasing the chamber pressure, it is possible to obtain high thrust levels; however, due to the structural limitations of the motor case and other components, maximum chamber pressure is limited for the design activities. This constraint on chamber pressure is usually named as Maximum Expected Operating Pressure (MEOP).

3.2.3 Thrust, Thrust Coefficient and Total Impulse

The thrust of a SRM is the force produced by a rocket propulsion system acting upon a vehicle as pushing it forward. It is generated by ejecting combustion product gases from the nozzle at a very high velocity. Since thrust time history is the main performance requirement of a SRM for the flight mission, it is the most significant parameter specifying the rocker motor design.

Thrust can be calculated from momentum equation applied on the overall rocket system as given in Equation (3.9).

$$F = \dot{m}V_e + (p_e - p_{amb})A_e \qquad (3.9)$$

where V_e is the exit velocity, p_e is the exit pressure and A_e is the nozzle exit area.

The exit velocity is found from isentropic flow of a perfect gas through nozzles as follows [25]:

$$V_e = \sqrt{\frac{2\gamma}{\gamma - 1} RT_i \left[1 - \left(\frac{p_e}{p_i} \right)^{\frac{\gamma-1}{\gamma}} \right] + V_i^2} \qquad (3.10)$$

where i subscript denotes the nozzle inlet conditions. Nozzle inlet conditions can be taken as the chamber conditions. Since the velocity in the chamber is very small compared to velocities in the nozzle, V_i^2 term can be neglected. Then, Equation (3.10) becomes as Equation (3.11).

$$V_e = \sqrt{\frac{2\gamma}{\gamma - 1} RT_c \left[1 - \left(\frac{p_e}{p_c} \right)^{\frac{\gamma-1}{\gamma}} \right]} \tag{3.11}$$

Substituting Equations (3.6) and (3.11) into Equation (3.9), thrust becomes:

$$F = A_t p_c \left[\sqrt{\frac{2\gamma^2}{\gamma - 1} \left(\frac{2}{\gamma + 1} \right)^{\frac{\gamma+1}{\gamma-1}} \left[1 - \left(\frac{p_e}{p_c} \right)^{\frac{\gamma-1}{\gamma}} \right]} + \frac{(p_e - p_{amb})}{P_c} \frac{A_e}{A_t} \right] \tag{3.12}$$

Then, it is possible to define the thrust with thrust coefficient C_F as:

$$F = C_F p_c A_t \tag{3.13}$$

where

$$C_F = \sqrt{\frac{2\gamma^2}{\gamma - 1} \left(\frac{2}{\gamma + 1} \right)^{\frac{\gamma+1}{\gamma-1}} \left[1 - \left(\frac{p_e}{p_c} \right)^{\frac{\gamma-1}{\gamma}} \right]} + \frac{(p_e - p_{amb})}{p_c} \frac{A_e}{A_t} \tag{3.14}$$

Physically, C_F is an expression for efficiency of the nozzle for a fixed propellant configuration. Because of loss mechanisms in nozzle flow like two-phase flow loss, divergence loss, boundary layer loss, kinetic loss, nozzle erosion and submerge nozzle [27], theoretical C_F found in Equation (3.14) differs from the delivered one which can be calculated easily from the measured data by Equation (3.13). Then, C_F efficiency is defined as:

$$\eta_{C_F} = \frac{C_{F_{del}}}{C_{F_{theo}}} \tag{3.15}$$

The thrust force integrated over the burning time t_b is named as total impulse. Generally, the performance requirement of a SRM for a specific flight mission is defined with total impulse instead of thrust-time history.

$$I_t = \int_0^{t_b} F dt \tag{3.16}$$

3.2.4 Nozzle Expansion Ratio

Nozzle expansion ratio ε is defined as the ratio of nozzle exit area to the nozzle throat area as given below. It is an important nozzle parameter used in the thrust calculations since it determines flow properties at the nozzle exit.

$$\varepsilon = \frac{A_e}{A_t} \tag{3.17}$$

3.3 INTERNAL BALLISTICS SOLVER

For the performance prediction of rocket motor, internal ballistics solver which is developed by Açık [13] is used in this study. The program uses 0-D quasi-steady model for the flow in combustion chamber. In the nozzle, steady 1-D isentropic flow equations are governed. The program calculates the pressure-time and thrust-time history of the rocket motor, with inputs of web versus burn area data, thermo chemical properties of the propellant, nozzle dimensions and performance efficiencies.

3.3.1 Assumptions

Main assumptions of the program are as follows:

1. The combustion gases are perfect gases.
2. The properties of the gases are spatially constant throughout the combustion chamber.
3. Variations of internal ballistic parameters with time are evaluated as the evolution of temporal point wise steady state conditions.
4. Burning rate of the propellant obeys Vielle's burn rate law: $r_b = ap_c^n$
5. Effects of transient mass addition and erosive burning can be neglected.
6. Steady 1-D isentropic flow exists throughout the nozzle.

3.3.2 Governing Equations

3.3.2.1 Equations for Chamber Flow

When the velocity of the flow in the rocket chamber is very low; that is Mach number in the chamber is nearly zero, it can be assumed that the properties of gases are constant along the grain length.

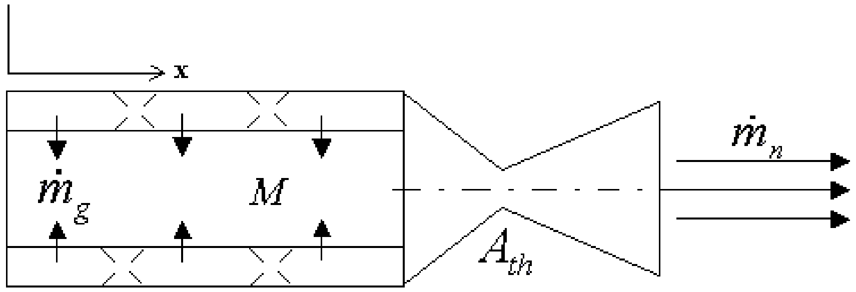


Figure 3.1 Zero-dimensional SRM Conservative Relations [20]

The conservation of mass for isentropic flow can be written as

$$\dot{m}_g = \frac{dM}{dt} + \dot{m}_n \tag{3.18}$$

where $\dot{m}_g$ is the rate of mass addition by burning propellant, *dM/dt* is the rate of change of stored mass in the combustion chamber and $\dot{m}_n$ is the rate of mass flow through the nozzle.

The rate of mass generation is calculated by the relation:

$$\dot{m}_g = \rho_p A_b r_b = \rho_p A_b a p_c^{\,n} \tag{3.19}$$

where ρ_p is the propellant density and A_b is the burn area of the grain at that instant.

The mass flow through the nozzle is calculated by the relation:

$$\dot{m}_n = \frac{p_c A_t}{c^*} \tag{3.20}$$

The throat area may change due to erosion of the nozzle insulation material. Calculated throat erosion rate from static firing tests can be given as an input to the program.

The rate of change of mass stored in the chamber is given as:

$$\frac{\mathrm{d}M}{\mathrm{d}t}=\frac{\mathrm{d}(\rho\forall)}{\mathrm{d}t}=\rho\frac{\mathrm{d}\forall}{\mathrm{d}t}+\forall\frac{\mathrm{d}\rho}{\mathrm{d}t} \tag{3.21}$$

where ρ is the gas density and $\forall$ is the gas volume.

The thermal equation of state for a perfect gas is:

$$p_c=\rho R T_c \tag{3.22}$$

Combining these equations, Equation (3.18) can be rewritten as:

$$\frac{\mathrm{d}p_c}{\mathrm{d}t}=\frac{1}{\forall}\left[RT_c\left(\rho_p A_b a p_c{}^n-\frac{p_c A_t}{c^*}\right)-p_c\frac{\mathrm{d}\forall}{\mathrm{d}t}\right] \tag{3.23}$$

where $\frac{\mathrm{d}\forall}{\mathrm{d}t}=r_b A_b$. Equation (3.23) is integrated with infinitesimal time steps as the propellant burns in order to obtain chamber pressure at these infinitesimal time steps.

3.3.2.2 Equations for Nozzle Flow

For the isentropic flow process in nozzle, the following relations hold between any points *x* and *y*:

$$\frac{T_x}{T_y}=\left(\frac{p_x}{p_y}\right)^{\frac{\gamma-1}{\gamma}}=\left(\frac{\upsilon_y}{\upsilon_x}\right)^{\gamma-1} \tag{3.24}$$

where *T* is the temperature, *p* is the pressure and *υ* is the specific volume.

The stagnation (total) temperature, T_0 is found from energy equation as:

$$T_0=T+\frac{V^2}{2c_p}=T\left(1+\frac{1}{2}(\gamma-1)M^2\right) \tag{3.25}$$

where c_p is the specific heat under constant pressure and *M* is the mach number.

The relationship of the stagnation pressure, p_0 to the local pressure in the flow can be found from the previous two equations as

$$p_0=p\left(1+\frac{1}{2}(\gamma-1)M^2\right)^{\frac{\gamma}{\gamma-1}} \tag{3.26}$$

The nozzle inlet conditions can be taken as chamber conditions. In addition, chamber pressure and temperature are assumed to be equal to total pressure and total temperature. The nozzle throat conditions is found by using Equations (3.25) and (3.26) by setting *M* = 1 at the throat.

Nozzle exit pressure is calculated by using Equation (3.27) which is derived from Equations (3.11), (3.24), (3.25) and (3.26) 1 as

$$\frac{A_t}{A_e} = \frac{V_e \rho_e}{V_t \rho_t} = \left[\frac{\gamma+1}{2}\right]^{\frac{1}{\gamma-1}} \left[\frac{p_e}{p_c}\right]^{\frac{1}{\gamma}} \sqrt{\frac{\gamma+1}{\gamma-1}\left[1-\left[\frac{p_e}{p_c}\right]^{\frac{\gamma-1}{\gamma}}\right]} \tag{3.27}$$

Using the chamber and nozzle exit pressure, the thrust coefficient C_F is calculated by using Equation (3.14).

Lastly, the thrust is calculated by the following relation,

$$F(t) = \eta_{C_F} C_F P_c A_t \tag{3.28}$$

CHAPTER 4

GENETIC ALGORITHMS

4.1 INTRODUCTION

Using the optimization philosophy, one approaches a complex decision problem, involving the selection of values for a number of interrelated variables, by focusing attention on a single objective designed to quantify performance and measure the quality of the decision [28]. In other words, optimization is the process of maximizing or minimizing a desired objective function, while satisfying the concerned constraints.

In engineering designs, there are an abundance of examples where the optimum system is sought and with the passing of years, lots of optimization methods are developed to answer the question *'Is this design the best one?'*. To apply an optimization method for a practical problem, firstly, the optimization algorithms should be understood, since the problems require tuning algorithmic parameters, scaling and even modifying existing techniques to suit the specific application [29]. Within this scope, firstly, optimization methods in literature are summarized in this chapter. Then, the theory of genetic algorithms and developed genetic optimizer code (GENOP) is presented in details. Lastly, the optimization code is validated with the cases whose solutions are known.

4.2 OPTIMIZATION METHODS

The optimization techniques are generally discussed in the main titles of linear and nonlinear programming. Nonlinear programming deals with the problem of optimizing an objective function in the presence of equality and inequality constraints. If all the functions are linear, the problem is called linear programming problem, otherwise it is called nonlinear programming problem [30]. Many realistic problems cannot be adequately represented or approximated as a linear program, because of the nature of the nonlinearity of the objective function or the constraints. Since the grain design problem is a nonlinear program, instructions in this section are mostly focused on it. Constrained minimization problems can be expressed in the following general nonlinear programming form:

$$\begin{array}{lll} \text{minimize} & f(\mathbf{x}) & \\ \text{subject to} & g_i(\mathbf{x}) \le 0 & i = 1,.....,m \\ & h_j(\mathbf{x}) = 0 & j = 1,.....,l \end{array} \tag{4.1}$$

where $\mathbf{x}=[x_1, x_{2,.....}, x_n]^{\mathrm{T}}$ is a column vector of n real-valued design variables. f is the objective or cost function, g's are inequality constraints and h's are equality constraints.

In this section, optimization methods in literature are briefly introduced in two groups as: gradient-based and derivative-free methods. The aim is only to give a general viewpoint about optimization methods and the place of genetic algorithms among these methods, not to classify every possible type of optimization methods.

4.2.1 Gradient-Based Methods

Gradient-based optimization strategies iteratively search the minimum of objective function. At first, algorithms take a starting point $\mathbf{x}_0$ and calculate the objective function.

Then, the following loop tries to achieve a minimum:

1. Check the convergence criterion of $f(\mathbf{x})$, if it is fulfilled, terminate with $\mathbf{x}_k$ as the solution
2. Compute step length α_k and direction vector $\mathbf{d}_k$ that improves the objective function
3. Evaluate the new point $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$, calculate $f(\mathbf{x}_{k+1})$, go to Step 1

The Steepest Descent Algorithm is the most basic one of the gradient-based methods. This method simply calculates the gradient of a function at each iteration and uses the gradient as a search direction onto the next design point as given in Equation (4.2). Step length α_k is determined by one-dimensional minimization algorithm (line search). This type of gradient-based method is a first order method, because it uses solely the gradient information [31].

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k) \qquad (4.2)$$

Other well-known gradient method, Newton's Method is a second order one. This method is similar to the Steepest Descent method, but it adds second order information to its calculations in the form of Hessian matrix $[\nabla^2 f(\mathbf{x}_k)]$ as given in Equation (4.3).

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left[\nabla^2 f(\mathbf{x}_k)\right]^{-1} \nabla f(\mathbf{x}_k)^T \qquad (4.3)$$

Conjugate Gradient Method, Davidon-Fletcher-Powell (DFP) Method, Broyden-Fletcher-Goldfarb-Shanno (BFGS) Method and Sequential Quadratic Programming are other gradient-based methods. The details of these methods can be found in References [28], [29] and [30].

A major difficulty with gradient-based optimizers is dealing with noisy problems or problems containing many local minima. Given a response surface with various local minima, the algorithms will generally converge to the nearest local minima [31].

4.2.2 Derivative-Free Methods

Derivative-free optimization methods are typically developed to solve optimization problems whose gradient computation of objective function is unavailable. Even if it is possible to estimate derivatives by numerical methods such as finite differences; in most real problems, the objective function is expensive to evaluate, so derivative approximation may be prohibitively costly. In addition, when the optimization problem has a noisy objective function, derivative estimation will not be accurate.

The simplest derivative-free method is the one referred to as Direct Search Methods. These methods sample the objective function at a finite number of points at each iteration and decide which actions to take next solely based on those function values and without any explicit or implicit derivative approximation or model building [32]. Through the years more and more sophisticated logic has been developed to allow these types of algorithms to intelligently search through the design space. Hooke and Jeeves Pattern Search Method, Rosenbrock's Method, Powell's Method of Conjugate Directions, Nelder and Mead Simplex Method, Simulated Anneling, Genetic Algorithms and Box Complex Method are some examples of Direct Search Methods [29].

4.3 GENETIC ALGORITHMS

Genetic algorithms are derivative-free, heuristic, global search methods based on the principles of natural selection and genetics. In 1975, John Holland led to the development of genetic algorithms with his work about the investigation of the mechanisms of natural adaptation [33]. Since that time, they have been widely studied, experimented and applied in many engineering fields.

