KINEMATIC SYNTHESIS USING FIREFLY ALGORITHM AND COMPUTER-AIDED ANALYSIS OF CONTINUOUS VARIABLE VALVE LIFT MECHANISM

AISHA ELKAWAFI

SEPTEMBER 2019

KINEMATIC SYNTHESIS USING FIREFLY ALGORITHM AND COMPUTER-AIDED ANALYSIS OF CONTINUOUS VARIABLE VALVE LIFT MECHANISM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES OF
ÇANKAYA UNIVERSITY
BY

AISHA ELKAWAFI

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

SEPTEMBER 2019

Title of the Thesis: **Kinematic Synthesis Using Firefly Algorithm and Computer-Aided Analysis of Continuous Variable Valve Lift Mechanism.**

Submitted by **Aisha ELKAWAFI**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.

Prof. Dr. Can ÇOĞUN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Haşmet TÜRKOĞLU
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Özgün SELVİ
Supervisor

Examination Date: 06.09.2019

Examining Committee Members

Asst. Prof. Dr. Erkin GEZGİN	İzmir Katip Çelebi University	
Asst. Prof. Dr. Hakan TANRIÖVER	Çankaya University	
Asst. Prof. Dr. Özgün SELVİ	Çankaya University	

STATEMENT OF NON-PLAGIARISM PAGE

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 : Aisha ELKAWAFI

Signature :

Date : 06.09.2019

ABSTRACT

KINEMATIC SYNTHESIS USING FIREFLY ALGORITHM AND COMPUTER-AIDED ANALYSIS OF CONTINUOUS VARIABLE VALVE LIFT MECHANISM

ELKAWAFI Aisha

M.Sc., Department of Mechanical Engineering

Supervisor: Asst. Prof. Dr. ÖZGÜN SELVİ

September 2019, 94 pages

In this thesis, a variable valve lift mechanism is proposed and then the kinematic synthesis has been performed using both MATLAB and Autodesk INVENTOR. Moreover, optimization of the lengths of the links has been done using the firefly algorithm in order to reduce the error between the objective valve lift profile, a polynomial cam profile of race engine, and the calculated profile. The optimized mechanism is simulated again with both MATLAB and INVENTOR and the family of valve lifts for various maximum valve opening are compared and show a good consistency. The valve velocity and acceleration are also calculated and compared with the objective functions. Thus, this mechanically based idea of the schematic variable valve lift system can be translated in the future into the design of a physical system with a selected actuation system to provide the best valve lift depending on the engine demand.

Keywords: CVVL system, Variable Valve Lift, Firefly optimization.

ÖZ

KINEMATIC SYNTHESIS USING FIREFLY ALGORITHM AND COMPUTER-AIDED ANALYSIS OF CONTINUOUS VARIABLE VALVE LIFT MECHANISM

ELKAWAFI Aisha

Yüksek Lisans, Makine Mühendisliği Anabilim Dalı

Tez Yöneticisi: Asst. Prof. Dr. ÖZGÜN SELVİ

Eylül 2019, 94 sayfa

Bu tezde bir mekanik değişken supap kaldırma sistemi önerilmiş ve sentezi MATLAB ve Autodesk INVENTOR kullanılarak yapılmıştır. Diğer taraftan, vana profili ve motorun kam profili arsındaki farkı azaltmak amacıyla çubukların uzunluğun optimizasyonu ateşböceği algoritmasını kullanarak gerçekleştirmiştir. Optimize edilmiş mekanizma MATLAB ve INVENTOR kullanılarak simule edilmiştir ve farklı maksimum vana açılması için pozisyon hız ve ivme için hata analizi yapılmıştır ve iyi tutarlılık elde edilmiştir. Böylece bu şematik degişken vana kaldırma sisteminin mekanik temelli fikri gelecekte bir fiziksel sistem tasarımına dönüştürebilir. Bu tasarım seçilmiş bir aktüatör sistem ile en iyi vana kaldırması sağlamak için motorun talebine bağlı gerçekletirilebilir.

Anahtar kelimeler: CVVL sistemi, Değişken Supap Kaldırma sistemi, Firefly optimizasyonu.

ACKNOWLEDGEMENT

I would like to thank my supervisor Dr. Özgün SELVİfor his supervision, patience and support. I am sincerely grateful for his help and guidance through the progress of my thesis.

Special thanks to my mother, father, sisters and brothers for their encouragement. My hearty thanks to my brother Mohammed and my sisters Hanan and Nisreen for their valuable support.

Finally, thanks to Besso and my husband Abdelrahim for all the wonderful things they offer me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In conventional valve train engines, the valve lift depends on the cam profile, which is a function of cam rotation angle. Moreover, the valve motion is fixed with respect to the crank shaft rotation for all the various operating conditions of the engine. The maximum valve lift is determined based on the fluid dynamics considerations. However, in SI engines, there is about 10% of inlet energy loss due to throttling. Throttling will cause the reduction of air to fuel ratio, which will cause the reduction of engine efficiency and; therefore, the increase of gas emission.

Nowadays, all car manufacturers are looking for solutions to reduce fuel consumption, increase engine performance and reduce gas emissions. One of these solutions is the new technology of variable valve lift system (VVL). This system is implemented on the inlet valve train mechanism. It was first used by BMW on 316i model in 2001, and called VALVTRONIC. Therefore, the VVL system today is used widely in petrol engines, but for diesel engines, there are some restrictions on the use of the VVL system, and other systems are developed instead, such as: Variable Geometry Compressor (VGC), and Exhaust Gas Recirculation (EGR), Common Rail, etc.

The VVL system is basically divided into two types:

- Discrete Variable Valve Lift (DVVL) is a two-step valve lift system. One minimal valve lift profile corresponds to minimum engine speed range. Whereas at the maximum engine speed range, the maximal valve lift profile in which the valve is fully opened will be taken place.
- Continuous Variable Valve Lift (CVVL) is a continuous system, where the valve lift varies with the engine speed, and demand. As a result, there will be a family of valve lift profiles.

Hence, studies show the VVL system reduces fuel consumption by 10-15%, reduces emission gases like NOx gases, and eliminates throttle waste in SI engines. Because of all these advantages, the VVL system has become a subject of great interest to many researchers and car manufacturers. Different type of actuation methods has been developed: mechanical, electrical, electro-magnetic, hydraulic and electro-hydraulic actuation.

1.1 Literature Survey

Various studies have been carried out on the VVL system. An innovative solution for throttle-free load control of spark-ignition engines is the Variable Valve Timing system (VVT System) or Variable Valve Actuation System (VVA system), it has been performed by *Mihalcea Stelian in 2010*. In this work, an analytic method for the kinematic analysis of the valve timing mechanism with three elements, the camshaft, the roller rocker finger and an intermediate rocker arm, is presented. This type of

mechanism guarantees a continuous valve lift between two extreme valve lifts, (VVL System) [1].

Xiaozhen Qu, Dojoong Kim in 2012, propose a mechanical CVVA system which uses a four-bar linkage, a rotating first cam, and oscillating second cams. Planar-cam theory is used to analyse the relationship between the cam profiles and valve motion. The kinematic behaviour of the CVVA system was verified with a 3D model built in PRO-ENGINEER software. In order to validate the accuracy of the analysis method, calculated valve lifts are compared with those obtained from CAD software. Results of analysis are used to determine the practicability of the CVVA mechanism for application on automotive engines. A prototype of CVVA system was designed and manufactured. This newly designed CVVA system is effectively operated on a test bench up to 7000 engine rpm, which is the range of the speed of most spark ignition engines. The CVVA system sustains proper valve timing and control over the whole speed range [2].

G. Dritsas, P. Nikolakopoulos and C. Papadopoulos in 2013 developed a design procedure related with the operation of the cam and variable valve lift. The targets of their work are the precise 3D modelling of each component using CATIA, the definition of their mechanical relations that are kinematically and dynamically analyzed and their precise assembly under certain operating conditions. The well-known mechanism from HONDA‟s VTEC was used to perform simulations for displacement, velocity and acceleration of the mechanism. An accurate CAD model is built based on the obtained results and offering the ability to change the depth of valves mechanically, and to get a number of results for important variable valve lift simulations. The novelty of their paper is consisted in showing an integrated simulation methodology in order to precisely

model the follower-cam mechanisms with variable valve lift, and the kinematic and dynamic analysis are performed using the available software [3].

A dynamic analysis as well as an experimental evaluation has been done in a research carried out by *Albatlan and Eid M.* in 2014. It shows an analytical simulation model of the VVL system. A polynomial cam profile is designed to have maximum lift which is higher than the lift of the conventional cam mechanism. A dynamic VVL mechanism model with three degrees of freedom is presented. The N42-BMW cylinder head with Valvetronic VVL of spark-ignition engine was used in the test. The test data focuses on the response of the VVL system and the evaluation of the advanced VVL mechanism. The performance results of a VVL for an overhead camshaft OHC were examined based on theoretical simulation and experimental data to confirm the validity of the VVL system in the test rig [4].

Stelian Mihalcea, et. al. in 2015 showed a new type of continuously variable valve lift (CVVL) system. The mechanism contains a standard roller finger follower, the camshaft, and an intermediate rocker arm. When the adjustment shaft rotates, the intermediate rocker arm is relocated with respect to the cam and the roller of the rocker finger. Hence, the rotation angle of the adjustment shaft is controlled and as a consequence, the valve lift is also controlled. The intermediate rocker arm is equipped with a special developed contact surface, which is one of the novelties of their work. The profile of the contact surface is analytically determined using the theory of the envelope curves. They also established the general conditions to the design of a variable valve timing (VVT) mechanism with continuous (VVL) and mechanical actuation. Numerical simulation shows that the valve lift and timing phases of the mechanism are continuously varied, whereas the valve‟s opening time remains constant. They also

studied the elasto-hydrodynamic lubrication and valve train dynamics. The study is performed by comparing the optimized cam and a standard harmonic which achieves the same maximum displacement of the valve. The optimized cam ensures a better safety in functioning and leading to a greater oil film thickness, and not causing extreme deformations in the engine [5].

Carmelina Abagnale, Mariano Migliaccio and Ottavio Pennacchia in 2015 designed a new mechanical variable valve actuation system, it was developed for high performance motorcycle engines, at University of Napoli Federico II. The study has been conducted implementing a numerical procedure designed to determine cam profile and kinematic and dynamic characteristics of the system, beginning with some input data: rocker arm geometry, relative positions and inertial data of elements, spring stiffness and preloading, camshaft speed, and valve lift profile. The theoretical study performed led to an algorithm used in a program written in Mathcad software. The work has progressed through three main steps leading to three types of variable valve actuation systems, all of which are mechanical systems. This work reports results easy to get with these simple systems that give good perspective of use for two-wheel vehicle engine [6].