The basic idea of the algorithms is to mimic the evolution of a group of individuals of the same species. Since the individuals who adapt better to the requirements imposed by their environment will survive in the population, their genes will be passed more frequently to subsequent generations than others. This means, the average fitness of the population increases with time [34]. This basic idea of can be implemented to design optimization problems as follows:

gene	→	encoded design variable
chromosome	→	set of design variables
individual	→	design alternative generated from specified design variables
population	→	group of design alternatives
generation	→	optimization iteration
fitness value	→	scaled value of objective function

Once the design variables are encoded (details given in the next section) in a chromosomal manner and a fitness measure for discriminating good solutions from bad ones has been chosen, genetic algorithms start to evolve solutions by using the following basic scheme [35]:

1. **Initialization:** The initial population of candidate solutions (design alternatives) is usually generated randomly across the search space.

2. **Evaluation:** Once the population is initialized or an offspring population (created from main population) is created, the fitness values of the candidate solutions are evaluated.

3. **Selection:** Selection allocates more copies of those solutions with higher fitness values (better designs) into mating pool and thus imposes the survival of the fittest mechanism on the candidate solutions. The main idea of selection is to prefer better solutions to worse ones, and many selection procedures have been proposed to accomplish, including roulette-wheel selection, stochastic universal selection, ranking selection and tournament selection.
4. **Recombination (Crossover):** Recombination combines parts of two or more parental chromosomes to create new, possibly better design alternatives (i.e. offspring). There are many ways of accomplishing this, and competent performance depends on properly designed recombination mechanism.
5. **Mutation:** While recombination operates on two or more parental chromosomes, mutation locally but randomly modifies chromosome by changing genes. Mutation performs a random walk in the vicinity of a candidate solution.
6. **Replacement:** The offspring population created by selection, recombination, and mutation replaces the original parental population. Many replacement techniques such as elitist replacement, generation-wise replacement and steady state replacement methods can be used.
7. Steps 2 to 6 are repeated until a terminating condition is met.

The implementation details of the genetic algorithm operators are explained in the next chapter.

4.4 GENETIC OPTIMIZER (GENOP)

An optimization code, called GENOP, using genetic algorithms is developed in FORTRAN during this thesis study. The flowchart of the code is given in Figure 4.1.

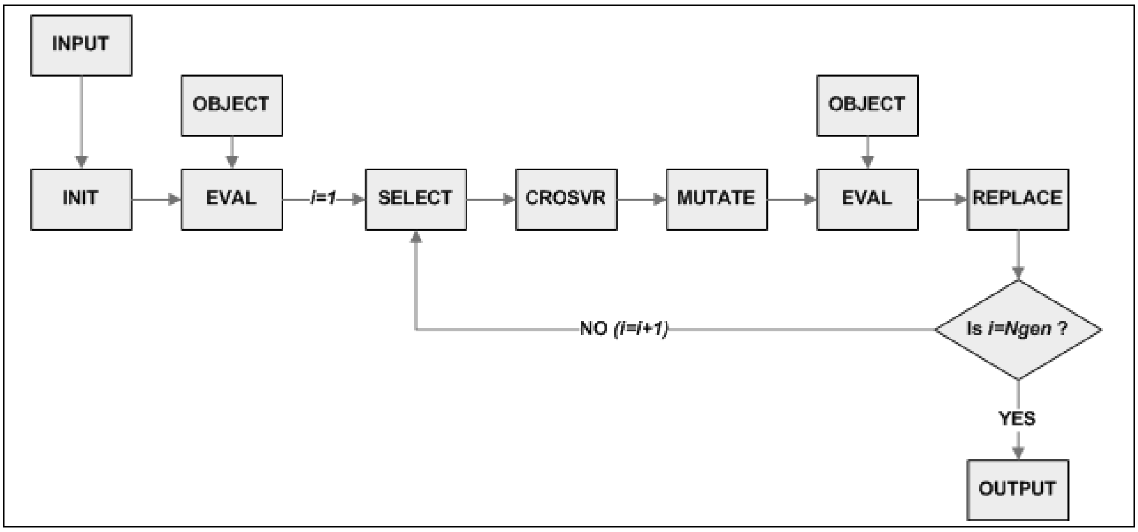


Figure 4.1 Flowchart of GENOP

GENOP, firstly, reads the input parameters of the genetic algorithm which are the number of design variables, upper and lower limits of the variables, bit number, population size, maximum generation number, crossover probability, mutation probability, selection method, crossover method and replacement method. After creating and evaluating the first population, the iteration procedure is started. At each iteration, SELECT, CROSVR, MUTATE and EVAL subroutines are called until the iteration is equal to the maximum generation number. Finally, the best solution in the last generation is given as output. The details of the subroutines are explained in the following sections.

4.4.1 INIT Subroutine

In this subroutine, initial population is created randomly in encoded form. In order to apply genetic operators like crossover and mutation, the design variables of each individual are encoded as a binary number: which is a string x_i^1 .... $x_i^{N_b}$ of zeros and ones where N_b denotes the length of string (number of bits). The feasible interval of variable x_i is divided into $2^{N_b} - 1$ intervals, and variable can be represented by any of the discrete representations as follows:

$$000000, 000001, 000010, 000011, \ldots\ldots, 111111 \qquad when\ N_b = 6 \tag{4.4}$$

The real value of the variable is calculated as given in Equation (4.5):

$$x_i = l_i + \frac{(u_i - l_i)}{2^{N_b} - 1} \cdot \sum_{j=1}^{N_b} \left(2^{j-1} x^j\right) \tag{4.5}$$

where **l** is a vector of length *n* (number of variables) containing lower bounds on variables and **u** is a vector of length *n* containing upper bounds on variables.

As can be seen in Equation (4.5), number of bits determines the precision of variables while searching the design space.

In order to create a random variable, RNUNF function in FORTRAN library is used. RNUNF function creates a uniform random real number between 0 and 1 by using a seed number to initialize its random number generator. If the generated random number is greater than 0.5, the string of the variable is filled with 1, otherwise with 0. To generate the initial population, a string of size ($n \cdot N_b$) bits is created for each individual of the population. The encoded variables of the individual are attached end to end as given in Figure 4.2. N_p is the size of the population which defines the number of individuals in the population.

$$\textit{member } 1 \quad \underbrace{101101}_{x_1}\underbrace{101001}_{x_2}\underbrace{001010}_{x_3}\ldots\underbrace{101111}_{x_n}$$

$$\textit{member } 2 \quad \underbrace{101001}_{x_1}\underbrace{100001}_{x_2}\underbrace{001110}_{x_3}\ldots\underbrace{101001}_{x_n} \qquad \textbf{\textit{where }} \boldsymbol{N_b = 6}$$

$$\ldots$$

$$\textit{member } N_p \quad \underbrace{101001}_{x_1}\underbrace{101101}_{x_2}\underbrace{101010}_{x_3}\ldots\underbrace{101100}_{x_n}$$

Figure 4.2 Creation of the Initial Population [29]

4.4.2 EVAL Subroutine

In this subroutine, firstly, the encoded form of the variables is decoded by using Equation (4.5). Then, the OBJECT function, where the objective (cost) function of the optimization problem is calculated, is called.

4.4.3 SELECT Subroutine

SELECT subroutine creates copies of individuals into mating pool for crossover operation by using one of the following methods.

4.4.3.1 Roulette-Wheel Selection

In roulette-wheel selection, each individual in the population is assigned a roulette wheel slot sized in proportion to its fitness value. Thus, good solutions have a larger slot size than the less fit solutions [35]. The method was implemented as follows:

1. Evaluate the fitness, f_i of each individual in the population.
2. Compute the probability (slot size), p_i, of selecting each member of the population as given in Equation (4.6) where N_p is the population size.

$$p_i = \frac{f_i}{\sum_{j=1}^{N_p} f_j} \qquad (4.6)$$

3. Calculate the cumulative probability, q_i for each individual: $q_i = \sum_{j=1}^{i} p_j$
4. Generate a random number, $r \in [0,1]$
5. If $r < q_1$, then select the first chromosome, x_1, else select the individual x_i such that $q_{i-1} < r \leq q_i$.
6. Repeat steps 4-5 N_p times to create N_p candidates in the mating pool.

4.4.3.2 Tournament Selection

In tournament selection, two individuals are selected randomly and entered into tournament against each other. The fittest individual wins the tournament and is selected into mating pool.

4.4.4 CROSVR Subroutine

In CROSVR subroutine, two individuals are selected from the mating pool and recombined (crossed over) with a probability P_c, called the crossover probability. That is, a random number, r is generated in the range 0 to 1 and if $r \leq P_c$, selected two individuals are crossed over with one of the crossover operators; *one-point* or *two-point*. Otherwise, if $r > P_c$, the two offspring are simply copies of their parent individuals.

4.4.4.1 One-Point Crossover

If one-point crossover operator is selected by the user, the subroutine generates a random integer k between 1 and $(n \cdot N_b - 1)$ and the chromosomes of the parents are truncated from the k point and the truncated parts of the chromosomes are exchanged between the individuals to create new offspring individuals as given in Figure 4.3.

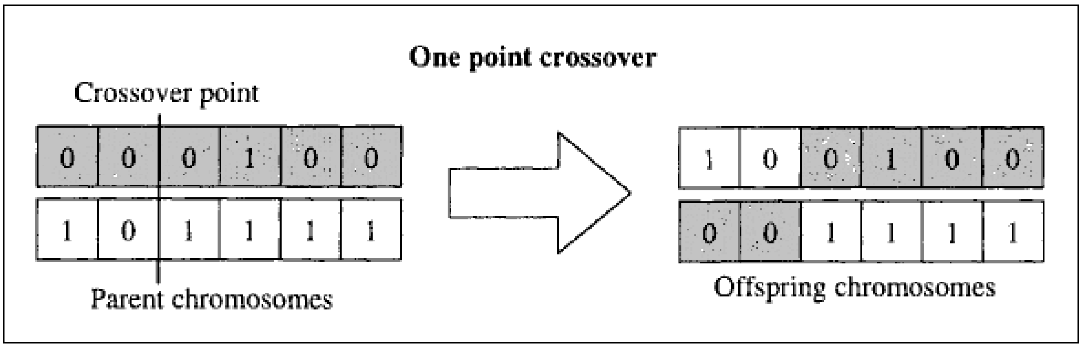


Figure 4.3 One-Point Crossover Operation

4.4.4.2 Two-Point Crossover

When the two-point crossover operator is selected by the user, the subroutine generates two random integers as given in former operator in order to determine the crossover site, and the truncated parts of the chromosomes are exchanged as illustrated in Figure 4.4.

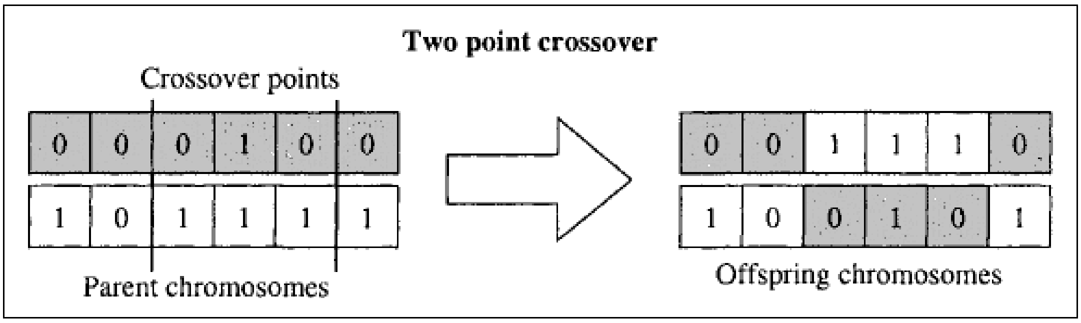


Figure 4.4 Two-Point Crossover Operation

4.4.5 MUTATE Subroutine

After the crossover operation, it is possible to get identical individuals since the crossed over part of the parental chromosomes can be same. Mutation is designed to overcome this problem in order to add diversity to the population and ensure that it is possible to explore the entire search space [35].

In order perform the mutation operation, MUTATE subroutine visits every bit of the chromosome of each individuals and changes the related bit with a probability P_m, called the mutation probability. That is, a random number, r is generated in the range 0 to 1, and if $r \leq P_m$, selected bit is changed to 0 if it is 1 and 1 if it is 0 as given in Figure 4.5. In the example given in Figure 4.5; 1st, 4th and 6th bits are mutated.

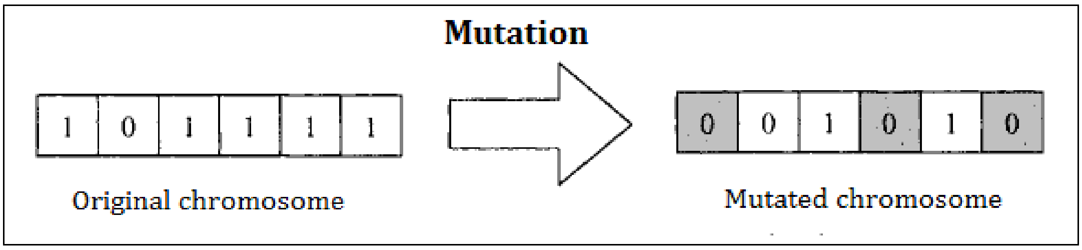


Figure 4.5 Mutation Operation

4.4.6 REPLACE Subroutine

Once the new offspring population is created using selection, crossover and mutation, it is introduced into the parental population. At this stage, the aim is increasing the average fitness value of the parental population. Therefore, the worse individuals among the parents and children are eliminated until the population size decreases to its initial size. The user can select one of the following methods.

4.4.6.1 Best Alive

In this method, firstly, parents and children are collected into a population and arranged according to their fitness values. Then, the best ones are selected in order to form the new parental population.

4.4.6.2 Elitism

In this method, the specified number of worst individuals of the offspring population is replaced with the best individuals of parent population. In this study, the elitism number is chosen as the half of the population size, $N_p/2$. In other words, the worst half of the children is changed with the best half of the parents.

4.5 VALIDATION

GENOP is validated with the analytical functions whose global minimum points are known. For the test cases; sphere, Rastrigin's and Rosenbrock's functions are taken as the objective function, which are commonly used as optimization test problems in literature. The search domain for each variable is taken as [-5, 5] and bit number of each variable is determined as 26 in order to approximate the exact solution with 6 decimal places, since the increment range while changing the value of variable will be $(5-(-5))/(2^{26}-1) \approx 1.5 \text{ x } 10^{-7}$. For the random number generator, the seed value is used as 123457 for each case.

4.5.1 Test Case-1

For the first test case; sphere function is selected in order to work with a simple, convex function. The optimization problem is given as follows:

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) = x_1^2 + x_2^2 \\ &\text{subject to} && -5 \le \mathbf{x} \le 5 \quad \mathbf{x} \in \mathbf{R}^2 \end{aligned} \tag{4.7}$$

The objective function is plotted in Figure 4.6 for the specified search domain. As seen in the figure, the function has no local minimum points and the global minimum point is $\mathbf{x}^* = (0,0)$ where the function value is $f(\mathbf{x}^*) = 0$.

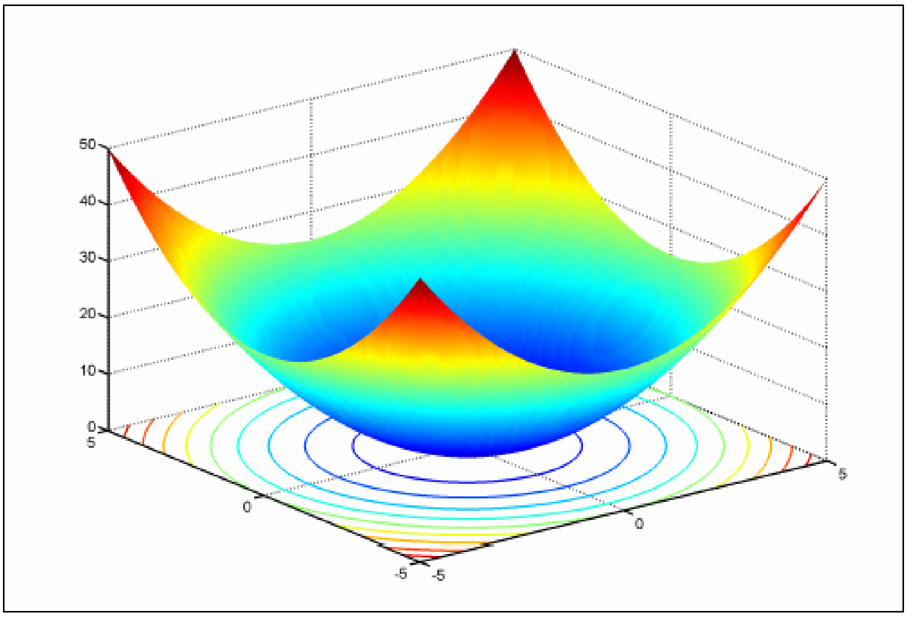


Figure 4.6 Sphere Function

The parameters of the genetic algorithm used for the problem solution are given in Table 4.1.

Table 4.1 Parameters of Genetic Algorithm of Test Case-1

$\boldsymbol{N_b}$	26	***Selection Method***	Roulette-Wheel
$\boldsymbol{N_p}$	100	***Crossover Method***	One-Point
$\boldsymbol{N_{gen}}$	40	***Replacement Method***	Best-Alive
$\boldsymbol{P_c}$	0.8		
$\boldsymbol{P_m}$	0.03		

The global minimum point is found after 37 generations. The best and average fitness values at each generation are plotted at the left in Figure 4.7. In the figure, the location of the best individual found at each generation can also be seen at the right side. The results are also given in tabulated form with the range of 5 generations in Table 4.2. When the results are analyzed, it is easily seen that the genetic algorithm finds the approximate place of the solution after few generations; however, it takes time to find the exact solution.

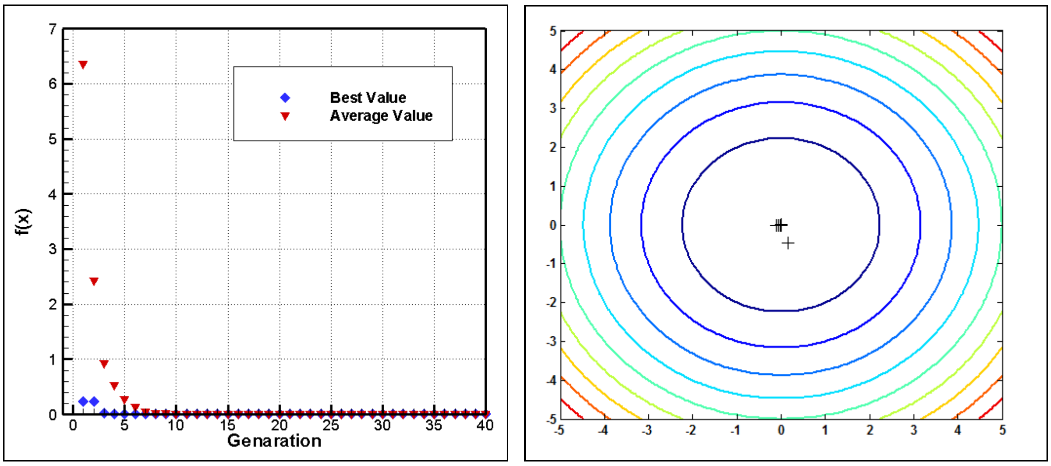


Figure 4.7 Best Fitness Values (left) and Best Individual (right) at Each Generation

Table 4.2 Results of Test Case-1 Solution

Generation Number	Best Individual			Average Fitness
	x(1)	x(2)	f(x)	
1	0.152480	-0.460208	0.235041	6.351674
5	-0.065109	-0.014900	0.004461	0.274413
10	-0.006211	-0.000836	0.000039	0.005981
15	-0.000021	-0.000261	0.000000	0.000091
20	-0.000021	-0.000260	0.000000	0.000005
25	-0.000020	-0.000059	0.000000	0.000000
30	-0.000007	-0.000001	0.000000	0.000000
35	-0.000001	-0.000001	0.000000	0.000000
40	0.000000	0.000000	0.000000	0.000000

4.5.2 Test Case-2

Rastrigin's function is selected for the second test case. The function is based on sphere function; however, with the addition of cosine modulation, regularly distributed many local minima are produced. The optimization problem is given as follows:

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) = 20 + x_1^{\,2} + x_2^{\,2} - 10\left(\cos(2\pi x_1) + \cos(2\pi x_2)\right) \\ &\text{subject to} && -5 \le \mathbf{x} \le 5 \quad \mathbf{x} \in \mathbf{R}^2 \end{aligned} \tag{4.8}$$

The objective function is plotted in Figure 4.8 for the specified search domain. As seen in the figure, the function has several local minimum points and the global minimum point is $\mathbf{x}^* = (0,0)$ where the function value is $f(\mathbf{x}^*) = 0$.

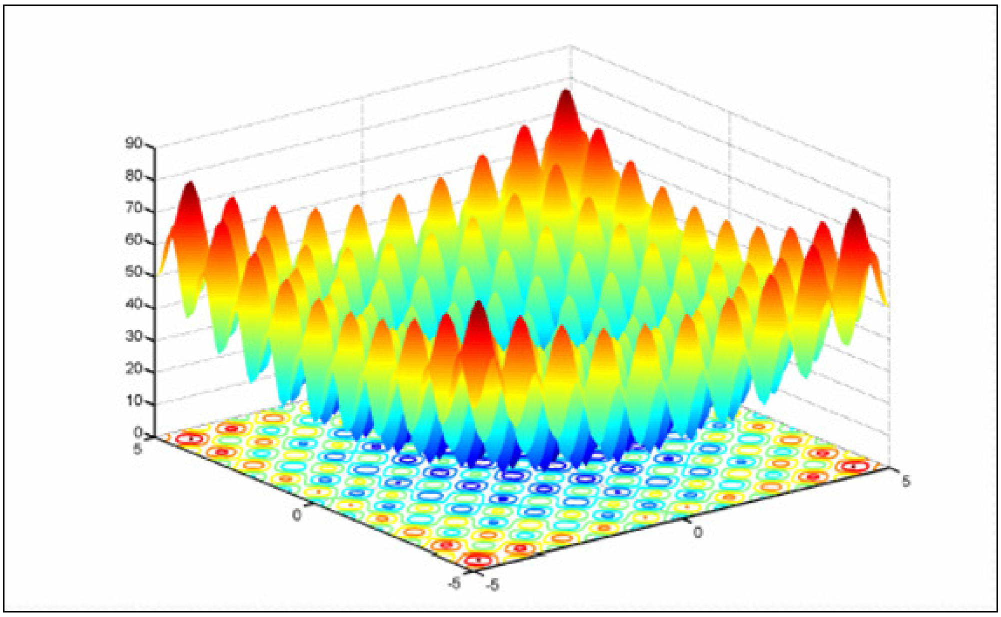


Figure 4.8 Rastrigin's Function

The parameters of the genetic algorithm used for the problem solution are given in Table 4.3.

Table 4.3 Parameters of Genetic Algorithm of Test Case-2

N_b	26	***Selection Method***	Tournament
N_p	100	***Crossover Method***	Two-Point
N_{gen}	40	***Replacement Method***	Elitism
P_c	0.8		
P_m	0.02		

The global minimum point is found after 38 generations. The best and average fitness values at each generation are plotted at the left in Figure 4.9. In the figure, the location of the best individual found at each generation can also be seen at the right side. The results are also given in tabulated form with the range of 5 generations in Table 4.4. When the results are examined, it is seen that the genetic algorithm firstly captures a local minimum point (0,1) and after 10 generations, it jumps the approximate place of the solution.

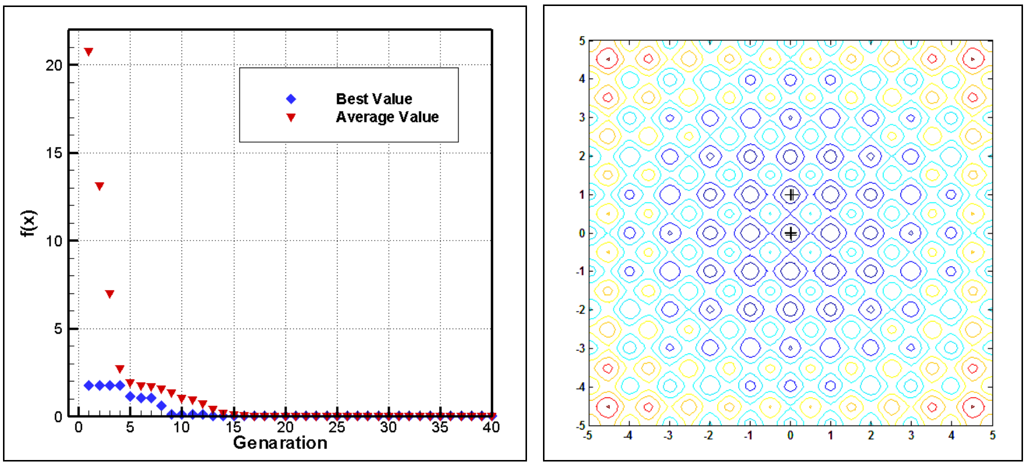


Figure 4.9 Best Fitness Values (left) and Best Individual (right) at Each Generation

Table 4.4 Results of Test Case-2 Solution

Generation Number	Best Individual			Average Fitness
	x(1)	x(2)	f(x)	
1	0.061736	0.986431	1.756130	20.785984
5	0.022674	0.986431	1.111191	1.915870
10	0.020026	0.009869	0.098778	1.035250
15	0.000312	0.000098	0.000021	0.081605
20	0.000009	0.000097	0.000002	0.000024
25	0.000006	0.000020	0.000000	0.000002
30	0.000006	0.000003	0.000000	0.000000
35	0.000002	0.000001	0.000000	0.000000
40	0.000000	0.000000	0.000000	0.000000

4.5.3 Test Case-3

For the third test case, Rosenbrock's function is selected. The global minimum point of the function is inside a long, narrow, parabolic shaped flat valley. Since convergence to the global minimum point is difficult, this problem has been repeatedly used for the performance of optimization algorithms. The optimization problem is given as follows:

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) = 100 \cdot \left(x_1^{\ 2} - x_2\right)^2 + \left(x_1 - 1\right)^2 \\ \text{subject to} & -5 \le \mathbf{x} \le 5 \qquad \mathbf{x} \in \mathbf{R}^2 \end{array} \tag{4.9}$$

The objective function is plotted in Figure 4.10 for the specified search domain. At right side, the fitness value is plotted in logarithmic scale in order to show clearly the valley where the global optimum is located. As seen in the figure, the function has several local minimum points and the global minimum point is $\mathbf{x}^* = (1,1)$ where the function value is $f(\mathbf{x}^*) = 0$.

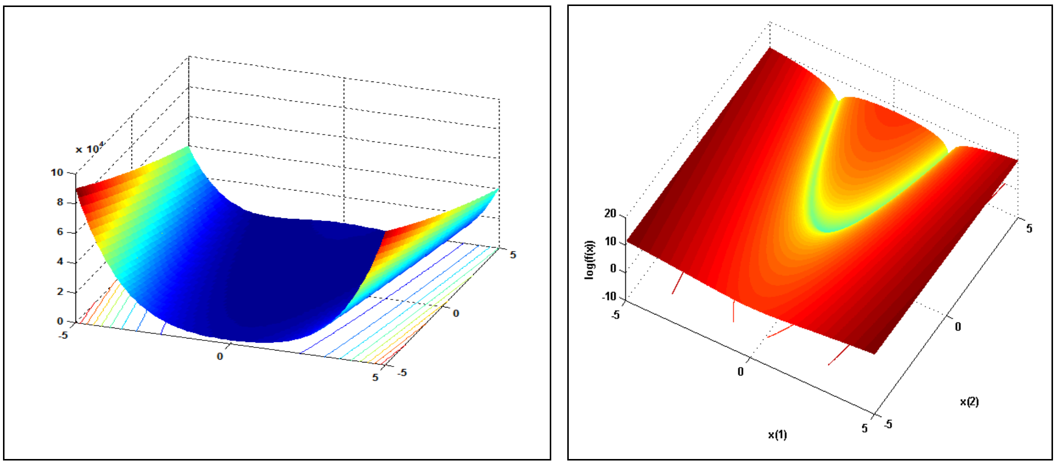


Figure 4.10 Rosenbrock's Function

The parameters of the genetic algorithm used for the problem solution are given in Table 4.5. The population and maximum generation number are increased in order to find the global optimum.

Table 4.5 Parameters of Genetic Algorithm of Test Case-3

N N_b	26	***Selection Method***	Roulette-Wheel
N_p	1000	***Crossover Method***	Two-Point
N_{gen}	100	***Replacement Method***	Elitism
P_c	0.8		
P_m	0.02		

The global minimum point is found after 94 generations. The best and average fitness values at each generation are plotted at the left in Figure 4.11.

In the figure, the locations of the best individual found at each generation are given at the right side. The results are also given in tabulated form with the range of 10 generations in Table 4.6. When the results are analyzed, after 50 generations, the algorithm gets closer to optimum point; however, until 94th generation, it travels in the vicinity of the exact solution.

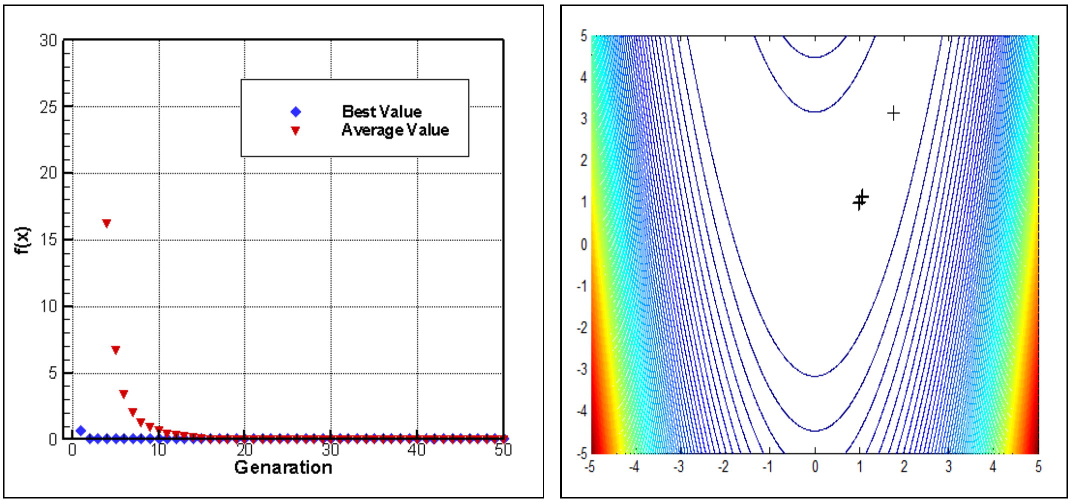


Figure 4.11 Best Fitness Values (left) and Best Individual (right) at Each Generation

Table 4.6 Results of Test Case-3 Solution

Generation Number	Best Individual			Average Fitness
	x(1)	x(2)	f(x)	
1	1.770535	3.151014	0.620034	1044.455227
10	1.069104	1.145117	0.005231	0.687646
20	1.005628	1.007566	0.001416	0.006257
30	0.999543	0.998962	0.000002	0.002764
40	0.999512	0.998963	0.000001	0.000166
50	1.000033	1.000067	0.000000	0.000015
60	0.999997	0.999994	0.000000	0.000009
70	0.999997	0.999994	0.000000	0.000000
80	0.999999	0.999999	0.000000	0.000000
90	1.000000	1.000001	0.000000	0.000000
100	1.000000	1.000000	0.000000	0.000000

CHAPTER 5

VALIDATION OF GRAIN DESIGN OPTIMIZATION TOOL

The main aim of this thesis study is to develop a ballistic design optimization tool for 3-D grain configuration. For this purpose, a grain design optimization tool, called GDOT, is developed which consists of three main modules which are grain geometry modeling and burnback, internal ballistic solver and genetic optimizer. GDOT is actually the integration of BB3D and SOLVER codes into GENOP code which are presented in previous chapters. The flowchart of GDOT is given in following figure.