Another research work proposes a kinematic analysis of novel CVVL mechanism with three elements for small capacity motorcycle engine (single cylinder) was performed by *Megha M. et al.* in 2016. Here, an inter-mediate rocker arm is equipped with a special contact surface which is based on the theory of envelop curves and has a plane-parallel motion that guarantees a continuous valve lift (VVL System) between two maximum lifts. Numerical solution, Newton Raphson method, carried out in MATLAB, shows that the mechanism continuously varies valve lift while the valves opening time remains the same. The general condition to design a mechanism for continuous VVL has been

calculated. The numerical simulation performed in Ricardo VALDYN-Kinematics, shows that the valve lifts of the mechanism are continuously varied. Additional Kinematic analysis of the same mechanism is calculated, which includes valve lift, velocity, and acceleration. The numerical results, such as velocity, acceleration and valve lift variation at different operating conditions are compared and found to be appropriate under certain operating conditions [7].

The following work was performed at the University of Pitesti by *Adrian C.* in 2017. It presents the analytical kinematic synthesis of continuously variable intake valve lift (VIVL) overhead valve (OHV) mechanism for in-line four-cylinder engine prototype. It is about finding out the required intake valve cam profile beginning with an imposed intake valve lift profile. Then, a computer aided kinematic analysis of the variable intake valve lift mechanism is carried out using CATIA software. The accuracy of the motion which is performed with CAD software is validated by controlling the degree of correlation between the obtained intake valve lift profile and the imposed valve lift used when performing the analytical synthesis [8].

1.2 State of the Art of CVVL System

A number of different principles have been applied to develop continuously variable valve lift systems (CVVL) that range from mechanical systems to hydraulic systems. The objective of reducing fuel consumption led to the introduction of several systems in production [9].

BMW Valvetronic system, with an electrical actuation, was the first continuously variable valve lift mechanism, which entered into production in the BMW 316ti in 2001.

The goal of using Valvetronic was to reduce fuel consumption. Since air flow, and engine output, is controlled by valve lift with the Valvetronic system, the conventional throttle valve is disabled which reduces pumping losses. BMW claims that Valvetronic can provide a 10% reduction in fuel consumption (Autozine 2013). Toyota implemented the Valvematic system on their 1.6L, 1.8L and 2.0L engines and later introduced the system on the 2014 MY 1.8L Corolla in the U.S. market. The Valvematic system has improved fuel economy by 5% and has increased power by 6% in the 2014 MY Corolla (Borge 2013). Fiat Chrysler Multiair system with electro-hydraulic actuation. A wide range of intake valve actuation modes can be obtained by controlling the solenoid valve. Fiat Chrysler claims that Multiair can provide a 10% reduction in fuel consumption with the removal of pumping losses. Moreover, Multiair can provide up to a 10% increase in power when a power-oriented mechanical cam profile is used. Low speed torque can be improved by up to 15% (Murphy 2010). Multiair was first introduced in the 2010 MY 1.4L Chrysler Fiat 500. The next application was in the 2013 MY 2.4L Dodge Dart. later applications have included the 2014 MY Jeep Cherokee, and the 2015 MY Chrysler 200, Jeep Renegade, and Ram ProMaster City [9].

1.3 Objective

Firstly, a structural synthesis based on a physical mechanism idea that will realize a desired motion to find the proposed mechanism is performed. Then, a kinematic analysis on the proposed mechanism to determine the parameters that will realize a given motion by using loop closure equations (LCE) will be carried out. The cam profile will be designed based on the work (The Right Lift) of prof. Gordon Blair on race engines [10], and the 3-4-5 polynomial method is used to construct the displacement, velocity and

accelerations profiles. Hence, the proposed mechanism will be constructed in INVENTOR software and simulations are performed and compared to MATLAB results of loop closure equations. Therefore, the unknown geometrical parameters are optimized by using FIREFLY algorithm with MATLAB and the family of valve lifts (the optimized mechanism) for various positions of the push rod, VVL system, is again found by INVENTOR and MATLAB. Finally, a kinematic analysis to find the velocity and the acceleration of the valve for the various families of valve lifts will be executed in both MATLAB (LCE calculations) and INVENTOR.

CHAPTER 2

KINEMATICS OF VARIABLE VALVE LIFT MECHANISM

2.1 Structural Synthesis

A proposed mechanism has been designed based on the idea of an overhead cam type (OHC), as shown in figure (1).

The mechanism consists of:

- Cam-follower pair, 10 and 11, rotates with a rotation angle(φ).
- Four bar mechanism consists of the links 1, 2 and 3 with lengths a_1, a_2 and a_3, where element 3 has a slot part in which its initial position has a fixed inclination angle (α), and a constant distance from point C to point F that is the sum of r_c and r_f.
- Auxiliary mechanism, responsible on the variation of the valve lift between minimum and maximum, consists of link 4 and 5. The angle (γ) will be held constant for each of the variable valve lifts positions, which means that link 4 will be fixed for each position.
- Push rod, element 6, consists of a roller in the slot of link 3 at F which has a sliding distance S_F. It has a length r_6 and connected at E with link 5 and at G with the rocker arm 7 with revolute joints.

- Rocker arm 7 connected at O with revolute joint and free to slide and rotate at P. It has a length a_7 and it swings and pushing down the valve 9 along tt – direction with a distance $S(\varphi)$, as shown in figure (2).

The Degree of freedom of the mechanism is found by using the general D.O.F equation:

$$F = \lambda(l - j - 1) + \sum_{i=1}^{j} f_i \quad (1)$$

Where,

λ : 3 for planar mechanism.

l : Number of links, $l = 12$.

j: Number of joints, $j = 12R + 1Cs + 1Cp + +1Rj + 1p = 16$.

R: Revolute, Cs: Cylinder in slot , Cp: Cam pair, Rj: Rocker joint and p: prismatic.

Note: the rocker arm joint tip is a two D.O.F joint, as there will be translation in tt – direction and sliding in nn – direction.

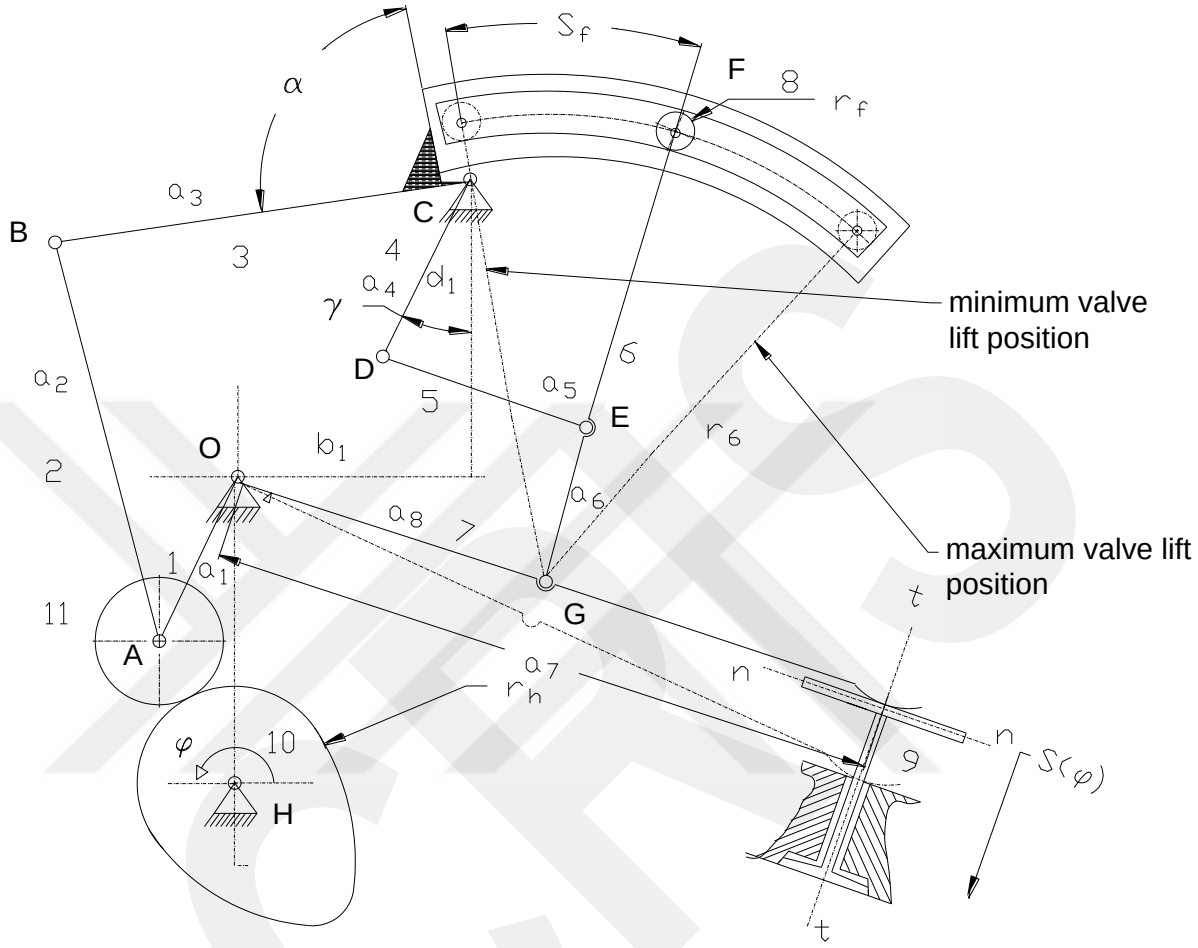
f_i: Degree of freedom of the i^{th} joint, $\sum_{i=1}^{j} f_i = 12 + 1 * 2 + 1 * 2 + 1 * 2 + 1 = 19$.

Then,

$$F = 3(12 - 16 - 1) + 19 = 4$$

Since there are two redundant degrees of freedom; rotation of the follower and the cylinder in slot around its center of rotation, the resulting actual degree of freedom of the mechanism is two. Actually, there will be a translation in the tt-direction, $S(\varphi)$, and a

sliding distance, $S_H(\varphi)$, of the rocker arm on the valve stem, N, as shown in figure(2).



Mechanism elements	
No.	Description
1	crank link
2	link 2
3	link 3
4	link 4
5	link 5
6	push rod
7	rocker arm
8	roller in slot
9	valve
10	Cam
11	Follower

Figure 1: Proposed Mechanism

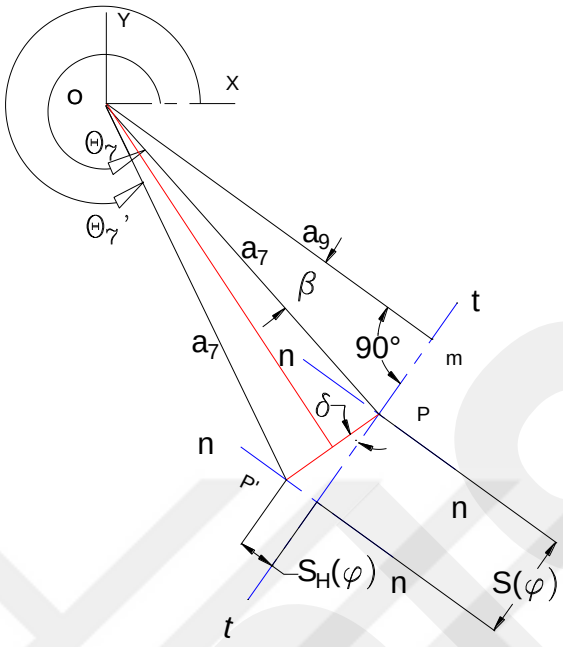


Figure 2: Rocker Arm Movement

2.2 Kinematic Analysis

In order to perform the kinematic synthesis, the number of independent loops should be identified to find the loop closure equations (LCE).