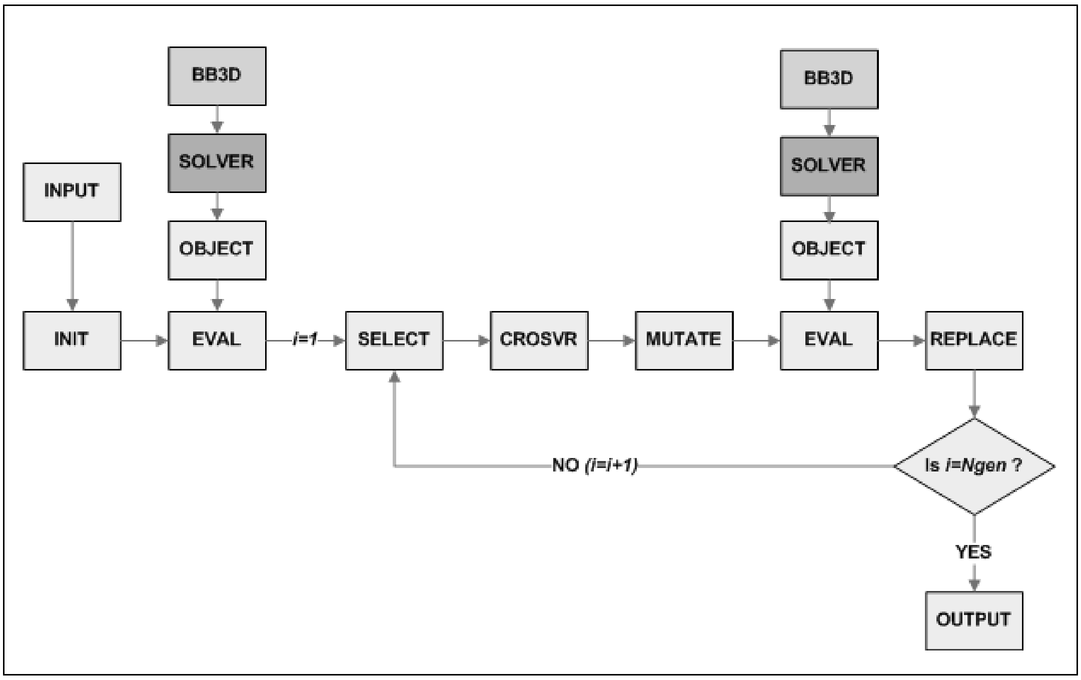


Figure 5.1 Flowchart of GDOT

GDOT uses genetic algorithm for finding the best solution in design space. Within the geometric constraints, firstly the design variables are selected randomly and grain designs are developed in the amount of population number.

Then, for the each grain design, the given steps are followed successively:

- “Burning surface area versus web increment” data is calculated with burnback analysis (BB3D).
- Ballistic performance parameters are found with internal ballistic solver by using the burnback data.
- Objective function (fitness value) is calculated by using the ballistic results.

After the initial design population is generated, the optimization iteration starts. At each iteration (generation), a new design population is developed in order to find the best design by using genetic operators (selection, crossover, mutation and replacement). The iteration continues until the stopping criterion is satisfied.

In order to run optimization tool, user should prepare the following inputs:

- Inputs of BB3D defining the grain configuration by using basic figures or grain modules
- Inputs of SOLVER consisting of propellant and nozzle properties
- Inputs of GENOP consisting of design constraints and optimization parameters
- Objective function in OBJECT subroutine

5.1 TEST CASES

In this section, three test cases are studied in order to validate the grain design optimization tool. For the first test case, an actual, predesigned Star Motor having a convex star grain configuration is used. For the second case, again a real rocket motor having a finocyl grain with slots at head and aft ends is examined. Lastly, a radial slot grain studied in literature is selected as a design optimization problem. During the study, a computer with Intel Core i5 2.67 GHz CPU and 4 GB RAM is used.

5.1.1 Test-Case 1

A simple 2-D grain configuration having less design variables than 3-D complex ones is selected as a starting case. For this purpose, the grain of a real solid rocket motor, called Star Motor (SM), which has a 2-D star convex grain configuration with 6 star points and was previously designed without the aid of any optimization tool by trial and error method, is studied. The off-scale solid model of the grain is presented in Figure 5.2 .

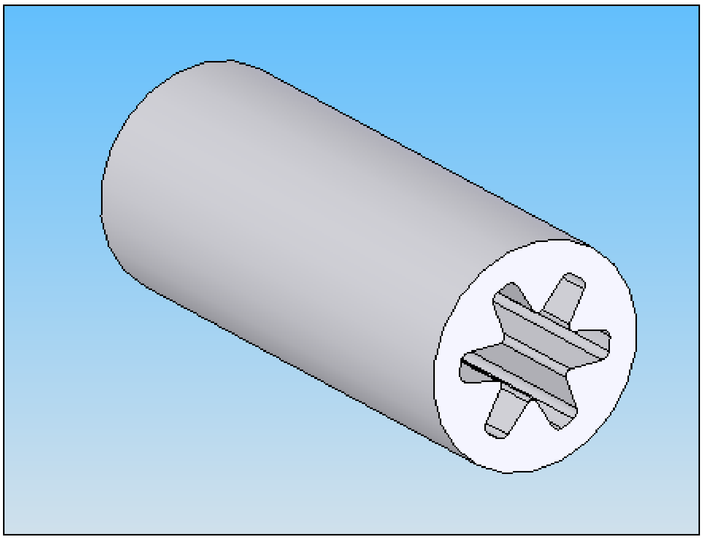


Figure 5.2 Solid Model of the Grain of SM

The aim of this study is to compare the result of developed optimization tool with an actual, predesigned grain using its objectives and constraints. In addition, this grain configuration was studied by Açık in her thesis study [13] with a 2-D grain design optimization tool using complex optimization algorithm. Therefore, it will be possible to evaluate the performance of the developed optimization tool by comparing with the one developed by Açık. In order to make an accurate comparison and avoid the differences between optimization tools while calculating the ballistic performance, 2-D burnback code of Açık's tool is replaced with BB3D. Since the ballistics solver code is same in the optimization tools, actually, the performance of the genetic and complex algorithm is compared by this study.

5.1.1.1 Design Variables

8 design variables and their dimensionless values for Star Motor are given in Table 5.1. Design variables other than grain length and nozzle throat diameter are the ones defining the cross-section geometry as seen in Figure 5.3.

Table 5.1 Nondimesional Geometric Parameters of Star Motor

Grain length, L/D_t	18.8
Outer diameter, D_{out}/D_t	3.55
Web thickness, w/w	1.00
Fillet radius, r_1/w	0.27
Cusp radius, r_2/w	0.18
Star point semi angle, η/w	3.27
Star angle, ξ/w	2.27
Nozzle throat diameter, D_t/D_t	1.00

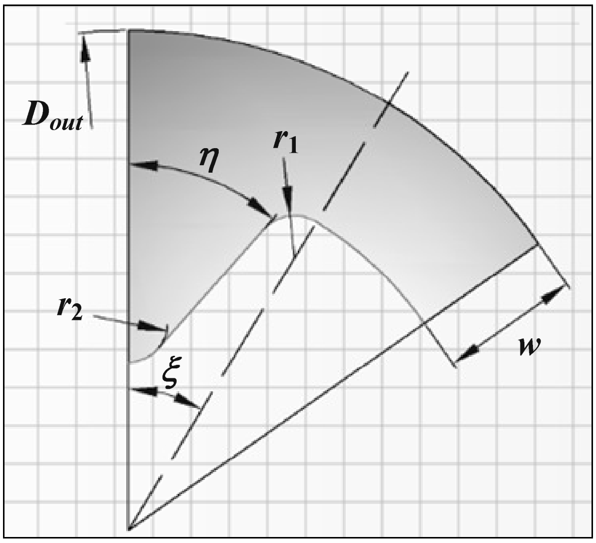


Figure 5.3 Geometric Parameters of Convex Star Grain

In this study, the problem is solved repeatedly with different genetic operators and parameters to find proper optimization parameter set for the grain design problems. Therefore, in order to reduce the computation time, *h* (web increment / maximum web) is taken as about 0.03 and *k* is used as 1.0 for the burnback analysis.

5.1.1.2 Design Constraints

Design constraints are taken as given in Reference [13]. The geometric bounds on the variables are defined as shown in Table 5.2. Values in Table 5.2 are nondimensionalized with the SM grain parameters. Grain length and outer diameter bounds come from the SM design process. However, the bounds of the other parameters were chosen arbitrarily considering manufacturability.

Table 5.2 Nondimensional Geometric Bounds of Star Grain Optimization

Parameter	**Lower Bound**	**Upper Bound**
Grain length, L / L_{SM}	0.96	1.05
Outer diameter, $D_{out} / D_{out,SM}$	0.96	1.02
Web thickness, w / w_{SM}	0.73	1.27
Fillet radius, $r_1 / r_{1,SM}$	0.50	1.50
Cusp radius, $r_2 / r_{2,SM}$	0.75	2.25
Star point semi angle, η / η_{SM}	0.83	1.11
Star angle, ξ / ξ_{SM}	0.80	1.20
Nozzle throat diameter, $D_t / D_{t,SM}$	0.77	1.54

Maximum chamber pressure is constrained to be 3000 psi and propellant mass is constrained to be 4 kg. The propellant and nozzle properties of the actual motor are used as the input of the ballistic solver.

5.1.1.3 Objective Function

Considering the mission requirements of a missile or rocket system, the grain design requirements can be simplified to the thrust-time history of the rocket motor. In this case study, the design requirements of SM like total impulse, burning time and average thrust level are defined as an objective thrust-time curve.

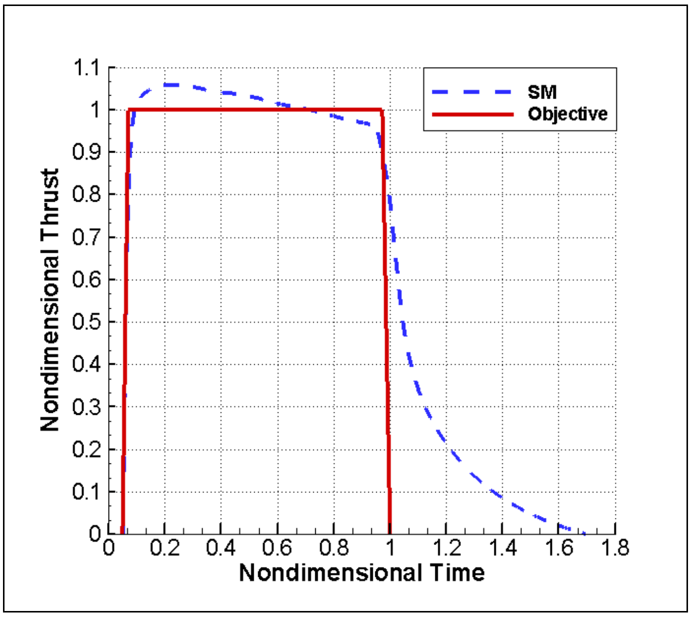


Figure 5.4 Nondimensional Objective and SM Thrust-Time Curves

The objective function and objective thrust-time curve are taken as given in Reference [13]. While specifying the objective thrust-time curve, the total impulse of SM up to 1.0 time value is taken into consideration since, after that point, thrust starts to decrease. This part of the curve, called sliver region, is almost useless due to the inefficient burning of the propellant. Any design having relatively small sliver region is more acceptable. Therefore, average thrust of SM is calculated up to time 1.0 and the remaining part of the curve is discarded.

Implementing this average thrust, the objective thrust-time curve of this test case is defined as in Figure 5.4. In the figure, thrust is normalized with respect to maximum thrust of the objective thrust-time curve and time is normalized with respect to total burning time of the objective thrust-time curve.

Using the objective thrust-time curve, the optimization problem becomes finding the grain design whose ballistic performance fits the objective curve best within the given design constraints. Therefore, the objective function can be defined as follows:

$$f(\mathbf{x}) = \frac{\sqrt{\sum_{i=1}^{N_{tb}} (F_{des\,i}(\mathbf{x}) - F_{obj\,i})^2}}{N_{tb} F_{obj_{ave}}} 100 \qquad (5.1)$$

The objective function is typically summation of differences between the desired (F_{obj}) and computed (F_{des}) thrust values at specified times during motor operation divided by average desired thrust ($F_{obj_{ave}}$) and total number of time data (N_{tb}).

Some constraints like maximum chamber pressure, propellant mass and burning time can be defined in order to discourage unrealistic designs. These constraints are enforced to the objective function by the penalty function method. Main idea of the penalty function method is adding a term to the objective function that gives a high cost when the constraints are violated. Therefore, the following term is added to objective function when the constraints are not satisfied.

$$\left(10 \frac{g(\mathbf{x}) - g_{const}}{g_{const}} + 1 \right)^{10} \qquad (5.2)$$

where *g(*x*)* is the constraint parameter found with related design variables and g_{const} is the upper value of the specified constraint.

Finally, the optimization problem can be defined as:

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & l_i \leq x_i \leq u_i \quad i = 1,...,8 \\ & p_c(\mathbf{x}) \leq 3000 \ \ \text{(psi)} \\ & m_p(\mathbf{x}) \leq 4 \ \ \text{(kg)} \end{array} \qquad (5.3)$$

where **l** is a vector containing the lower bounds and **u** is a vector containing the upper bounds on design variables.

5.1.1.4 Optimization Parameters

After defining the design constraint and objective function, the following studies are done in order to find genetic algorithm parameters of the optimization tool giving the best solution. Considering the number of function evaluations and computation time, the maximum number of generation is taken as 50. Different population numbers as 20, 30 and 40 are used. The bit number is set to 8, which is sufficient for the precision of variables.

5.1.1.4.1 Study-1

In order to find the genetic operators giving the best solution for the studied case; cross-over probability and mutation probability is taken as 0.8 and 0.03, respectively and the test case is studied with all possible selection, cross-over and replacement operators. The used optimization parameters are given in Table 5.3.

Table 5.3 Optimization Parameters of Study-1

N_b	8	***Selection Method***	Roulette-Wheel / Tournament
N_p	20, 30, 40	***Crossover Method***	One-Point / Two-Point
N_{gen}	50	***Replacement Method***	Best Alive / Elitism
P_c	0.8		
P_m	0.03		

The results are presented in Table 5.4. The last column given in the table is the value of objective function of found solution and shows the difference between objective thrust-time curve and thrust-time curve of present solution.