Firstly, the cam pair loop is disconnected. Therefore, to calculate the number of loops the following equation is considered:

$$L = j - l + 1 \tag{2}$$

$l = 9$, since the cam pair and the valve will not be counted.

$j = 12$, since the cam pair joint and the valve prismatic joint are not counted.

So, $L = 12 - 9 + 1 = 4$ loops.

Hence, the mechanism will be disconnected into 4 loops, at points A, F, G and P :

1. The first loop will result in a four bar mechanism by connecting point A, as shown in figure (3).

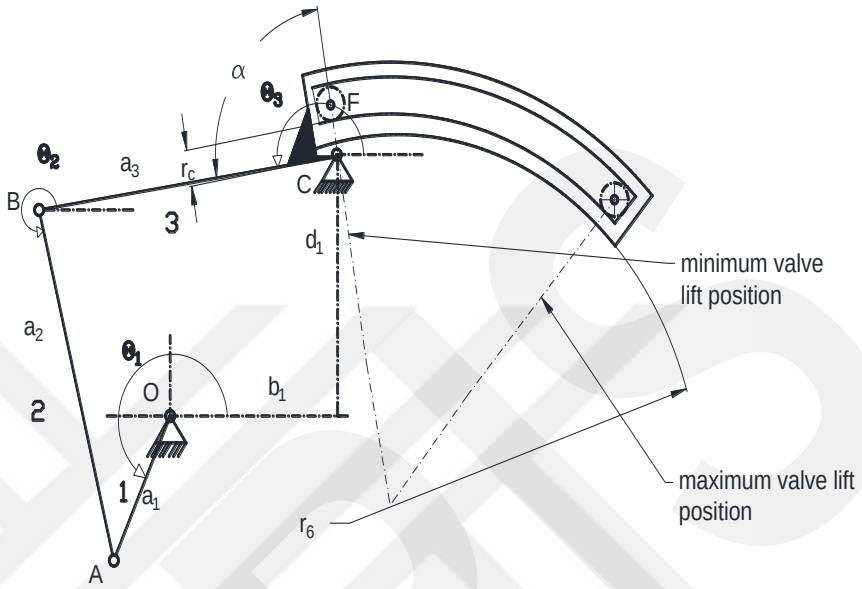


Figure 3: First Loop – Four Bar Mechanism

The loop closure equation is as follows:

$$\overrightarrow{OA} = \overrightarrow{OC} + \overrightarrow{CB} + \overrightarrow{BA} \tag{3}$$

Using complex numbers: $a_1 e^{i\theta_1} = b_1 + id_1 + a_3 e^{i\theta_3} + a_2 e^{i\theta_2}$ (4)

Re: $a_1 cos\theta_1 = b_1 + a_3 cos\theta_3 + a_2 cos\theta_2$ (5)

Im: $a_1 sin\theta_1 = d_1 + a_3 sin\theta_3 + a_2 sin\theta_2$ (6)

2. The second loop will result by connecting point F, as shown in figure (4). In this loop the angle, γ, is kept constant by fixing the link 4.

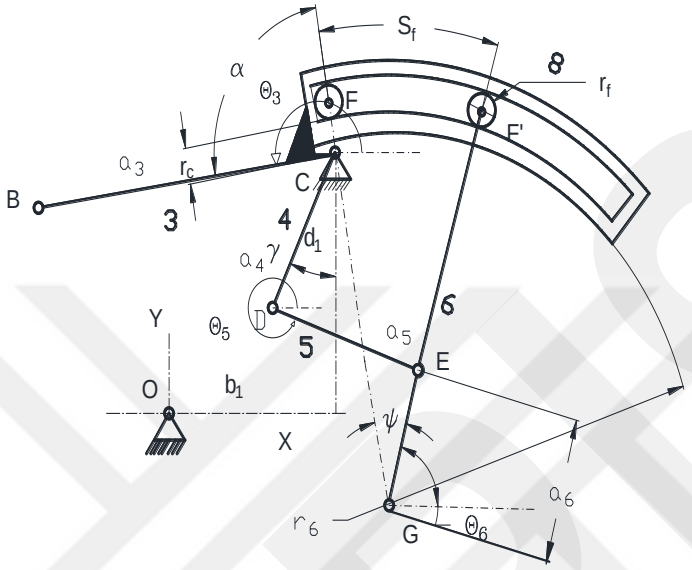


Figure 4: Second Loop – Four Bar Mechanism

The loop closure equation is:

$$\overrightarrow{CF} + \overrightarrow{FF'} = \overrightarrow{CD} + \overrightarrow{DE} + \overrightarrow{EF'} \tag{7}$$

Using complex numbers:

$$(r_c + r_F)e^{i(\theta_3 - \alpha)} + S_F\overrightarrow{S_F} = a_4 e^{-i(\gamma + \frac{\pi}{2})} + a_5 e^{i\theta_5} + (r_6 - a_6)e^{i\theta_6} \tag{8}$$

$$(r_c + r_F)e^{i(\theta_3 - \alpha)} + S_F\overrightarrow{S_F} = a_4 e^{-i\gamma} e^{-i\frac{\pi}{2}} + a_5 e^{i\theta_5} + (r_6 - a_6)e^{i\theta_6} \tag{9}$$

Re: $$(r_c + r_F)\cos(\theta_3 - \alpha) + S_{F_x} = -a_4 \sin\gamma + a_5 \cos\theta_5 + (r_6 - a_6)\cos\theta_6 \tag{10}$$

Im: $$(r_c + r_F)\sin(\theta_3 - \alpha) + S_{F_y} = -a_4 \cos\gamma + a_5 \sin\theta_5 + (r_6 - a_6)\sin\theta_6 \tag{11}$$

Where, S_{F_x} and S_{F_y} can be calculated from equations 14 and 15.

There is also the angle relation:

$$\theta_6 = \theta_3 - \alpha - \psi \tag{12}$$

Where, α is a geometry fixed angle, and ψ is the angle between the minimum valve lift and the consecutives positions.

For the calculation of the sliding distance S_f, an approximation is made by considering the cord length, between F and F', of the sector S_F, as shown in figure (5).

The cord length can be found from the following equation:

$$S_F = 2r_6 sin\frac{\psi}{2} \tag{13}$$

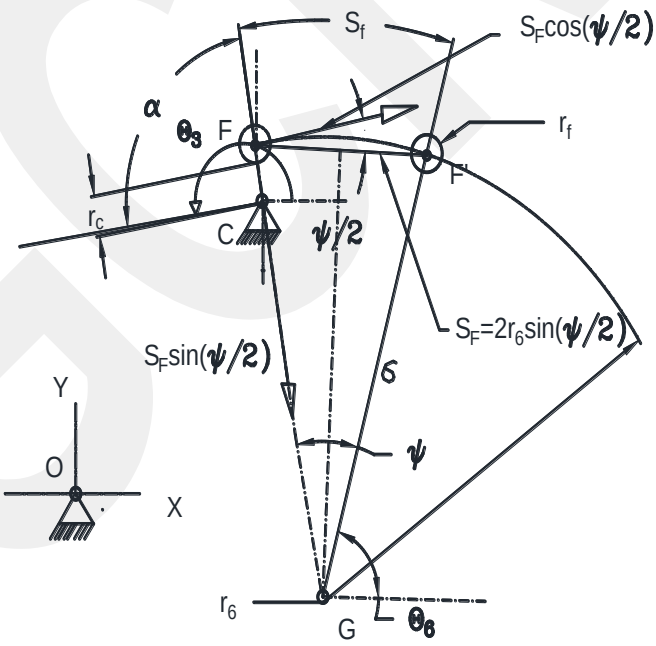


Figure 5: Illustration for the Calculation of S_F Components

Therefore, the x and y-components respectively can be calculated by theses equations, as shown in fig. (5):

$$S_{F_x} = S_F\left[cos(\psi/2)\sin\left(\pi - (\theta_3 - \alpha)\right) + sin(\psi/2)cos\left((\theta_3 - \alpha) + \pi\right)\right] \quad (14)$$

$$S_{F_y} = S_F\left[cos(\psi/2)\cos\left(\pi - (\theta_3 - \alpha)\right) + sin(\psi/2)sin\left((\theta_3 - \alpha) + \pi\right)\right] \quad (15)$$

3. The third loop will result by connecting point G, as shown in figure (6).

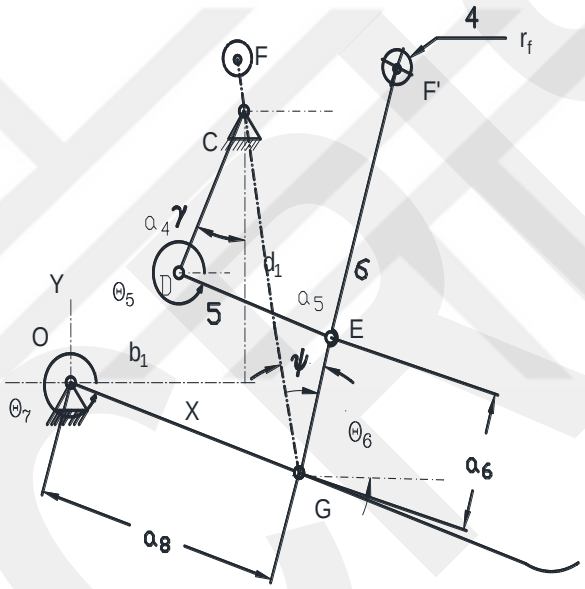


Figure 6: Third Loop – Push Rod and Rocker Arm

The loop closure equation is:

$$\overrightarrow{OG} = \overrightarrow{OC} + \overrightarrow{CD} + \overrightarrow{DE} + \overrightarrow{EG} \quad (16)$$

Using complex numbers:

$$a_8 e^{i\theta_7} = b_1 + id_1 + a_4 e^{-i(\gamma + \frac{\pi}{2})} + a_5 e^{i\theta_5} + a_6 e^{i(\theta_6 + \pi)} \quad (17)$$

$$a_8 e^{i\theta_7} = b_1 + id_1 + a_4 e^{-i\gamma} e^{-i\frac{\pi}{2}} + a_5 e^{i\theta_5} + a_6 e^{i(\theta_6 + \pi)} \quad (18)$$

Re: $$a_8 cos\theta_7 = b_1 - a_4 sin\gamma + a_5 cos\theta_5 - a_6 cos\theta_6 \quad (19)$$

Im: $$a_8 sin\theta_7 = d_1 - a_4 cos\gamma + a_5 sin\theta_5 - a_6 sin\theta_6 \quad (20)$$

4. The fourth loop will result by connecting point P, as shown in figure (7).

The rocker arm 7 is considered at the initial position, θ_7 and at an arbitrary position, θ_7' .

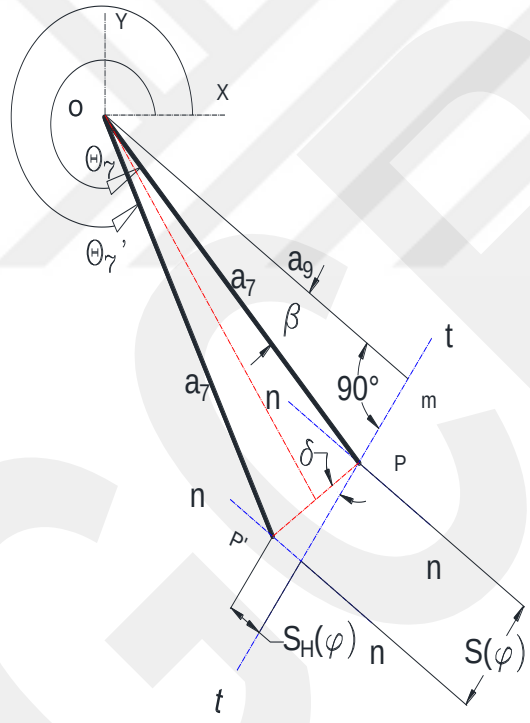


Figure 7: Fourth Loop - Rocker Arm and Valve Stem

The valve lift $S(\varphi)$ in the $t-t$ direction, between the two positions shown of the rocker arm θ_7 and θ_7', and the displacement in the $n-n$ direction, $S_H(\varphi)$, can be found by the following relation:

$$PP' = 2a_7 sin\left(\frac{\theta_7 - \theta_7'}{2}\right) \tag{21}$$

$$\delta = 180 - \left[(90 - \beta) + \left(90 - \left(\frac{\theta_7 - \theta_7'}{2}\right)\right)\right] \tag{22}$$

$$\delta = \beta + \left(\frac{\theta_7 - \theta_7'}{2}\right) \tag{23}$$

$$S(\varphi) = 2a_7 sin\left(\frac{\theta_7 - \theta_7'}{2}\right) cos\left(\beta + \left(\frac{\theta_7 - \theta_7'}{2}\right)\right) \tag{24}$$

$$S_H(\varphi) = 2a_7 sin\left(\frac{\theta_7 - \theta_7'}{2}\right) sin\left(\beta + \left(\frac{\theta_7 - \theta_7'}{2}\right)\right) \tag{25}$$

$$\beta = \cos^{-1}\frac{a_9}{a_7} \tag{26}$$

Finally, the unknown variable parameters are:

$\{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \psi, S, S_H\}$, 10 unknown variables; whereas γ is held constant each push rod position. The number of equations required can be found from the following relation:

$$E = V - F \tag{27}$$

Where,

E: number of equations. V: number of unknown variables. F: Degree of freedom of the mechanism.

$$E = V - F = 10 - 2 = 8$$

Actually, there are eight equations, eqns. 5,6,10,11,19,20,24 and 25, resulted from the four loops.

2.3 The Objective Function

The main point in designing a four-stroke engine is to intake air and exhale exhaust gas and this function is performed by valves whose valve lift and duration at a specific design speed are the real objective of the design of an engine. In other words, the valves and valve lift are firstly designed to satisfy an engine breathing requirement and then secondly the design of the entire cam mechanism to perform that required job successfully [10]. In the work of *Prof. Gordon Blair* on race engines, five different designs of valve lift profile are presented with the same total lift of 10 mm and total durations of 200 degrees of the camshaft rotation. The first and last part of the lift is very shallow. It is known as the „ramp", which is designed to take up the valve clearance (valve lash). This ramp is 25 degrees long and 0.3 mm rise. The valve lash, at hot running engine, will be about 0.2 or 0.25 mm. The valve lift profile that will be used as an objective function in this research will be estimated considering this work of the Right Lift [10].

The maximum valve lift profile, the valve velocity and acceleration are shown in figures 8, 9 and 10, respectively, which is described by using a 3-4-5 polynomial given by the following equations [11]:

$$S(\varphi) = C_0 + C_1\left(\frac{\varphi}{\phi}\right) + C_2\left(\frac{\varphi}{\phi}\right)^2 + C_3\left(\frac{\varphi}{\phi}\right)^3 + C_4\left(\frac{\varphi}{\phi}\right)^4 + C_5\left(\frac{\varphi}{\phi}\right)^5 \quad (28)$$

$$v(\varphi) = \frac{1}{\phi}\left[C_1 + 2C_2\left(\frac{\varphi}{\phi}\right) + 3C_3\left(\frac{\varphi}{\phi}\right)^2 + 4C_4\left(\frac{\varphi}{\phi}\right)^3 + 5C_5\left(\frac{\varphi}{\phi}\right)^4\right] \quad (29)$$

$$a(\varphi) = \frac{1}{\phi^2}\left[2C_2 + 6C_3\left(\frac{\varphi}{\phi}\right) + 12C_4\left(\frac{\varphi}{\phi}\right)^2 + 20C_5\left(\frac{\varphi}{\phi}\right)^3\right] \quad (30)$$

With the following boundary conditions, and $\phi = 100°$:

At $\varphi = 25^\circ$ & 175°, $S = 0.3\,mm,\ v = 0.012\ \frac{mm}{deg}$ and, $a = 0$.

At $\varphi = 50^\circ$, $S = 3.7mm,\ v = 0.2327\frac{mm}{deg}.$ and, $a = 0$.

At $\varphi = 100^\circ$, $S = 10mm,\ v = 0$ and, $a = -0.0045\ \frac{mm}{\text{deg}^2}$.

Solving equations 28, 29 and 30 with using the boundary conditions, we obtain the constants as in table (1).

Table 1: Constants of 3-4-5 Polynomials to Construct the Objective Valve Lift Profile

Constant	Values for first polynomial $25^\circ \le \varphi \le 50^\circ$ & $150^\circ \le \varphi \le 175^\circ$	Values for second polynomial $50^\circ \le \varphi \le 150^\circ$
C_0	-18.90	17.06
C_1	275.30	-157.28
C_2	-1512.50	494.34
C_3	3898.60	-632.44
C_4	-4637.40	375.68
C_5	2096.60	-87.36

The relation is linear for the ramp of 0.3mm with 25° long, for $0^{\circ} \leq \varphi \leq 25^{\circ}$ and $175^{\circ} \leq \varphi \leq 200^{\circ}$ the equations are:

$$S_1(\varphi) = 1.2\left(\frac{\varphi}{\phi}\right) \tag{31}$$

$$v_1(\varphi) = 0.012\ mm/deg$$

$$a_1(\varphi) = 0$$

The valve lift profile equation for $25^{\circ} \leq \varphi \leq 50^{\circ}$ and $150^{\circ} \leq \varphi \leq 175^{\circ}$:

$$S_2(\varphi) = -18.9 + 275.3\left(\frac{\varphi}{\phi}\right) - 1512.5\left(\frac{\varphi}{\phi}\right)^2 + 3898.6\left(\frac{\varphi}{\phi}\right)^3 - 4637.4\left(\frac{\varphi}{\phi}\right)^4 + 2096.6\left(\frac{\varphi}{\phi}\right)^5 \tag{32}$$

$$v_2(\varphi) = \frac{1}{\phi}\left[275.3 - 2 * 1512.5\left(\frac{\varphi}{\phi}\right) + 3 * 3898.6\left(\frac{\varphi}{\phi}\right)^2 - 4 * 4637.4\left(\frac{\varphi}{\phi}\right)^3 + 5 * 2096.6\left(\frac{\varphi}{\phi}\right)^4\right] \tag{33}$$

$$a_2(\varphi) = \frac{1}{\phi^2}\left[-2 * 1512.5 + 6 * 3898.6\left(\frac{\varphi}{\phi}\right) - 12 * 4637.4\left(\frac{\varphi}{\phi}\right)^2 + 20 * 2096.6\left(\frac{\varphi}{\phi}\right)^3\right] \tag{34}$$

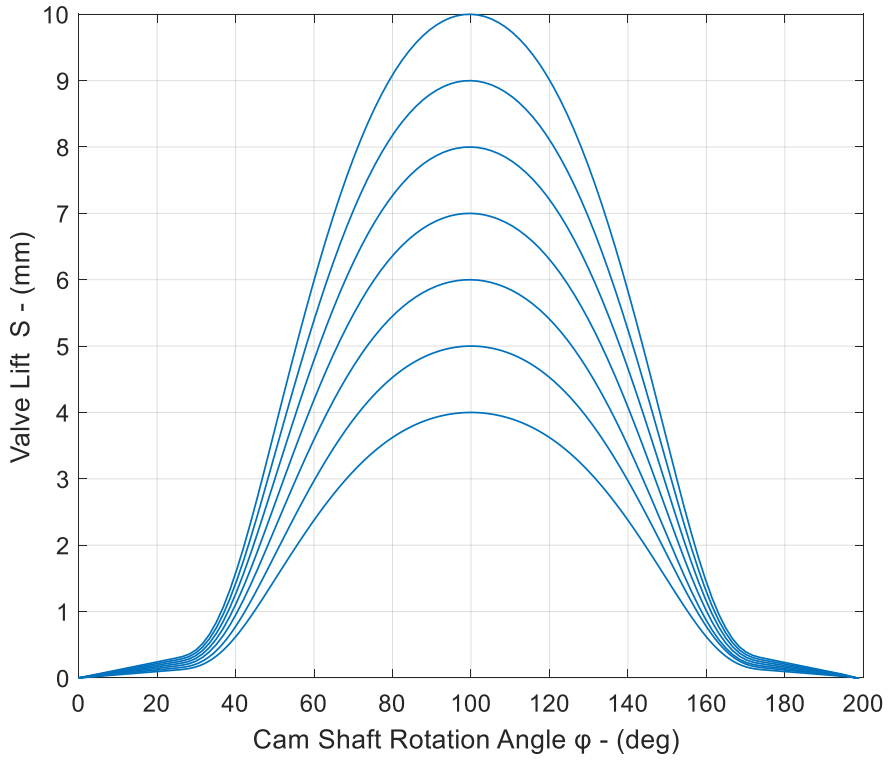


Figure 8: The Objective Family of Valve Lift Profile

The valve lift profile equation for $50^{\circ} \leq \varphi \leq 150^{\circ}$:

$$S_3(\varphi) = 17.06 - 157.28\left(\frac{\varphi}{\phi}\right) + 494.34\left(\frac{\varphi}{\phi}\right)^2 - 632.44\left(\frac{\varphi}{\phi}\right)^3 + 375.68\left(\frac{\varphi}{\phi}\right)^4 - 87.36\left(\frac{\varphi}{\phi}\right)^5 \quad (35)$$

$$v_3(\varphi) = \frac{1}{\phi}\left[-157.28 + 2 * 494.34\left(\frac{\varphi}{\phi}\right) - 3 * 632.44\left(\frac{\varphi}{\phi}\right)^2 - 4 * 375.68\left(\frac{\varphi}{\phi}\right)^3 + 5 * 87.36\left(\frac{\varphi}{\phi}\right)^4\right] \quad (36)$$

$$a_3(\varphi) = \frac{1}{\phi^2}\left[2 * 494.34 - 6 * 632.44\left(\frac{\varphi}{\phi}\right) + 12 * 375.68\left(\frac{\varphi}{\phi}\right)^2 - 20 * 87.36\left(\frac{\varphi}{\phi}\right)^3\right] \quad (37)$$