When the table is analyzed, it is seen that larger population number increases the probability of finding best solution and the solutions get closer to each other. However, all solutions with different population numbers are dispersed into a narrow interval about 1 % error band. Therefore, it is reasonable to work with small population number for the cases whose computation time is high.

Table 5.4 Results of Study-1

No	N_p	Selection	Cross-Over	Replacement	Obj. Func. (%)
1	40	Roulette	Two-Point	Elitism	7.35
2	40	Tournament	One-Point	Best Alive	7.39
3	30	Tournament	Two-Point	Elitism	7.41
4	20	Roulette	Two-Point	Elitism	7.43
5	20	Roulette	Two-Point	Best Alive	7.44
6	40	Tournament	Two-Point	Best Alive	7.54
7	40	Roulette	Two-Point	Best Alive	7.56
8	40	Roulette	One-Point	Best Alive	7.58
9	40	Tournament	Two-Point	Elitism	7.61
10	40	Roulette	One-Point	Elitism	7.64
11	20	Roulette	One-Point	Best Alive	7.64
12	20	Roulette	One-Point	Elitism	7.68
13	40	Tournament	One-Point	Elitism	7.72
14	30	Roulette	One-Point	Best Alive	7.73
15	30	Roulette	One-Point	Elitism	7.77
16	30	Roulette	Two-Point	Best Alive	8.14
17	20	Tournament	Two-Point	Best Alive	8.29
18	30	Roulette	Two-Point	Elitism	8.30
19	20	Tournament	Two-Point	Elitism	8.37
20	20	Tournament	One-Point	Elitism	8.40
21	30	Tournament	One-Point	Elitism	8.42
22	30	Tournament	One-Point	Best Alive	8.42
23	30	Tournament	Two-Point	Best Alive	8.43
24	20	Tournament	One-Point	Best Alive	8.43

Examining the table, it is hard to say that a specific genetic operator gives better solution. For a specified population number, the best working operator triad is changing. However, roulette-wheel/two-point/elitism operator triad gives the best solution in the population number of 20 and 40.

5.1.1.4.2 Study-2

After finding the best working operators for the studied case with Study-1, the effect of the cross-over probability and mutation probability are investigated with this study. For this purpose, roulette-wheel/two-point/elitism operator triad is selected and the population number of 20 is used to decrease the computation time. The crossover probability is changed with 0.1 increments between 0.1 and 1.0. On the other hand, the mutation probability is changed with 0.01 increments between 0.01 and 0.10. The optimization parameters used in this study are given in Table 5.5.

Table 5.5 Optimization Parameters of Study-2

N_b	8	***Selection Method***	Roulette-Wheel
N_p	20	***Crossover Method***	Two-Point
N_{gen}	50	***Replacement Method***	Elitism
P_c	0.1-1.0		
P_m	0.01-0.10		

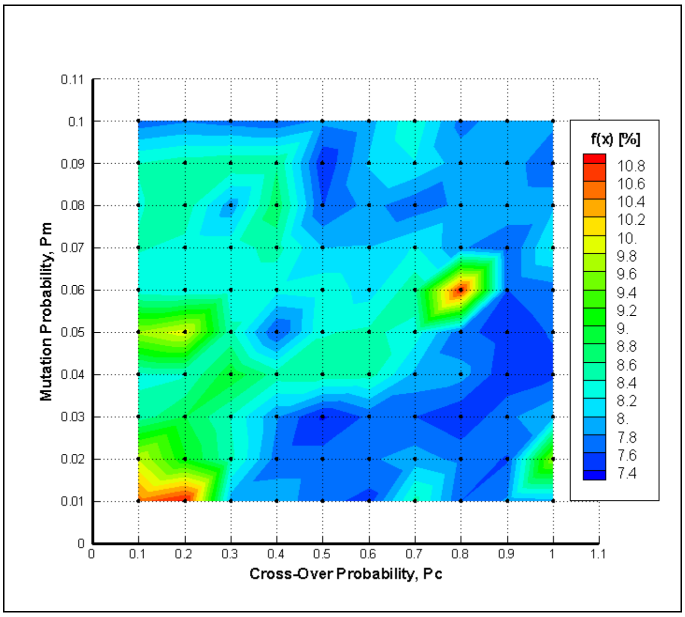


Figure 5.5 Results of Study-2

The contour plot of the objective function value for different cross-over and mutation probabilities is presented in Figure 5.6. As seen from the figure, when P_c is increased, P_m should decrease in order to find better solutions. On the other hand when P_c is decreased, Pm should increase for better performance. Because using high cross-over and mutation probabilities increases the diversity of the population so much that converging to an optimal solution can be impossible. Therefore, there should be a balance between cross-over and mutation.

The best solution is found as 7.37 % with $P_c = 0.5$ and $P_m = 0.03$ and the worst solution is get as 10.91 % with $P_c = 0.2$ and $P_m = 0.01$. The study shows that the value of the objective function can change about 3.5 % with the change of cross-over and mutation probabilities. However, working in the region of $0.5 \leq P_c \leq 0.9$ and $0.01 \leq P_m \leq 0.05$ decreases the change of the objective function value to the level of 1 %. Therefore, the solution found by using the optimization parameters as roulette-wheel/two-point/elitism operator triad with $P_c = 0.8$ and $P_m = 0.03$ is reliable for this engineering application.

After examining the results of Study-1 and Study-2, the optimization parameters for this test case are used, as shown in the following table.

Table 5.6 Optimization Parameters of Test Case-1

N_b	8	***Selection Method***	Roulette-Wheel
N_p	40	***Crossover Method***	Two-Point
N_{gen}	50	***Replacement Method***	Elitism
P_c	0.8		
P_m	0.03		

5.1.1.5 Results

In this section, the best solution found with the developed optimization tool using genetic algorithm is compared with the actual SM geometry and found solution with Açık's tool using complex algorithm. After the implementation of BB3D into Açık's tool, optimization algorithms are the only difference between Açık's tool and GDOT. Therefore, performance of the genetic algorithm is compared with complex algorithm in the given results. To make an accurate comparison, the maximum number of function evaluations is taken as 2040 for the both algorithms. Optimization process with the specified number of function evaluations takes about one hour for the both algorithms.

In the genetic algorithm, objective function is evaluated in the number of population size at each generation; therefore, in this study, objective function is calculated 40 times at each generation. Average and best objective function value (fitness value) of the population at each generation of genetic algorithm is given in Figure 5.6. The objective function value of the found geometry after 50 generation is 7.4 %.

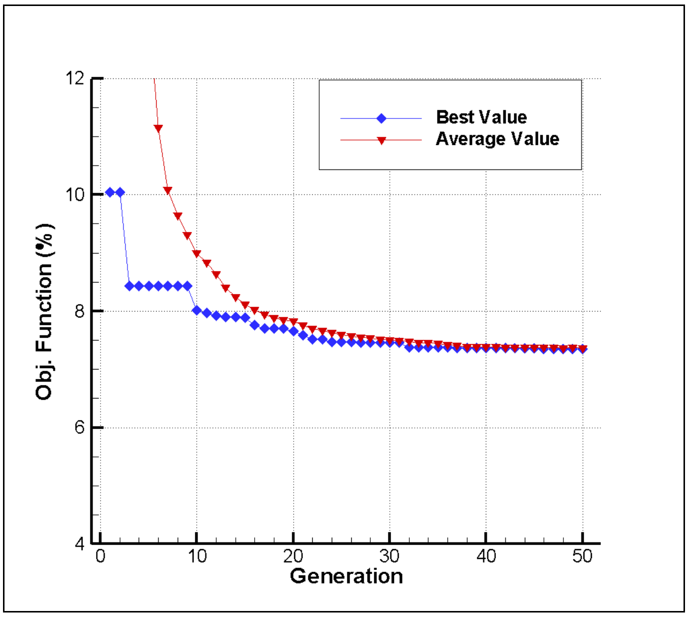


Figure 5.6 Best and Average Fitness Values at Each Generation of Genetic Algorithm

Fitness value evolution with respect to number of function evaluations for genetic and complex algorithms is given in Figure 5.7. Objective function of the found geometry by using complex algorithm is 8.4 %. As seen from the figure, complex algorithm converges to the found solution after 500 number of function evaluations. On the other hand, genetic algorithm improves the solution up to the value of 7.4 % until 1320 number of function evaluations.

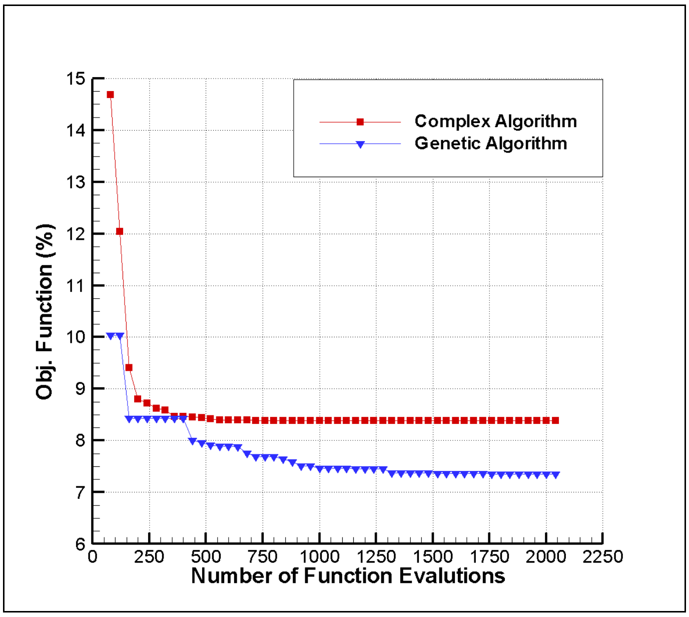


Figure 5.7 Fitness Value Evolution of Genetic and Complex Algorithms

Table 5.7 Nondimensional Results for Star Grain

Parameters	**SM**	**Complex Algorithm**	**Genetic Algorithm**
Grain length, L / L_{SM}	1.00	1.03	1.00
Outer diameter, $D_{out} / D_{out,SM}$	1.00	1.00	0.97
Web thickness, w / w_{SM}	1.00	0.93	1.10
Fillet radius, $r_1 / r_{1,SM}$	1.00	0.85	1.13
Cusp radius, $r_2 / r_{2,SM}$	1.00	1.50	0.76
Star point semi angle, η / η_{SM}	1.00	0.93	0.97
Star angle, ξ / ξ_{SM}	1.00	0.80	0.80
Throat diameter, $D_t / D_{t,SM}$	1.00	1.05	0.85
Objective Function (%)	11.2	8.4	7.4
Propellant mass, $m_p / m_{p,SM}$	1.00	0.90	0.86
Max. chamber pressure, $p_{max} / p_{max,SM}$	1.00	0.88	1.30
Total impulse, $I_t / I_{t,SM}$	1.00	0.97	0.99

The nondimensional results are presented in Table 5.7. Objective function of the actual SM geometry is calculated as 11.2 % for the given objective thrust-time curve. Therefore, with the genetic algorithm, a star grain geometry, having a better curve fit with the objective thrust-time curve than actual SM geometry and found geometry by using complex algorithm, is found. In addition, the maximum chamber pressure is increased in the solution found by genetic algorithm, since the throat diameter is decreased. However, this change make available to achieve nearly same total impulse with the actual motor design by a grain design with 14 % less propellant mass.

The cross-sectional views of the found grain geometries are presented in Figure 5.8. The geometry with blue color is the actual SM geometry. The red one is the geometry found by genetic algorithm and the green one is the geometry found by complex algorithm. The light blue area is the design space on the cross-section geometry bounded with the given constraints.

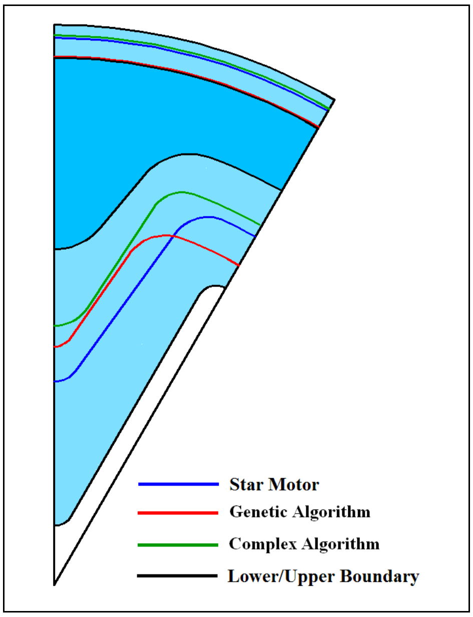


Figure 5.8 Cross-Sectional View of Grain Geometries

Nondimensional thrust-time curve of the present solution is given in the following figure. As seen from the figure, after 1.0 time value, the solution found by genetic algorithm has smaller sliver region than other grains.

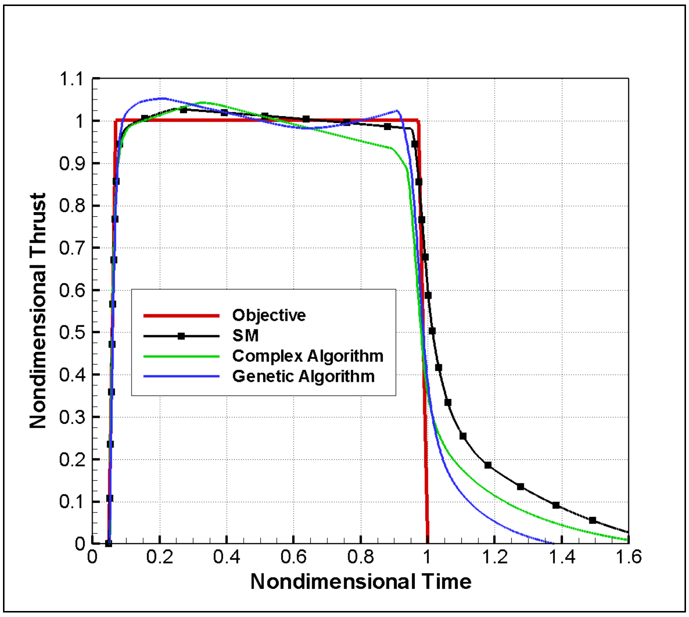


Figure 5.9 Nondimensional Thrust-Time Curve of the Present Solution

Since the complex algorithm takes an initial design point to start the optimization process, this affects the found solution of the algorithm. When the initial point is given nearer to the found solution by the genetic algorithm, complex algorithm finds the similar results with the genetic one. In this study, initial point for the complex algorithm is taken as given in Reference [13].

The results show that complex algorithm converged to a local optimum point in the search space of this grain design problem with the selected initial point. On the other hand, it is not possible to say that genetic algorithm is superior to the complex one by comparing the results of only this test case. Since coming such a conclusion is not the aim of this study, complex algorithm is not studied for the other test cases.

5.1.2 Test-Case 2

In this case study, the grain of a real solid rocket motor, called Boost-Sustain Motor (BSM), which has a 3-D finocyl grain configuration with 8 fins at the fore and aft ends, is studied. BSM was previously designed without the aid of any optimization tool like Star Motor given in Test-Case 1. The off-scale solid model of the grain is presented in Figure 5.10. In the burnback analysis, h is taken as about 0.01 and k is used as 0.5 and 1.0 for the radial and axial station increments respectively.