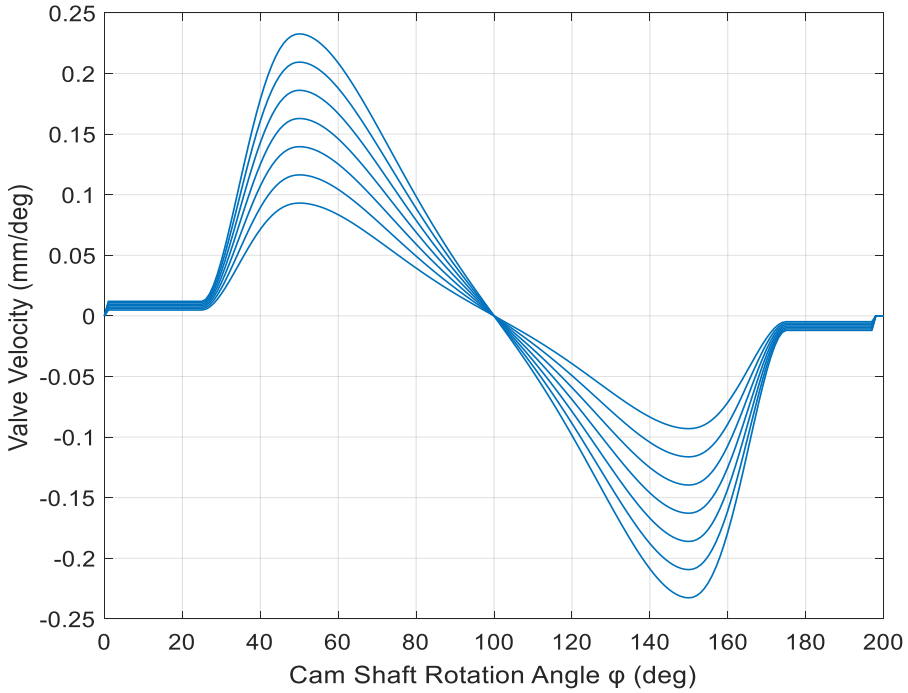


Figure 9: The Objective Family of Valve Velocity Profile

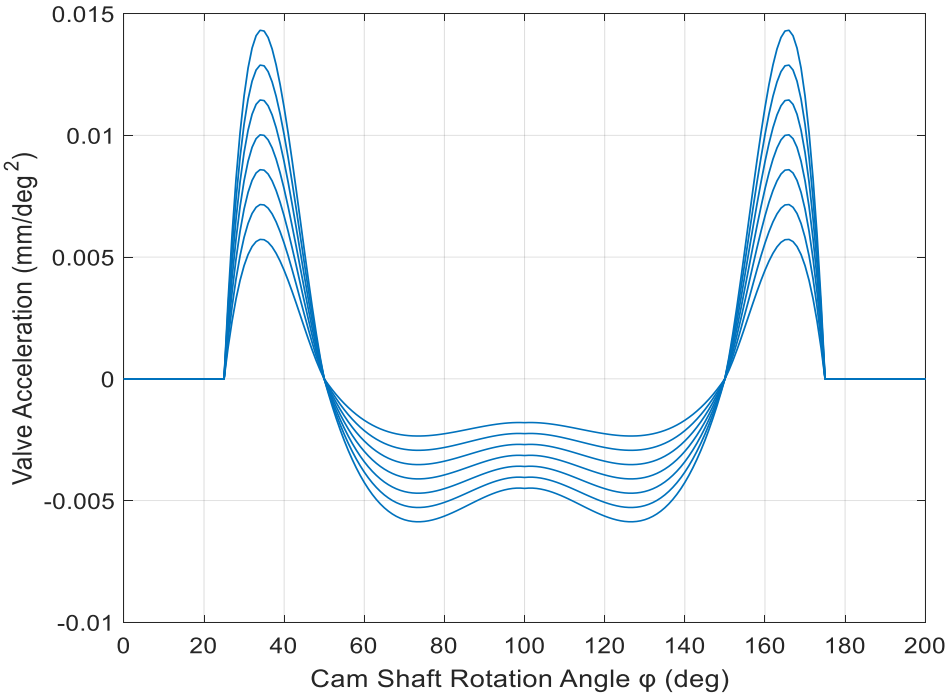


Figure 10: The Objective Family of Valve Acceleration Profile

Lift-duration envelope ratio, Kld, is the area under the valve lift curve divided by the rectangle in which it stays and, whereas it can be deduced at any valve lift, usually it is determined for the valve opening duration at a lift of 0.3 mm or more than 150 degrees of actual valve opening.

The area under the curve:

$$A_1 = \int_{25}^{50} -18.9 + 275.3(\phi/100) - 1512.5(\phi/100)^2 + 3898.6\,(\phi/100)^3 - 4637.2(\phi/100)^4 + 2096.6(\phi/100)^5 d\phi = 34.73 \quad (38)$$

$$A_2 = \int_{25}^{50} 17.06 - 157.28(\phi/100) + 494.34(\phi/100)^2 - 632.44\,(\phi/100)^3 + 375.68(\phi/100)^4 - 87.36(\phi/100)^5 d\phi = 392.17 \quad (39)$$

$$A_c = 2(A_1 + A_2) = 853.8 \quad (40)$$

The area of the rectangle:

$$A_r = 150 * 10 = 1500$$

Then,

$$Kld = \frac{A}{A_r} = \frac{853.8}{1500} = 0.5692 \quad (41)$$

Comparing to "The Right Lift", this ratio is similar to designs A and C in [10].

2.4 Cam Design

The cam can be designed based on the output valve lift profile selected. Some basic assumptions have been considered:

- No eccentricity
- The rule of thumb is to maintain the minimum radius of curvature of the cam pitch curve, ρ_{min}, at least 2 or 3 times as large as the follower radius, R_r.

- Scaling the output valve lift profile to an input profile with lesser maximum profile lift ($8\ mm$), scale of $1: 0.8$

Assuming:

The radius of follower, $R_r = 7.5\ mm$.

The prime circle radius of the cam, $R_0 = 20\ mm$.

The cam profile can be found by using the general expression for radius of curvature of a curve in polar coordinates [12]:

$$\rho_{pitch} = \frac{\left[(R_0+y)^2+(y\prime)^2\right]^{3/2}}{(R_0+y)^2+2(y\prime)^2-(R_0+y)y\prime\prime} \tag{42}$$

The input valve lift profile considered, as shown in figure (11), is given by the following polynomials:

The relation is linear for the ramp of 0.24 mm with 25° long, for $0^\circ \leq \varphi \leq 25^\circ$ and $175^\circ \leq \varphi \leq 200^\circ$ the equations are:

$$S_1(\varphi) = 0.96\left(\frac{\varphi}{\phi}\right) \tag{43}$$

$$v_1(\varphi) = 0.0096\ mm/deg$$

$$a_1(\varphi) = 0$$

The valve lift profile equation for $25^\circ \leq \varphi \leq 50^\circ$ and $150^\circ \leq \varphi \leq 175^\circ$:

$$S_2(\varphi) = C_{01} + C_{11}\left(\frac{\varphi}{\phi}\right) + C_{21}\left(\frac{\varphi}{\phi}\right)^2 + C_{31}\left(\frac{\varphi}{\phi}\right)^3 + C_{41}\left(\frac{\varphi}{\phi}\right)^4 + C_{51}\left(\frac{\varphi}{\phi}\right)^5 \tag{44}$$

$$v_2(\varphi) = \frac{1}{\phi}\left[C_{11} + 2C_{21}\left(\frac{\varphi}{\phi}\right) + 3C_{31}\left(\frac{\varphi}{\phi}\right)^2 + 4C_{41}\left(\frac{\varphi}{\phi}\right)^3 + 5C_{51}\left(\frac{\varphi}{\phi}\right)^4\right] \tag{45}$$

$$a_2(\varphi) = \frac{1}{\phi^2}\left[2C_{21} + 6C_{31}\left(\frac{\varphi}{\phi}\right) + 12C_{41}\left(\frac{\varphi}{\phi}\right)^2 + 20C_{51}\left(\frac{\varphi}{\phi}\right)^3\right] \tag{46}$$

The valve lift profile equation for $50^{\circ} \leq \varphi \leq 150^{\circ}$:

$$S_3(\varphi) = C_{02} + C_{12}\left(\frac{\varphi}{\phi}\right) + C_{22}\left(\frac{\varphi}{\phi}\right)^2 + C_{32}\left(\frac{\varphi}{\phi}\right)^3 + C_{42}\left(\frac{\varphi}{\phi}\right)^4 + C_{52}\left(\frac{\varphi}{\phi}\right)^5 \quad (47)$$

$$v_3(\varphi) = \frac{1}{\phi}\left[C_{12} + 2C_{22}\left(\frac{\varphi}{\phi}\right) + 3C_{32}\left(\frac{\varphi}{\phi}\right)^2 + 4C_{42}\left(\frac{\varphi}{\phi}\right)^3 + 5C_{52}\left(\frac{\varphi}{\phi}\right)^4\right] \quad (48)$$

$$a_3(\varphi) = \frac{1}{\phi^2}\left[2C_{22} + 6C_{32}\left(\frac{\varphi}{\phi}\right) + 12C_{42}\left(\frac{\varphi}{\phi}\right)^2 + 20C_{52}\left(\frac{\varphi}{\phi}\right)^3\right] \quad (49)$$

Solving these equations by Matlab, as shown in Appendix A, and using the scaled boundary conditions, the following constants are obtained as shown in table (2).