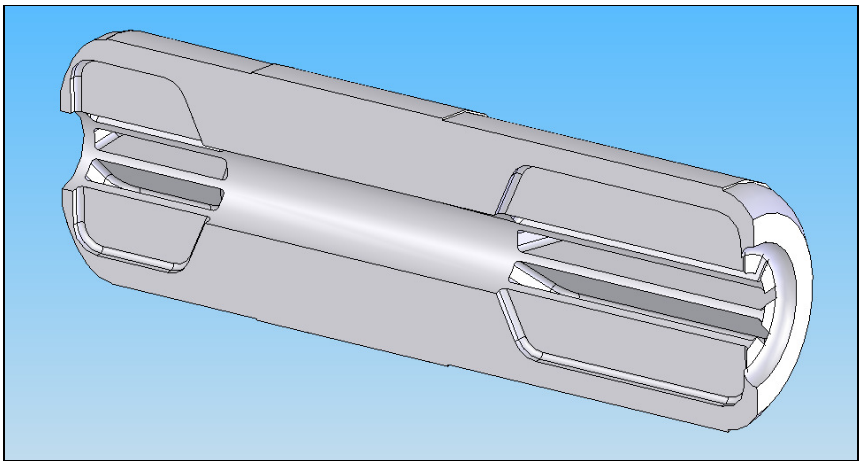


Figure 5.10 Solid Model of the Grain of BSM

5.1.2.1 Design Variables

19 geometric parameters, which are the grain length, port diameter, 8 variables of each slots and nozzle throat diameter, are selected as the design variables. The design variables are presented in Figure 5.11. The outer diameter, ellipsoid ratios of the dome parts, fore and aft openings of the grain are not changed, since they are the design constraints of BSM.

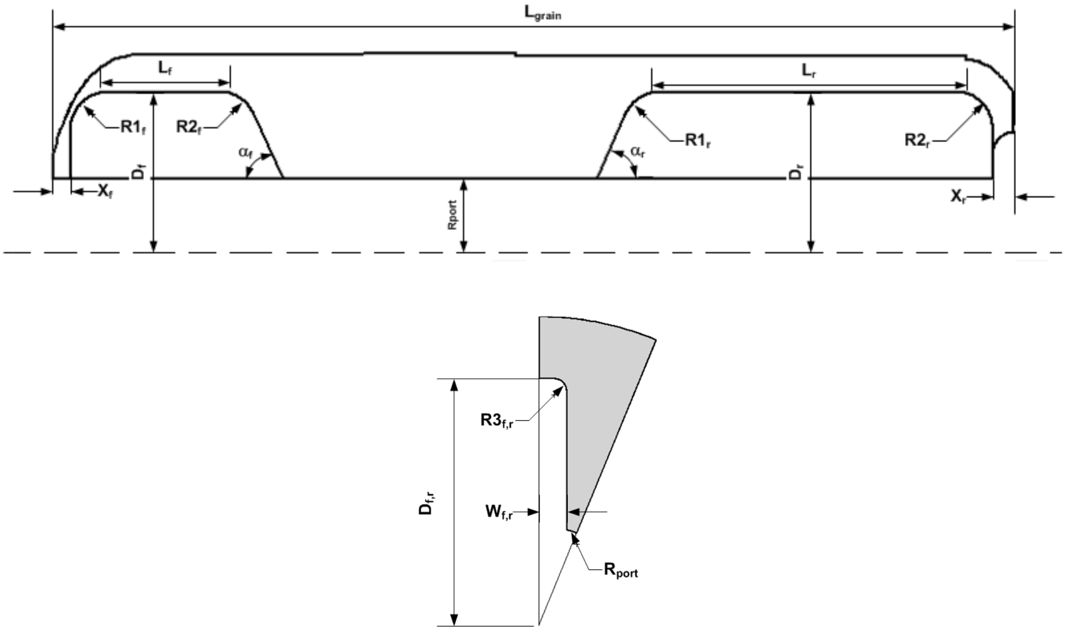


Figure 5.11. Design Variables of Finocyl Grain

5.1.2.2 Design Constraints

The nondimensional geometric bounds on the variables are defined, as shown in Table 5.8. A larger design space is created with the given geometric bounds. In this way, the performance of the optimization tool is experienced with large number of design variables and large design space.

Maximum chamber pressure is constrained to be 1300 psi. Maximum propellant mass is not taken as a design constraint. The propellant and nozzle properties of the actual motor are used as the input of the ballistic solver.

Table 5.8 Nondimensional Geometric Bounds of Finocyl Grain Optimization

No	**Parameter**	**Lower Bound**	**Upper Bound**
1	Grain length, L / L_{SM}	0.96	1.02
2	Port diameter, $D_{port} / D_{port,BSM}$	0.80	1.22
3	Distance of front slot, $X_f / X_{f,BSM}$,	0.71	1.43
4	Length of front slot, $L_f / L_{f,BSM}$	0.54	1.61
5	Depth of front slot, $D_f / D_{f,BSM}$	0.85	1.19
6	Width of front slot, $W_f / W_{f,BSM}$	0.57	1.13
7	Radius-1 of front slot, $R1_f / R1_{f,BSM}$	0.33	1.67
8	Radius-2 of front slot, $R2_f / R2_{f,BSM}$	0.33	1.67
9	Radius-3 of front slot, $R3_f / R3_{f,BSM}$	0.42	1.67
10	Transition angle of front slot, $\alpha_f / \alpha_{f,BSM}$	0.30	1.21
11	Distance of rear slot, $X_r / X_{r,BSM}$,	0.59	1.47
12	Length of rear slot, $L_r / L_{r,BSM}$	0.41	1.64
13	Depth of rear slot, $D_r / D_{r,BSM}$	0.85	1.19
14	Width of rear slot, $W_r / W_{r,BSM}$	0.57	1.13
15	Radius-1 of rear slot, $R1_r / R1_{r,BSM}$	0.33	1.67
16	Radius-2 of rear slot, $R2_r / R2_{r,BSM}$	0.33	1.67
17	Radius-3 of rear slot, $R3_r / R3_{r,BSM}$	0.42	1.67
18	Transition angle of rear slot, $\alpha_r / \alpha_{r,BSM}$	0.30	1.21
19	Nozzle throat diameter, $D_t / D_{t,SM}$	0.88	1.09

5.1.2.3 *Objective Function*

The design requirements of BSM are defined as an objective thrust-time curve given in Figure 5.12 in nondimensional format. Thrust is normalized with respect to maximum thrust of the objective thrust-time curve and time is normalized with respect to total burning time of the objective thrust-time curve. While specifying the objective thrust-time curve, average thrust of boost and sustain levels, total impulse and total burning time of BSM are taken into consideration.

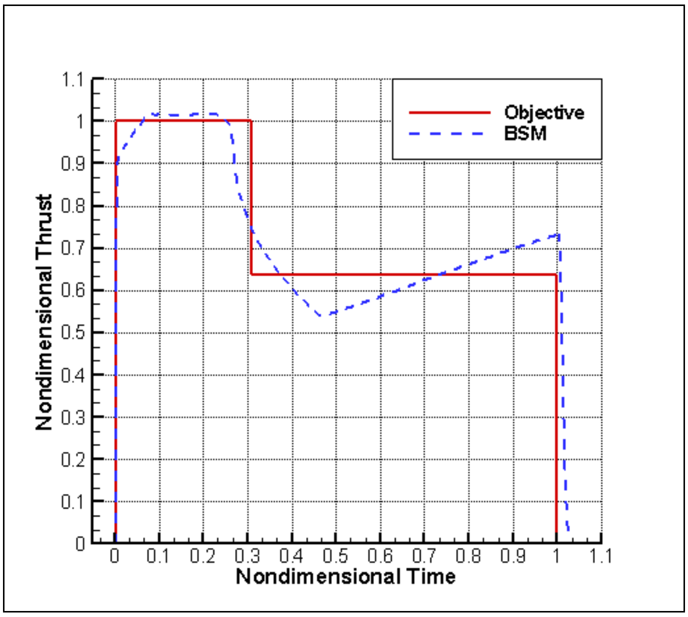


Figure 5.12 Nondimensional Objective and BSM Thrust-Time Curves

An objective function similar to the one in Test Case-1 is defined. In order to calculate the difference between objective and optimized curve accurately in dual thrust case, Equation (5.1) is changed as given in Equation (5.4). Penalty term for the constraint of maximum pressure is added to objective function as given in Equation (5.2).

$$f(\mathbf{x}) = \frac{\sum_{i=1}^{N_{tb}} \sqrt{\left(\left(F_{des\,i}(\mathbf{x}) - F_{obj_i}\right) t_{inc}\right)^2}}{\sum_{i=1}^{N_{tb}} \left(F_{obj_i} t_{inc}\right)} 100 \qquad (5.4)$$

The optimization problem becomes:

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) \\ &\text{subject to} && l_i \le x_i \le u_i \quad i = 1,\ldots,19 \\ & && p_c(\mathbf{x}) \le 1300 \ \text{(psi)} \end{aligned} \qquad (5.5)$$

where **l** is a vector containing the lower bounds and **u** is a vector containing the upper bounds on design variables.

5.1.2.4 Optimization Parameters

Optimization parameters are taken as shown in Table 5.9. Small population number is used to decrease the number of function evaluations since the computation time of each design is about 1 minute. Considering the total number of function evaluations is 1020 for N_p = 20 and N_{gen} = 50, optimization process takes about 15 hours.

Table 5.9 Optimization Parameters of Test Case-2

N_b	8	***Selection Method***	Roulette-Wheel
N_p	20	***Crossover Method***	Two-Point
N_{gen}	50	***Replacement Method***	Elitism
P_c	0.8		
P_m	0.03		

5.1.2.5 *Results*

Average and best objective function value (fitness value) of the population at each generation is given in Figure 5.13. The objective function value of the found geometry after 50 generation is 3.66 %.

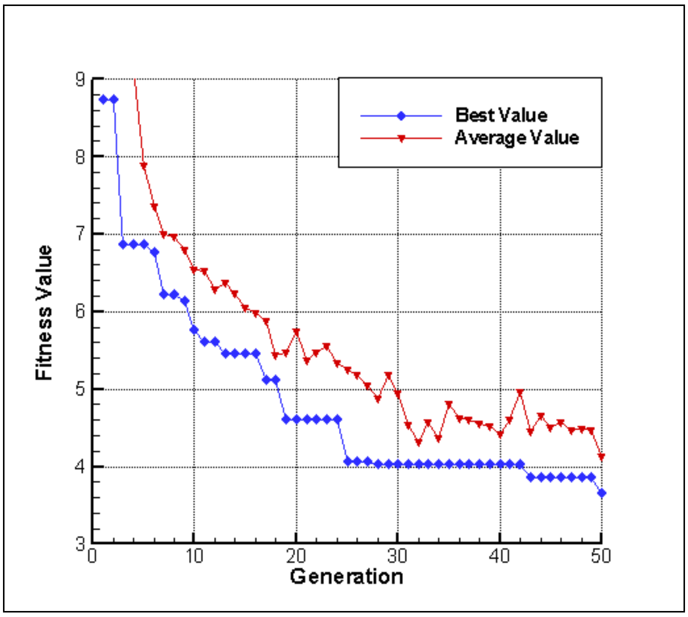


Figure 5.13 Best and Average Fitness Values at Each Generation

The best solution found with the developed optimization tool is shown in Table 5.10. Objective function of the actual BSM geometry is calculated as 7.9 % for this objective thrust-time curve. With the developed optimization tool, a finocyl grain geometry having about 4 % better curve fit with the objective thrust-time curve than actual motor is found.

Table 5.10 Nondimensional Results for Star Grain

Parameter	**Present Solution**
Grain length, L / L_{SM}	1.01
Port diameter, $D_{port} / D_{port,BSM}$	0.99
Distance of front slot, $X_f / X_{f,BSM}$,	3.00
Length of front slot, $L_f / L_{f,BSM}$	0.86
Depth of front slot, $D_f / D_{f,BSM}$	1.00
Width of front slot, $W_f / W_{f,BSM}$	1.02
Radius-1 of front slot, $R1_f / R1_{f,BSM}$	0.34
Radius-2 of front slot, $R2_f / R2_{f,BSM}$	1.10
Radius-3 of front slot, $R3_f / R3_{f,BSM}$	0.60
Transition angle of front slot, $\alpha_f / \alpha_{f,BSM}$	0.31
Distance of rear slot, $X_r / X_{r,BSM}$,	1.31
Length of rear slot, $L_r / L_{r,BSM}$	0.68
Depth of rear slot, $D_r / D_{r,BSM}$	0.95
Width of rear slot, $W_r / W_{r,BSM}$	1.06
Radius-1 of rear slot, $R1_r / R1_{r,BSM}$	0.61
Radius-2 of rear slot, $R2_r / R2_{r,BSM}$	1.58
Radius-3 of rear slot, $R3_r / R3_{r,BSM}$	0.48
Transition angle of rear slot, $\alpha_r / \alpha_{r,BSM}$	0.31
Nozzle throat diameter, $D_t / D_{t,SM}$	0.99
Propellant mass, $m_p / m_{p,BSM}$	1.01
Maximum chamber pressure, $p_{max} / p_{max,BSM}$	1.02
Total impulse, $I_t / I_{t,BSM}$	1.02

In the present solution, the radius-1, radius-3 and transition angles of both slots are decreased; the distances of the slots are increased. In addition, the length of the rear slot is decreased, as shown in Figure 5.14 and Figure 5.15.

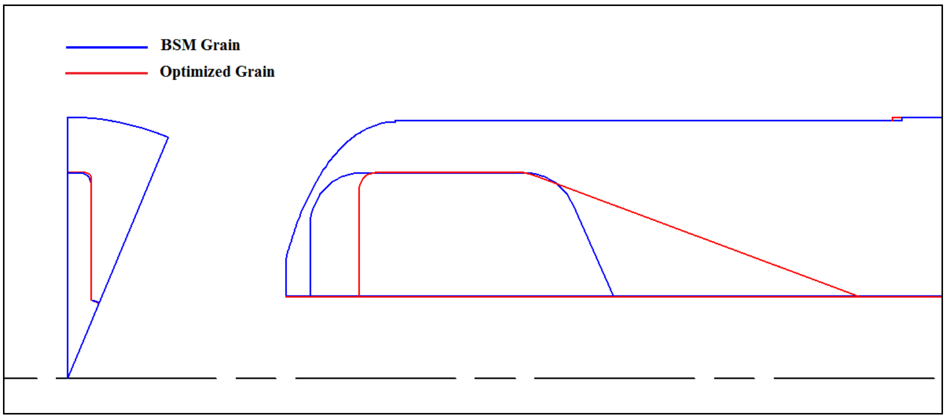


Figure 5.14 Cross-Sectional View of Present Solution at the Head End

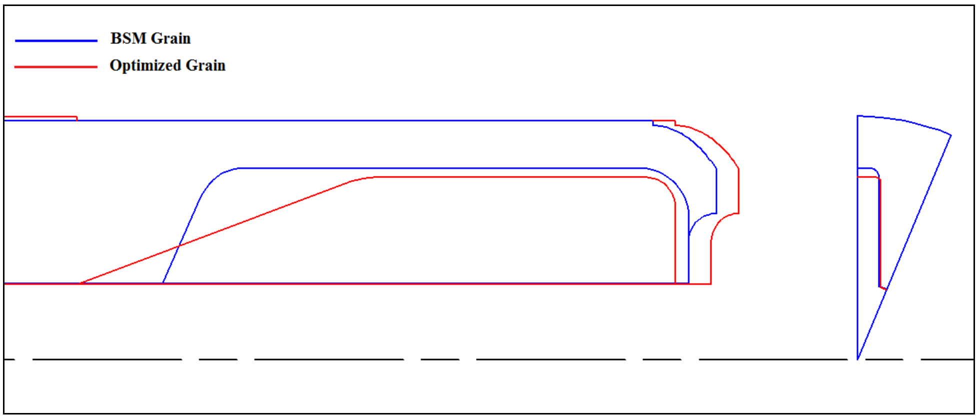


Figure 5.15 Cross-Sectional View of Present Solution at the Aft End

Thrust-time curve of the present solution is given in Figure 5.16. As seen from the figure, the solution has neutral burning in the sustain part as desired with objective thrust-time curve, while the actual motor has progressive burning. This is the main reason why the present solution fits better to the objective curve.