Table 2: Constants of Polynomials to Construct the Valve Lift Profile to Design the Cam

Constant	Value	Constant	Value
C_{01}	-15.10	C_{02}	13.648
C_{11}	220.30	C_{12}	-125.824
C_{21}	-1210.00	C_{22}	395.472
C_{31}	3118.80	C_{32}	-505.952
C_{41}	-3710.00	C_{42}	300.544
C_{51}	1677.30	C_{52}	-69.888

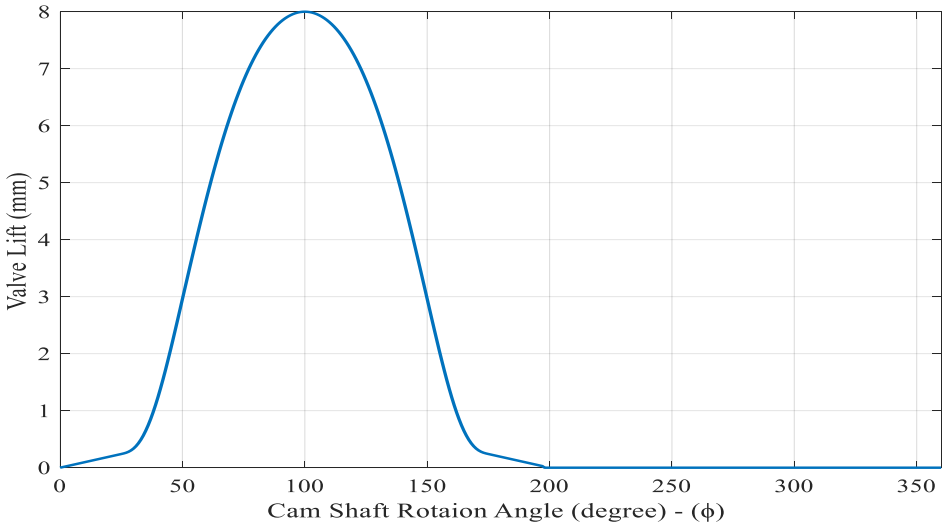


Figure 11: Valve Lift Profile with 8 mm Maximum Lift

By using Matlab, the profile equations (43 - 49) and the curvature equation (42) are solved and the cam profile is found, as shown in figure (12).

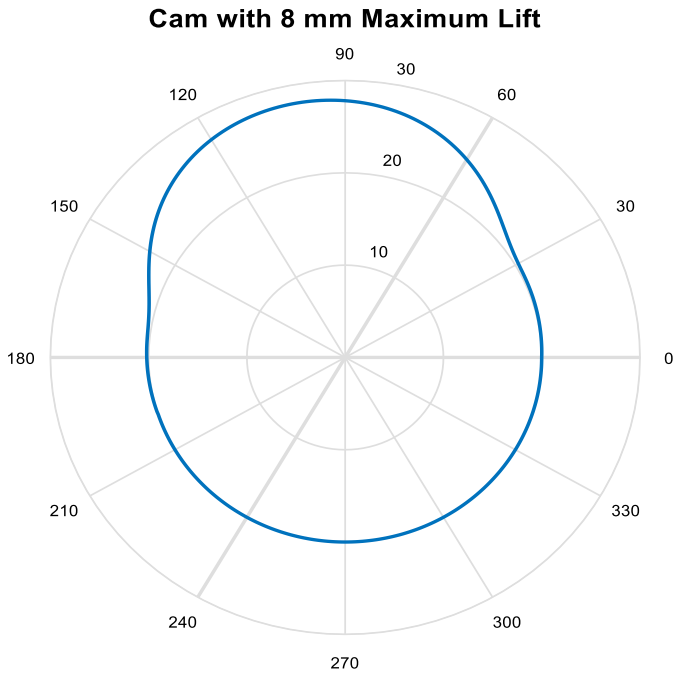


Figure 12: The Cam profile for a Maximum Lift of 8 mm

2.5 Method of Solution Using MATLAB

The loop closure equations are solved by using Matlab code, firstly; the equations are solved for the initial position of the mechanism at the dwell position of the cam. The initial angle for the rocker arm, θ_7, is calculated and used as reference to calculate the angles and the valve displacement with the cam rotation.

From LCE 1, since θ_1 is known at the initial position of cam dwell, $\theta_2 \; and \; \theta_3$ are calculated by solving equations 5 and 6.

Consequently, by using LCE 2, considering that θ_6 is known at the initial position from geometry, then θ_5 & ψ at the initial position can be determined by fixing the angle of the variable valve mechanism, γ, and solving equations 3 and 4 with the substitution of $S_{F_x} \; and \; S_{F_y}$ from equations 12 and 14. After that, θ_6 , can be determined from LCE 2, equations 10 and 11, by back substitution. Hence, θ_7 is calculated from LCE 3 by finding the arctangent from equations 19 and 20.

Finally, the displacement of the valve in the t-t direction $S(\varphi)$, and the sliding of the rocker arm in the n-n direction $S_H(\varphi)$ are calculated from equations 24 and 25.

Moreover, the same above calculations are repeated for various input $\theta_1$angles in Matlab and at every selected push rod position controlled by the angle γ. Therefore, the valve lifts, $S(\varphi)$, are calculated in families and the objective function of the VVL system is reached.

CHAPTER 3

CALCULATION AND SIMULATION

3.1 Mechanism Geometry

In order to check the validity of the kinematic analysis of the mechanism, all lengths of links and geometrical angles, as shown in figure 13, have been assumed as follows:

$a_1 = 25\ mm,\ a_2 = 50\ mm,\ a_3 = 46\ mm,\ a_4 = 40\ mm,\ a_5 = 40\ mm,\ a_6 = 35\ mm,$ $a_7 = 60\ mm\ (effective\ is\ 57.50\ mm),\ a_8 = 35\ mm,\ a_9 = 51.40\ mm,\ b_1 = 17\ mm,$ $d_1 = 48\ mm,\ r_6 = 85\ mm\ \ r_c = 5\ mm,\ r_F = 5\ mm,\ \alpha = 90.5°,$ as shown in figure (13).

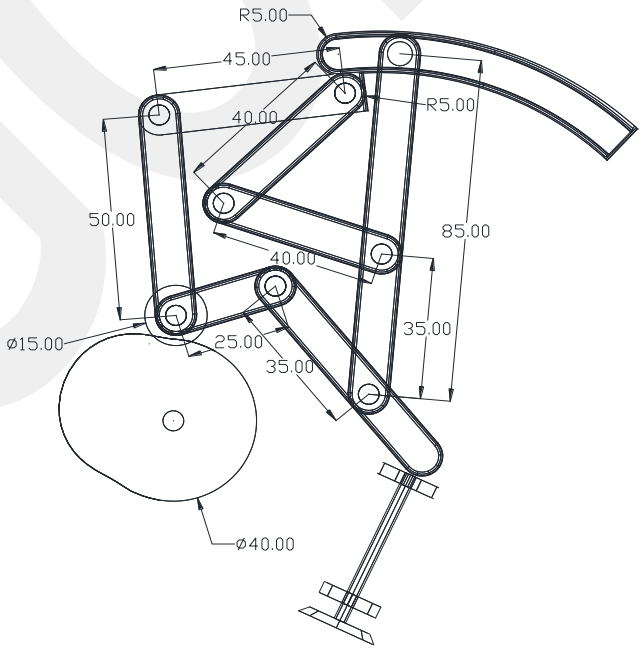


Figure 13: The Proposed Dimensional Mechanism

Note: a_9, has been found after drawing the mechanism as it is the perpendicular distance from the point O to the line of action of the valve, $t - t$.

3.2 Matlab and Simulation Plan:

A plan has been scheduled to compare the results obtained by the kinematic analysis equations, performed by MATLAB, and the simulation carried out by INVETOR at the dwell, $\theta_1 = 193.8°$, and at the maximum valve lift position, $\theta_1 = 175.37°$, these values have been obtained after considering the dimensions of the mechanism and the cam-follower, as follows:

Table 3: Simulation Plan Cases

Case	$\gamma\ (deg.)$
1	45°
2	40°
3	35°
4	30°

According to this plan, the calculation of the following parameters θ_2, θ_3, θ_5, θ_6, $\theta_7, \psi\ and\ S$ is found by both MATLAB and INVENTOR to check the validity of the kinematic synthesis.

3.3 INVENTOR Simulations

The mechanism has been built in the Autodesk INVENTOR software in order to perform the simulations, as shown in figure (14). The simulations are carried out as the plan scheduled in order to compare the results with those of Matlab. The cam aligned with the follower with the dimensions and the profile designed before is used. Also, all the links are built with the predefined lengths and proper dimensions. Different type of joints and constraints were used in order to build the whole mechanism. The cam is assumed to rotate with a constant velocity of $2\,rps$, $720\,deg/s$, and the time is considered one second.

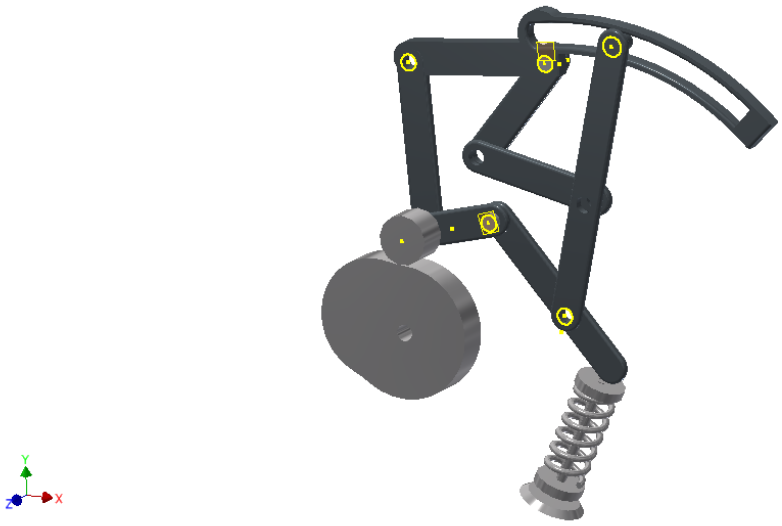


Figure 14: The Proposed VVL Mechanism Built in INVETOR

When performing the simulations as the scheduled plan, the link 4 of the auxiliary mechanism at each simulation is rotated to obtain the required value of the angle, γ, then

grounded. For the different four cases a simulation videos are recorded and, the parameters results are described in the following tables:

CASE 1:

Table 4: Maximum and Minimum Values for Case 1 at γ=45°

θ_1 (deg°)	θ_2 (deg°)	θ_3 (deg°)	ψ (deg°)	θ_5(deg°)	θ_6 (deg°)	θ_7 (deg°)	S (mm)
193.83 Min	275.59	185.23	11.31	341.74	82.20	308.21	0.00
175.37 max	274.88	175.20	13.23	337.49	79.34	301.41	5.65

Case 2:

Table 5: Maximum and Minimum Values for Case 2 at γ=40°

θ_1 (deg°)	θ_2 (deg°)	θ_3 (deg°)	ψ (deg°)	θ_5(deg°)	θ_6 (deg°)	θ_7 (deg°)	S (mm)
193.83 Min	275.59	185.23	16.93	344.33	77.08	308.15	0.00
175.37 Max	274.88	175.20	18.95	338.22	70.97	298.95	7.54

CASE 3:

Table 6: Maximum and Minimum Values for Case 3 at γ=35°

θ_1 (deg°)	θ_2 (deg°)	θ_3 (deg°)	ψ (deg°)	θ_5(deg°)	θ_6 (deg°)	θ_7 (deg°)	S (mm)
193.83 min	275.59	185.23	22.20	346.19	71.71	308.09	0.00
175.37 max	274.88	175.20	25.02	338.12	67.63	296.76	9.17

CASE 4:

Table 7: Maximum and Minimum Values for Case 4 at γ=30°

θ_1 (deg°)	θ_2 (deg°)	θ_3 (deg°)	ψ (deg°)	θ_5(deg°)	θ_6 (deg°)	θ_7 (deg°)	S (mm)
193.83 min	275.59	185.23	27.92	347.15	66.30	308.04	0.00
175.37 max	274.88	175.20	31.10	337.07	61.84	294.87	10.54

Moreover, the family of valve lifts, velocities and accelerations for all the cases considered in the simulation, by Inventor, are plotted against the time, which was considered as half a second for one revolution of the cam, in one graph for comparison, respectively, as shown in figures (15, 16 and 17).