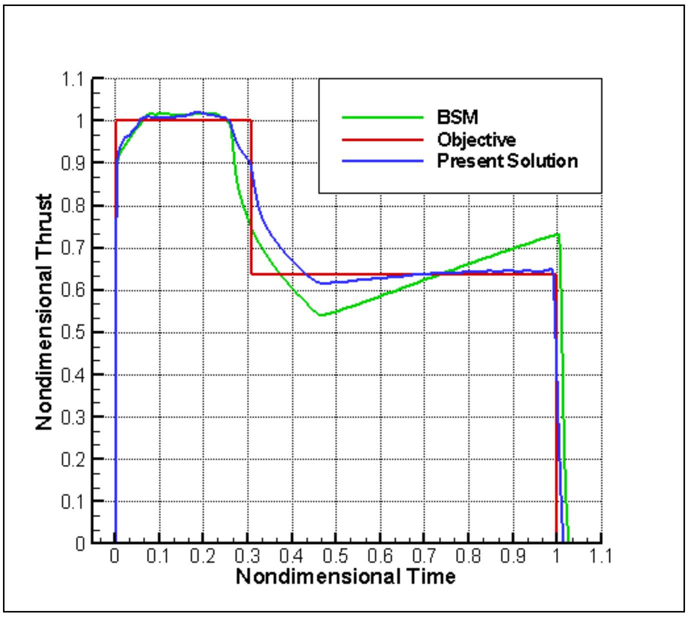


Figure 5.16 Nondimensional Thrust-Time Curve of the Present Solution

5.1.3 Test-Case 3

In Reference [12], 3-D radial slot grain presented in Figure 5.17 is designed using a genetic algorithm. The geometric modeling and burnback analysis of the grain are done with CAD software through dynamic variables which define the complex configuration. Equilibrium pressure method is applied to calculate the internal ballistics. The study in Reference [12] is used in this test case. In the burnback analysis, *h* is taken as about 0.01 and *k* is used as 0.5 and 1.0 for the axial and radial station increments respectively.

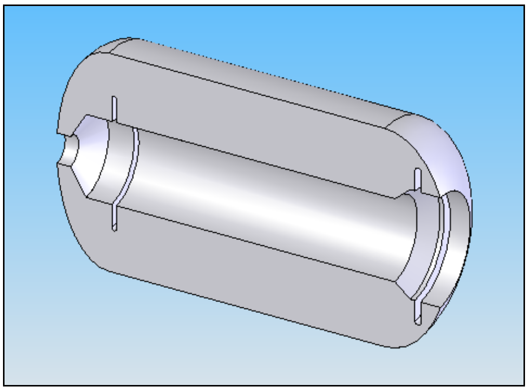


Figure 5.17 Solid Model of Radial Slot Grain

Design variables, constraints, objective function are taken as given in the reference and the results are compared. In order to make a correct comparison, the internal ballistics solver is simplified, as given in reference. Steady-state chamber pressure is calculated by equating the mass generated in chamber to the mass ejected through nozzle throat as follows:

$$p_c = \left(\rho_p a c^* \frac{A_b}{A_t} \right)^{\frac{1}{1-n}} \tag{5.6}$$

Thrust is determined by Equation (3.28) where η_{C_F} is assumed as 0.93 and vacuum conditions are used. Propellant and nozzle parameters used in the study are given in Table 5.11.

Table 5.11 Propellant and Nozzle Parameters

Parameter	**Unit**	**Value**
D_t	mm	160
ε	-	16
$c*$	m/s	1550
ρ_p	kg/m^3	1750
n	-	0.34
a	mm/(s.Pa)	0.0311
γ	-	1.2

5.1.3.1 Design Variables

The geometric parameters of front/rear radial slot grain are shown in Figure 5.18.

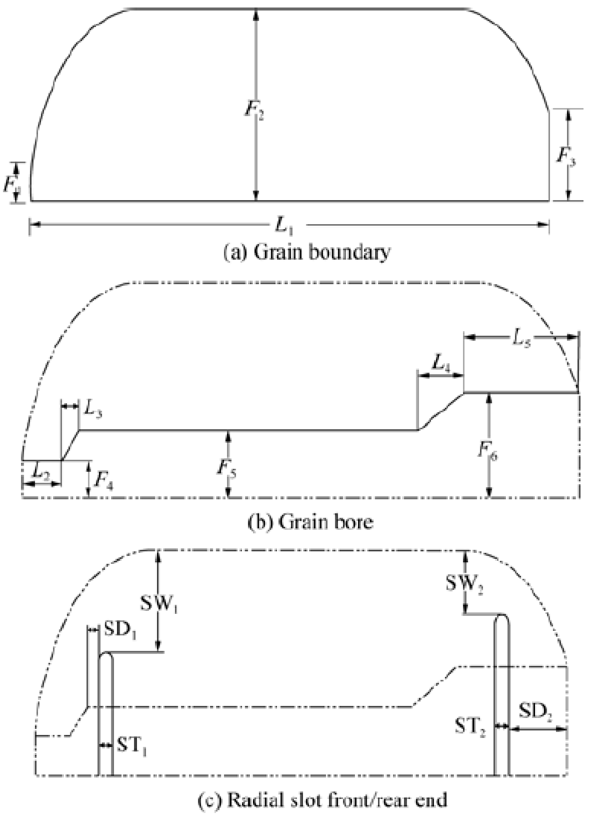


Figure 5.18 Geometric Variables of Radial Slot Grain

Ellipsoid ratio at domes is 2:1. L_1 and F_2 are taken as 2395 mm and 700 mm respectively. While modeling the grain boundary, F_2 is assumed as equal to F_4 and F_3 is equal to F_6. Other 13 parameters are determined as design variables.

5.1.3.2 Design Constraints

The geometric bounds on the variables are defined as shown in Table 5.12. In addition, maximum chamber pressure is constrained to be 65 bar. Maximum propellant mass is taken as 5100 kg. The burning duration is set as 71 s $\leq t_b \leq$ 77 s.

Table 5.12 Geometric Bounds of Design Variables

No	Parameter	Lower Bound (mm)	Upper Bound (mm)
1	F_4	80	120
2	F_5	220	280
3	F_6	330	400
4	ST_1	25	50
5	ST_2	25	50
6	SD_1	100	200
7	SD_2	80	200
8	SW_1	150	280
9	SW_2	150	250
10	L_2	70	130
11	L_3	80	120
12	L_4	80	120
13	L_5	150	250

5.1.3.3 Objective Function

The aim in this study is to find the grain geometry that generates maximum average thrust by meeting the design constraints. Therefore, the objective function is defined as given Equation (5.7), which is the average thrust.

$$f(\mathbf{x}) = \frac{\sum_{i=1}^{N_{tb}} F_{des\,i}(\mathbf{x})}{N_{tb}} \tag{5.7}$$

Then, the optimization problem can be stated:

$$\begin{aligned} &\text{maximize} && f(\mathbf{x}) \\ &\text{subject to} && l_i \leq x_i \leq u_i \quad i = 1,\ldots,13 \\ &&& p_c(\mathbf{x}) \leq 65 \ \text{(bar)} \\ &&& m_p(\mathbf{x}) \leq 5100 \ \text{(kg)} \\ &&& 71 \leq t_b(\mathbf{x}) \leq 77 \ \text{(s)} \end{aligned} \tag{5.8}$$

where **l** is a vector containing the lower bounds and **u** is a vector containing the upper bounds on design variables.

5.1.3.4 Optimization Parameters

Optimization parameters are determined as given in Table 5.13. The population number and generation number are taken as 20 and 30 respectively to use the same number of function evaluations given in the reference as 600.

Table 5.13 Optimization Parameters of Test Case-3

N_b	8	***Selection Method***	Roulette-Wheel
N_p	20	***Crossover Method***	Two-Point
N_{gen}	30	***Replacement Method***	Elitism
P_c	0.8		
P_m	0.03		

5.1.3.5 Results

Average and best objective function value (fitness value) of the population at each generation is given in the following figure. Optimization process takes about 7 hours and the average thrust value of the found geometry after 30 generations is 181.2 kN.

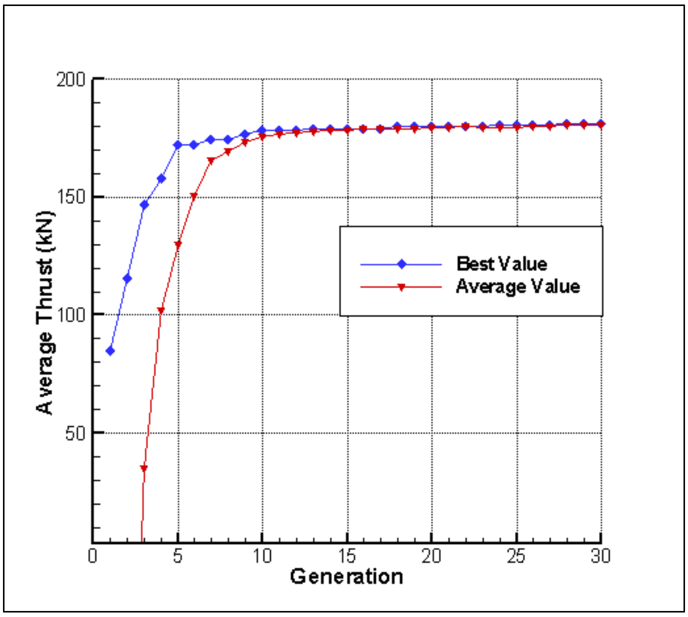


Figure 5.19 Best and Average Values of Objective Function at Each Generation

The solutions found with the developed optimization tool and given in the reference are presented in Table 5.14. The present grain has similar ballistic performance with the reference geometry. The average thrust is increased 4.6 kN and the maximum chamber pressure gets closer to upper bound.

Table 5.14 Results for Radial Slot Grain

Parameter	**Unit**	**Reference Solution**	**Present Solution**
F_4	mm	96.5	108.1
F_5	mm	266.4	267.5
F_6	mm	352.0	331.9
ST_1	mm	28.6	25.4
ST_2	mm	36.6	26.8
SD_1	mm	160.8	100.4
SD_2	mm	122.5	86.1
SW_1	mm	268.5	274.4
SW_2	mm	196	216.3
L_2	mm	83.7	90.0
L_3	mm	98.3	80.0
L_4	mm	96.0	98.5
L_5	mm	188.0	161.4
F_{av}	kN	176.6	181.2
m_p	kg	4937	4969
t_b	s	74.5	71.5
p_{c_max}	bar	61.6	65.0

The cross-sectional view of the present grain is presented in Figure 5.20. As seen from the figure, the inner cylindrical void is almost the same with the reference solution. Mainly, the location of the radial slots is changed by the present study.

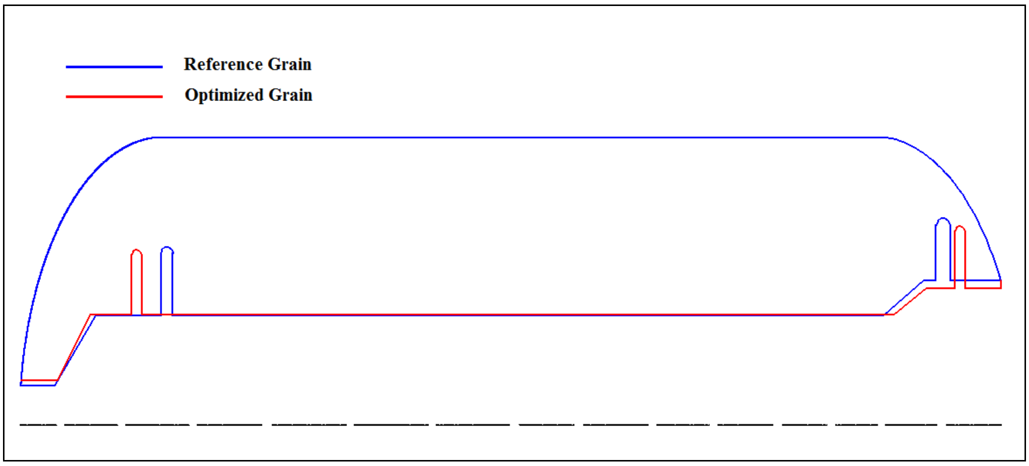


Figure 5.20 Cross-Sectional View of the Present Solution

Thrust-time curves of the present and reference grains are shown in Figure 5.21.

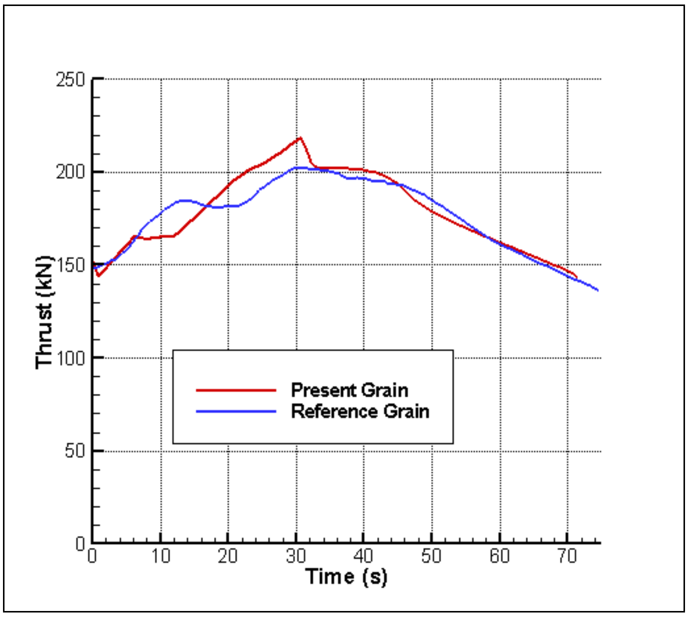


Figure 5.21 Thrust-Time Curve of the Present Solution

This study shows that the developed optimization tool gives similar results with the reference study. To make a certain comparison between two studies is not possible, since the method of burnback analysis used in the studies is different, so each solution has different numerical errors.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

In this thesis study, an optimization tool that can be used for 2-D and 3-D grain design problems is developed. The optimization tool enables to model the grain geometry parametrically and to predict the ballistic performance of the modeled geometry. By using genetic algorithms, the geometry is optimized for a specified ballistic objective and design constraints.

For the geometric modeling and burnback analysis of the propellant grain, a FORTRAN code, named BB3D is developed. Grain geometry can be modeled parametrically by using simple geometries; like cylinder, cone, sphere, ellipsoid, prism and torus, defined in BB3D. The code calculates the burning surface area of the grain for each burnback steps by evaluating the volume change with respect to web thickness. The code is validated with 2-D and 3-D grain samples and the results are compared with analytical, numerical and CAD solutions. Taking the analytical and CAD solutions as a reference, BB3D calculates the change of burning surface area with an error less than 0.2 %. The number of axial and radial stations used for volume calculation specifies the error level. Therefore, the calculation step size is an important parameter for the burnback analysis.

To predict the ballistic performance of the solid rocket motor, the internal ballistics solver developed by Açık [13] is used. The solver calculates the ballistic performance parameters by using 0-D quasi-steady flow equations in the combustion chamber and 1-D isentropic flow equations in the nozzle.