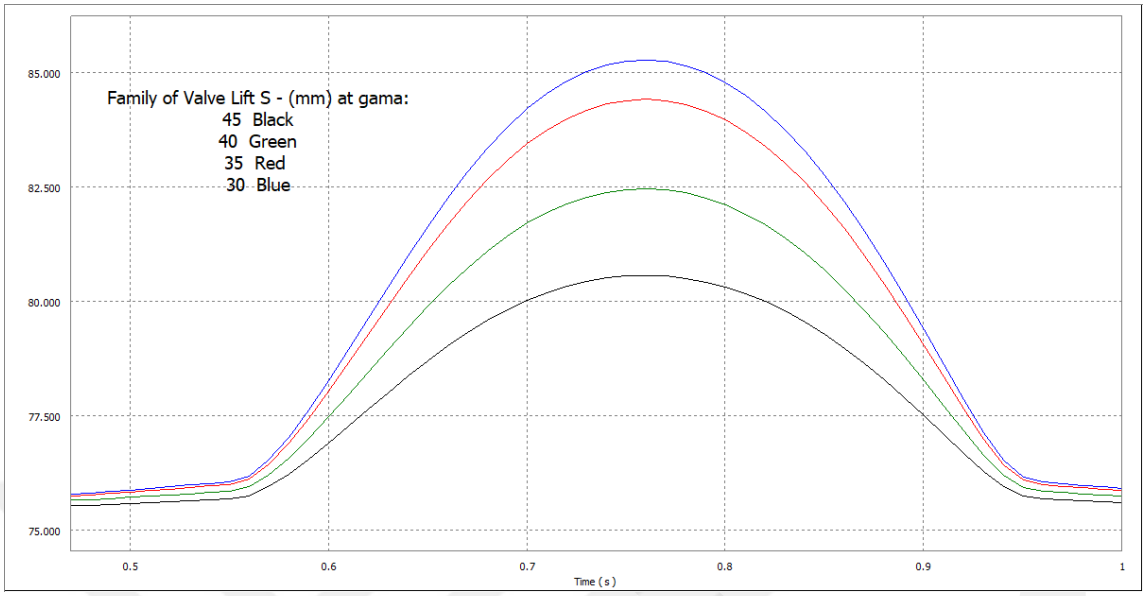


Figure 15: The Family of Valve Lifts at Various Push Rod Positions, γ, by Inventor

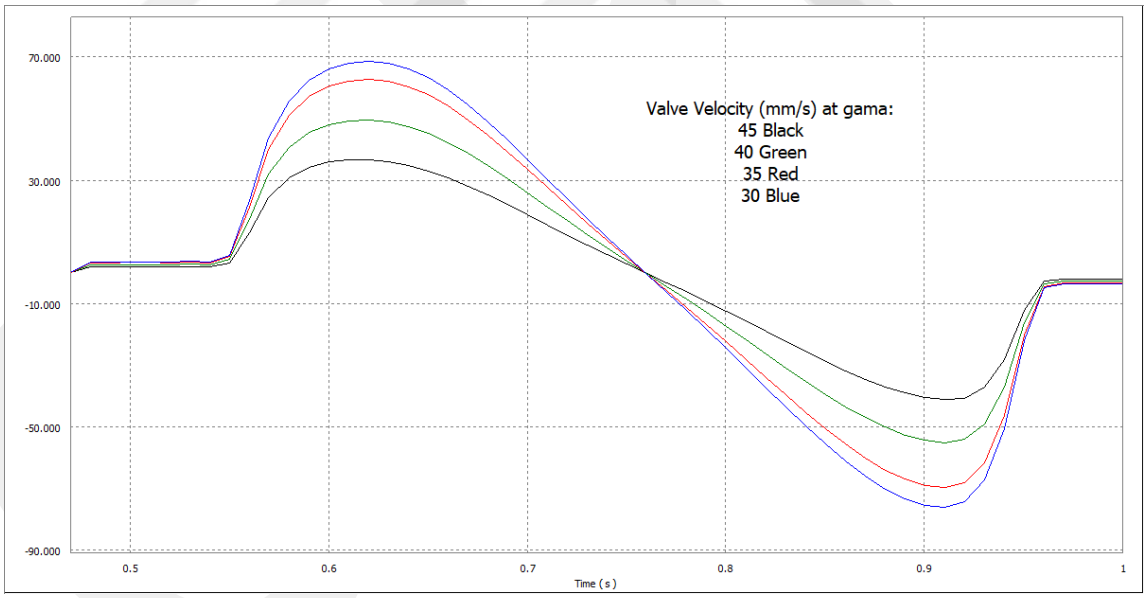


Figure 16: The Family of The Velocity of Valve at Various positions of γ by Inventor

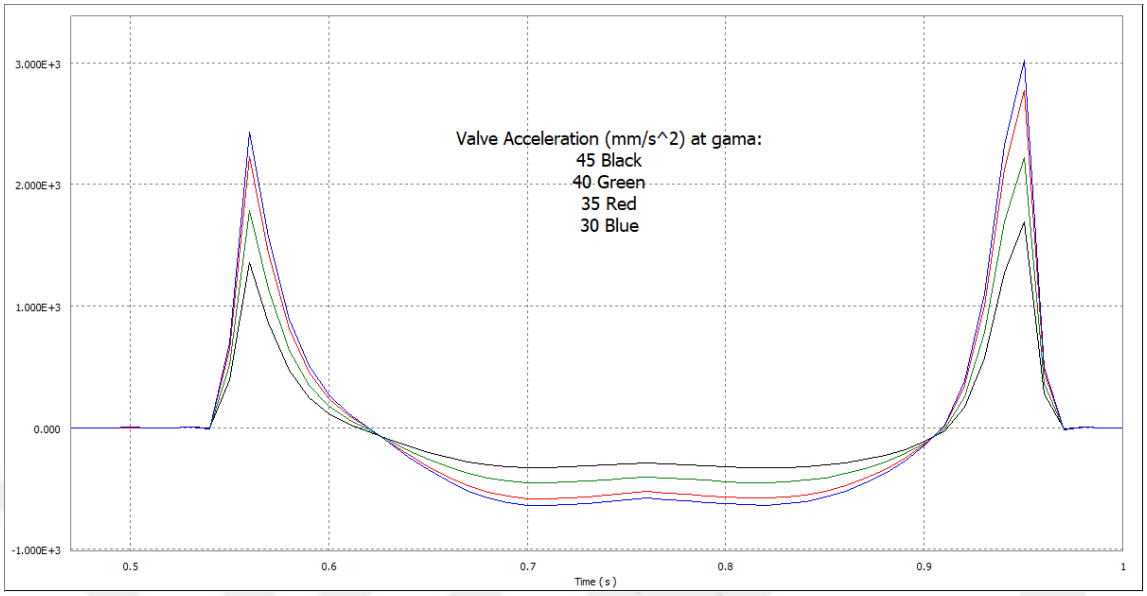


Figure 17: The Family of Acceleration of Valve at Various positions of γ by Inventor

3.4 MATLAB Calculations

The kinematic equations have been solved in Matlab to find the above mentioned parameters at different positions of the push rod in order to achieve the variable valve system function (VVL).

The auxiliary link 4 is the responsible for varying the position of the push rod, which depends on the demand of the engine. The activation of this link can be mechanical, electrical, hydraulic, etc. but it is not going to be specified in this work. After the push rod position is selected, link 4 is held fixed during the rotation of the cam. Therefore, a series of selected orientations of link 4, γ, will be considered, as in the plan, and the results of the valve lift, S, and the parameter angles will be found.

Firstly, a check is performed so that the variation of the valve lift at the maximum lift should not be more than 10 mm. Hence, as the angle, γ, increases the push rod goes to the left of the arc direction, which means reducing the valve lift and vice versa. Thereafter, equations of loop 1 are solved to find the angles θ_2 and θ_3, whereas; the angles θ_5 and ψ are calculated from loop 2 equations and, after that finding the angle θ_6 by the back substitution with the updated value of ψ. Subsequently, the angle of rotation of the rocker arm, θ_7, is calculated by using the equations of loop3. Finally, the valve lift, S, corresponds to each push rod position can be found by using the equation of loop 4.

Matlab results are shown in the following table:

CASE 1:

Table 8: Maximum and Minimum Values for Case 1 at γ=45°

θ_1 (deg)	θ_2 (deg)	θ_3 (deg)	ψ (deg)	θ_5(deg)	θ_6 (deg)	θ_7 (deg)	S (mm)
193.83 min	275.20	185.22	10.96	343.04	83.76	310.90	0.00
175.37 max	274.49	175.18	11.30	337.80	80.26	303.56	6.35

Case 2:

Table 9; Maximum and Minimum Values for Case 2 at γ=40°

θ_1 (deg)	θ_2 (deg)	θ_3 (deg)	ψ (deg)	θ_5(deg)	θ_6 (deg)	θ_7 (deg)	$S\ (mm)$
193.83 min	275.20	185.22	16.09	345.83	78.63	310.90	0.00
175.37 max	274.49	175.18	16.69	338.95	74.87	302.32	7.38

CASE 3:

Table 10: Maximum and Minimum Values for Case 3 at γ=35°

θ_1 (deg)	θ_2 (deg)	θ_3 (deg)	ψ (deg)	θ_5(deg)	θ_6 (deg)	θ_7 (deg)	$S\ (mm)$
193.83 Min	275.20	185.22	21.29	347.82	73.42	310.90	0.00
175.37 Max	274.49	175.18	22.33	339.18	69.23	300.98	8.47

CASE 4:

Table 11: Maximum and Minimum Values for Case 4 at γ=30°

θ_1 (deg)	θ_2 (deg)	θ_3 (deg)	ψ (deg)	θ_5(deg)	θ_6 (deg)	θ_7 (deg)	$S\ (mm)$
193.83 min	275.20	185.22	26.68	348.98	68.03	310.90	0.00
175.37 max	274.49	175.18	28.27	338.43	63.29	299.52	9.63

Moreover, the values of the rotating angle of the follower link, θ_1, have been taken from the simulation carried out by inventor. This resulting vector of link 1 rotation angle, θ_1, as shown in table (20) in Appendix,which consists of 114 points corresponds to the valve opening and closing cycle, has been entered in the Matlab code and the valve lifts for the various planned γ cases have been calculated, the resulting family of valve lifts have been plotted in figure (18).