As an optimization algorithm, genetic algorithms which are derivative-free, global search methods are utilized. Different genetic operators (roulette wheel and tournament selection methods, one-point and two-point crossover methods, best alive and elitism replacement methods) are studied in order to find proper optimization parameters giving the best solution for the grain design problem. Additionally, the effect of the crossover and mutation probabilities on the solution is investigated. Although the effect of the optimization parameters on the problem solution can change according to problem type and design space; with the given test case study, it is seen that the selection of genetic operators can alter the solution about 1 % when the proper probabilities are used.

Lastly, the studied test cases show that the developed grain design optimization tool is robust enough for such an engineering application. Grain designer kicks off the preliminary design phase with a good solution satisfying the geometric and ballistic requirements by using the optimization tool.

6.2 FUTURE WORK

For this study;

1. implementation of adaptive step size method for burnback analysis that can decrease the computation time of BB3D code,
2. implementation of 1-D flow solvers handling the performance prediction more detail such as erosive burning effects,
3. improving the performance and effectiveness of genetic algorithms by hybridization or parallelization,
4. implementation of structural design parameters into optimization process,

would be the future areas of interest for further research.

REFERENCES

[1] Sutton, G.P., Biblarz, O., *“Rocket Propulsion Elements”*, 7th ed., John Wiley and Sons, Inc., New York , 2001.

[2] Anderson, M., Burkhalter, J., Jenkins, R., "*Multi-Disciplinary Intelligent Systems Approach to Solid Rocket Motor Design, Part I: Single and Dual Goal Optimization*", AIAA 2001-3600, 2001.

[3] Billheimer, J.S., *"Optimization and Design Simulation in Solid Rocket Design,”* AIAA 68-488, 1968.

[4] Woltosz, W.S., *“The Application of Numerical Optimization Techniques to Solid-Propellant Rocket Motor Design"*, M.S. Thesis, Auburn University, March 1977.

[5] Foster, W.A., Sforzini, R.H., *"Optimization of Solid Rocket Motor Igniter Performance Requirements”*, AIAA 78-1016, 1978.

[6] Sforzini, R.H., *“An Automated Approach to Design of Solid Rockets Utilizing a Special Internal Ballistics Model”*, , AIAA/SAE/ASME 16th Joint Propulsion Conference, AIAA 80-1135,1980.

[7] Sforzini, R.H., *"Design and Performance Analysis of Solid-Propellant Rocket Motors Using a Simplified Computer Program"*,Final Report, NASA Contractor Report CR-129025, Auburn University, Oct. 1972.

[8] Clegern, J.B., *"Solid Rocket Motor Conceptual Design - The Development Of A Design Optimization Expert System With A Hypertext User Interface,"* AIAA 93-2318, 1993.

[9] Hainline, R.C., *“Design Optimization of Solid Rocket Motor Grains for Internal Ballistic Performance”*, M.S. Thesis, Dept. of Mechanical, Materials and Aerospace Engineering, University of Central Florida, 2006.

[10] Nisar, K., Guozhu, L., *“A Hybrid Approach for Design Optimization of Wagon Wheel Grain for SRM”*, AIAA 2008-4893, 2008.

[11] Nisar, K., Guozhu, L., *“A Hybrid Optimization Approach for SRM FINOCYL Grain Design”*, Chinese Journal of Aeronautics, Vol. 21, pp. 481-487, 2008.

[12] Kamran, A., Guozhu, L., *"Design and Optimization of 3D Radial Slot Grain Configuration"*, Chinese Journal of Aeronautics, Vol. 23, pp. 409-414, 2010.

[13] Açık, S., *"Internal Ballistic Design Optimization of a Solid Rocket Motor"*, MS. Thesis, Dept. of Mechanical Engineering, METU, 2010.

[14] Kamran, A., Guozhu, L., *"An Integrated Approach for Optimization of Solid Rocket Motor"*, Aerospace Science and Technology, Vol. 17, pp. 50-64, March 2012.

[15] *"Solid Propellant Grain Design and Internal Ballistics"*, NASA, SP-8076, 1972.

[16] Hartfield, R., Jenkins, R., Burkhalter, J., Foster, W., *"A Review of Analytical Methods for Solid Rocket Motor Grain Analysis",* AIAA 2003-4506, July 2003.

[17] Ricciardi, A., *"Generalized Geometric Analysis of Right Circular Cylindrical Star Perforated and Tapered Grains",* J. Propulsion, Vol. 8, Jan-Feb 1992.

[18] Peterson, E.G., Nielson, G.C., Johnson, W.C., Cook, K.S., Barron, J.G., *"Generalized Coordinate Grain Design and Internal Ballistics Evaluation Program",* AIAA 68-490, CPIA/AIAA 3rd Solid Propulsion Conference, June 1968.

[19] Dunn, S.S., Coats, D.E., *"3-D Grain Design and Ballistic Analysis Using the SPP97 Code",* AIAA 97-3340, 33rd AIAA / ASME / SAE / ASEE Joint Propulsion Conference, 1997.

[20] Toker, K.A., *"Three-Dimensional Retarding Walls and Flow in Their Vicinity"*, Ph.D. Thesis, Dept. of Mechanical Engineering, METU, 2004.

[21] Yıldırım, C., *"Analysis of Grain Burnback and Internal Flow In Solıd Propellant Rocket Motors in 3- Dımensıons"*, Ph.D. Thesis, Dept. of Mechanical Engineering, METU, 2007.

[22] Püskülcü, G., *"Analysis of 3-D Grain Burnback of Solid Propellant Rocket Motors and Verification with Rocket Motor Tests"*, MS. Thesis, Dept. of Mechanical Engineering, METU, 2004.

[23] Willcox, M.A., Brewster, M.Q., Tang, K.C., Stewart, D.S., Kuznetsov, I., *"Solid Rocket Motor Internal Ballistics Simulation Using Three-Dimensional Grain Burnback"*, *",* Journal of Propulsion and Power, Vol. 23, No. 3, May-June 2007.

[24] Dauch, F., Ribereau, D.,*"A Software for SRM Grain Design and Internal Ballistics Evaluation, PIBAL"*, AIAA 2002-4299, July 2002.

[25] Zucrow, M.J., Hoffman, J.D, *"Gas Dynamics, Volume I"*, John Wiley and Sons, Inc., New York, 1976.

[26] Davenas, A., *"Solid Rocket Propulsion Technology"*, Pergamon Press, 1993.

[27] Coats, D.E., Nickerson, G.R., Dang, A.L., Dunn, S.S., Kehtarnavaz, H., *"Solid Performance Program (SPP)"*, AIAA 1987-1701, July 1987.

[28] Luenberger, D.G., Ye, Y., *"Linear and Nonlinear Programming"*, 3rd ed., Springer Science+Business Media, New York, 2008.

[29] Belegundu, A.D., Chandrupatla, T.R*., "Optimization Concepts and Applications in Engineering"*, Prentice-Hall, New Jersey, 1999.

[30] Bazaraa, M.S., Sherali, H.D., Shetty, C.M., *"Nonlinear Programming"*, 3rd ed., John Wiley and Sons, Inc., New Jersey, 2006.

[31] Ryan A. Fellini., *"Derivative-Free and Global Search Optimization Algorithms in an Object-Oriented Design Framework"*, M.S. Thesis, Dept. of Mechanical Engineering and Applied Mechanics, University of Michigan, 1998.

[32] Conn, A.R., Scheinberg, K., Vicente, L.N., *"Introduction to Derivative-Free Optimization"*, Society for Industrial and Applied Mathematics and the Mathematical Programming Society, Philadelphia, 2009.

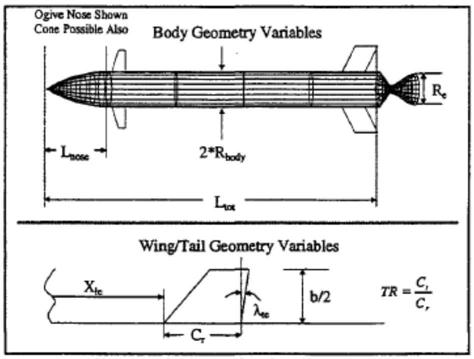
[33] Holland, J.H., "*Adaptation in Natural and Artificial Systems",* University of Michigan Press, 1975.

[34] Hartmann, A.K., Rieger, H., *"Optimization Algorithms in Physics"*, 1st ed., Wiley-VCH Verlag Berlin GmbH, Berlin, 2002.

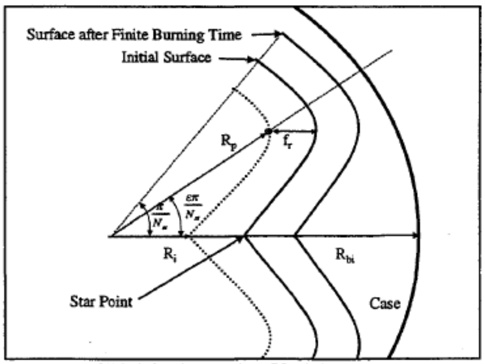
[35] Sastry, K., Goldberg, D., Kendall, G., "*Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques"*, Springer, 2006.

APPENDIX A

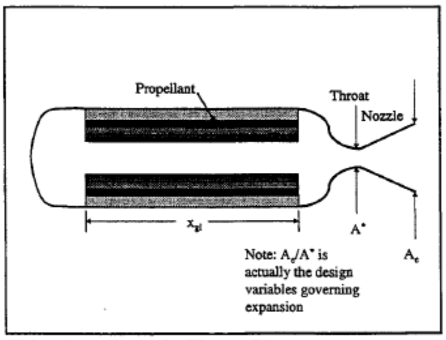
DESIGN VARIABLES OF REFERENCE [2]



External Shape Design Variables



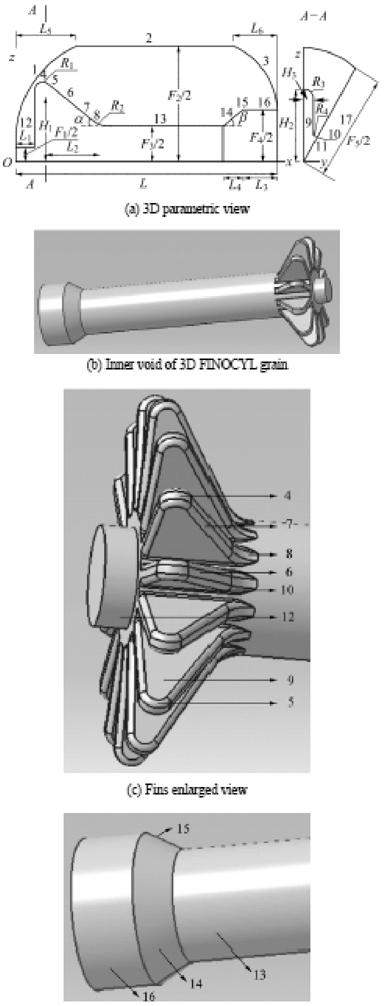
SRM Grain Design Variables


Propellant
Throat
Nozzle
x_{gl}
A^*
A_e
Note: A_e/A^* is actually the design variables governing expansion

SRM Grain Length, Throat Diameter, and Expansion Ratio

APPENDIX B

DESIGN VARIABLES OF REFERENCE [11]



(a) 3D parametric view

(b) Inner void of 3D FINOCYL grain

(c) Fins enlarged view

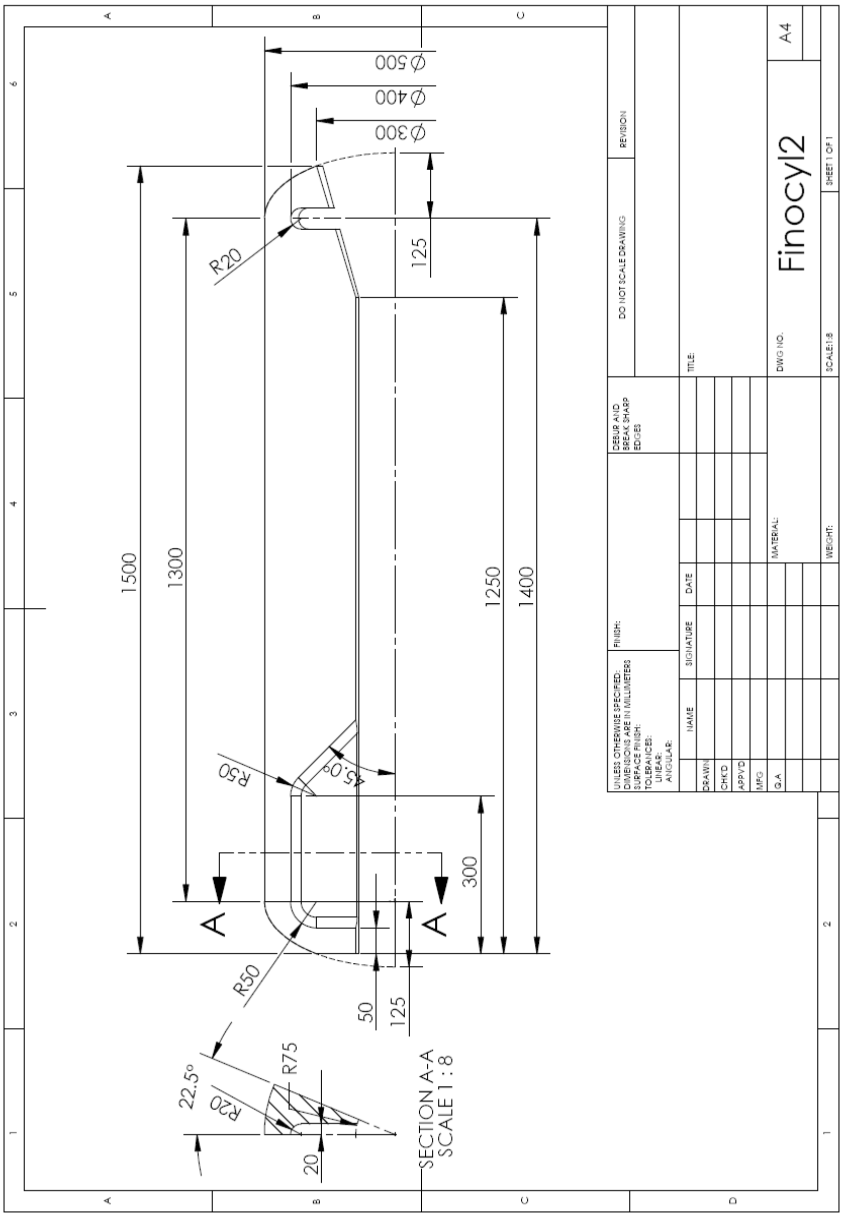
(d) Cone enlarged view

1—Ellipsoid 2—Cylinder x-axis 3—Ellipsoid
4—Cylinder y-axis 5—Ring y-axis 6—Inclined plane
7—Cylinder x-axis 8—Cylinder y-axis 9—Inclined plane
10—Cylinder x-axis 11—Cylinder x-axis
12—Cylinder x-axis 13—Cylinder x-axis 14—Cone
15—Ring x-axis 16—Cylinder x-axis 17—Symmetric plane

Fig.1 3D FINOCYL grain.

APPENDIX C

TECHNICAL DRAWING OF FINOCYL GRAIN



Ø500
Ø400
Ø300
125
R20
1500
1300
1250
1400
R50
45.0°
300
A
A
50
125
R50
22.5°
R75
R20
20
SECTION A-A
SCALE 1 : 8
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBUR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
MATERIAL:
DWG NO.
Finocyl2
A4
WEIGHT:
SCALE:1:8
SHEET 1 OF 1