In additinon, the valve velocity and accelerations have been calculated in Matlab and plotted against the cam rotation angle, φ, in mm/deg and mm/deg^2, as shown in figures (19 &20), respectively.

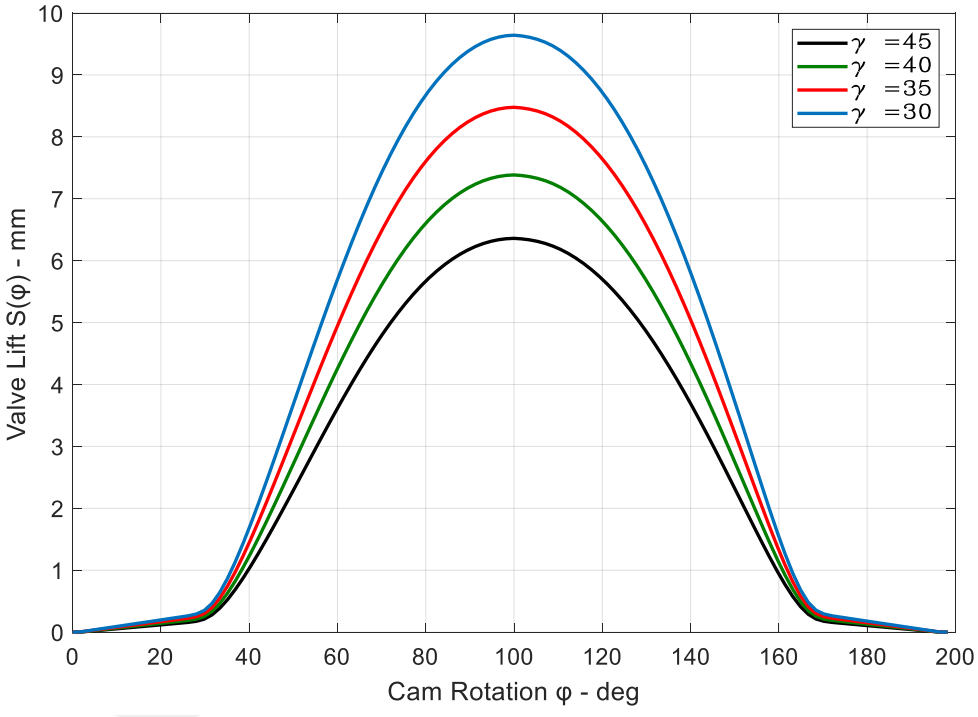


Figure 18: The Family of Valve Lifts at Various Push Rod Positions, γ, by Matlab

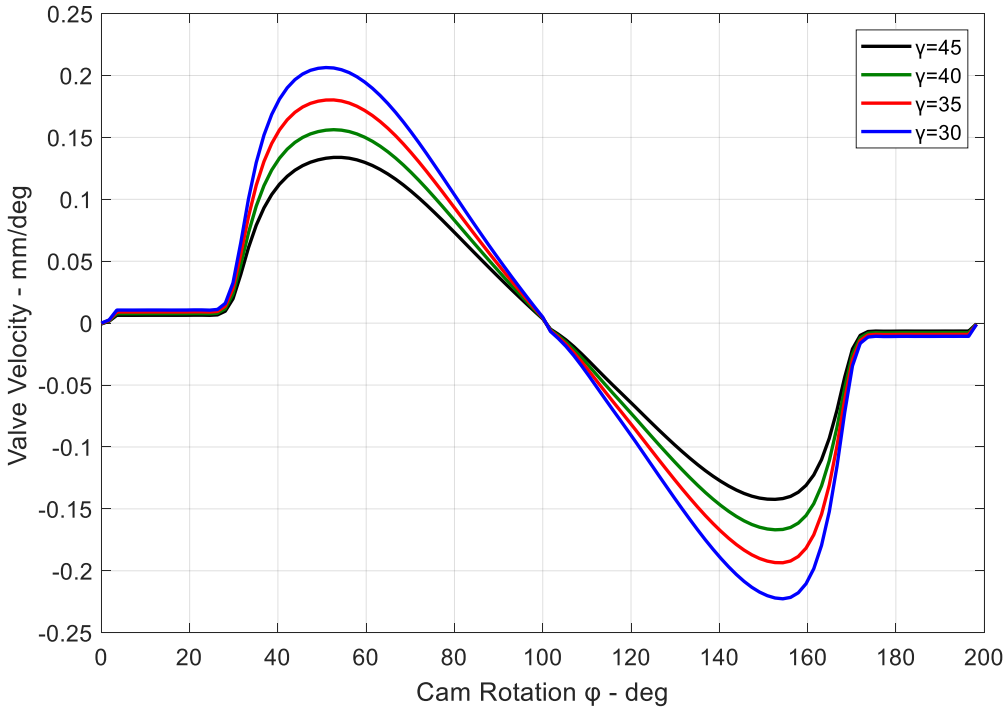


Figure 19: The Family of The Velocity of Valve at Various positions of γ by Matlab

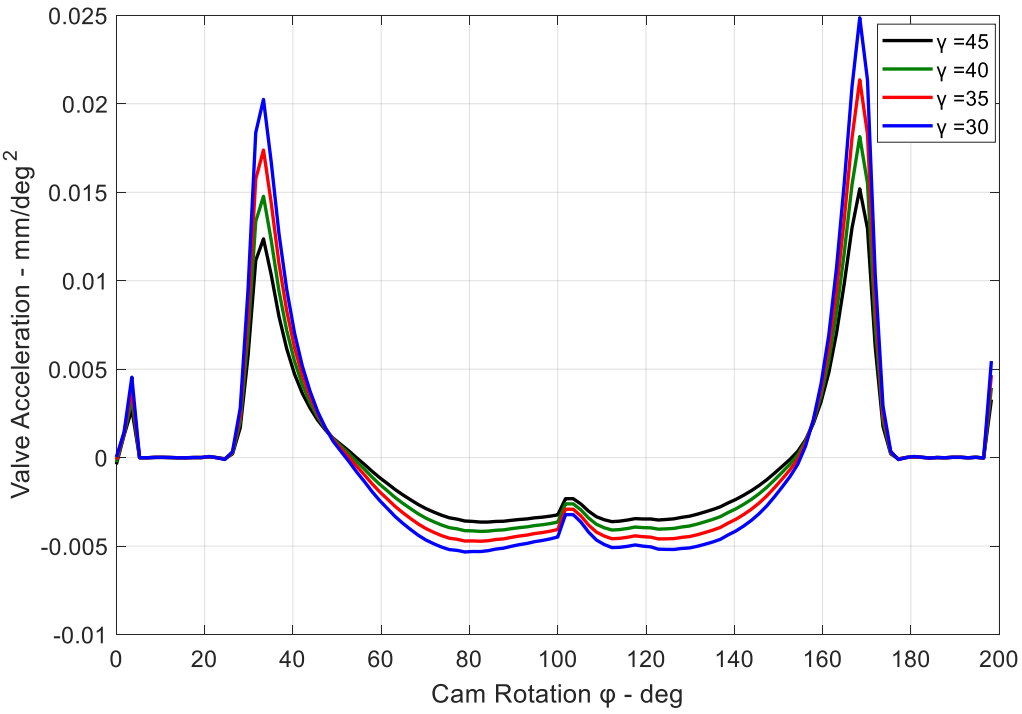


Figure 20: The Family of The Valve Acceleration at Various positions of γ by Matlab

Comparison of the results of the rocker arm angle, θ_7, and the valve lift, S, obtained by Matlab and Inventor can be shown in the following table (12) with the percentage error.

Table 12: Percent Error between MATLAB Calculations and INVENTOR

$\gamma\ (deg)$	Parameters	Matlab Results	Inventor Results	Error (%)
45	θ_7 (deg)	303.56	301.41	0.71
	$S\ (mm)$	6.35	5.65	12.38
40	θ_7 (deg)	302.32	298.95	1.12
	$S\ (mm)$	7.38	7.54	-2.12
35	θ_7 (deg)	300.98	296.76	1.42
	$S\ (mm)$	8.47	9.17	-7.63
30	θ_7 (deg)	299.52	294.87	1.57
	$S\ (mm)$	9.63	10.54	-8.63

From the comparison of errors, it is obvious that the error in calculating the rocker arm angle is not so big and does not exceed 2%, but the error in calculating the valve lift is quite large (12.38%) when the pushing rod angle, γ, is around 45 degrees. This could be an error in INVENTOR due to non-precision in constructing the mechanism as it is very sensitive to the small changes in dimensions.

CHAPTER 4

MECHANISM OPTIMIZATION

4.1 Introduction to Optimization Algorithms

Optimization algorithm can be classified in many different ways. The simplest way is to consider the nature of the algorithm, and this in turn divides the algorithms into two kinds: deterministic and stochastic algorithms. Deterministic algorithms are based on a hard procedure, and its values and paths of both functions and variables are repeatable. Whereas, stochastic algorithms are based on randomness, and Firefly algorithm is a good example; the population solutions will be different each time the program is executed since they use random numbers, though the final results may not vary a lot, but the paths of each individual firefly are not repeated in the same way.

For stochastic algorithms, generally there are two types: heuristic and metaheuristic, although their difference is small. Heuristic means „to discover by trial and error". Solutions to a difficult optimization problem by stochastic algorithms can be achieved in a reasonable amount of time, but the optimal solutions are not guaranteed to be reached. Advance development of the heuristic algorithms is what is called metaheuristic. Here, meta- means „high level", and they usually perform better than simple heuristics. Moreover, all metaheuristic algorithms use specific local search and randomization.

However, randomization gives a good way to move away from the local search to the global search. Classification of metaheuristic algorithms can be done in different ways. They can be classified as: population-based and trajectory –based. Thus, Firefly algorithm is considered a population-based, which has been developed in 2009 by Xin-She Yang at Cambridge University, UK [13], and it will be used in this work as an optimization tool.

4.2 Firefly Algorithm

The fireflies produce a flashing light that make a wonderful scene in the sky of the tropical regions. There are various species of fireflies, and they produce short and rhythmic flashes. The real function of such signalling systems is still not really clear and is debated between scientists. However, attracting matting partners and attacking potential prey are the two fundamental functions of such flashes. In addition, flashing can also work as a warning mechanism for protection.

It is known that the light intensity at a certain distance r from the light source obeys the inverse square law. The light intensity I decreases with the increasing of distance r in terms of $I \propto 1/r^2$. Furthermore, the air absorbs the light which becomes weaker and weaker with increasing the distance. Because of these two factors most fireflies become visual to a limited distance, about several hundred meters in the dark, which is good enough for the communication of fireflies [13].

The Flashing light formulation is associated with the objective function to be optimized, and this makes it possible to formulate the Firefly (FA) algorithm, fig. (21).

Objective function $f(x),\ x = (x_1, x_2, \ldots, x_d)^T$
Initialize a population of fireflies $x_i (i = 1,2,..,n)$
Define light absorption coefficient γ^*
while $(t < Maximum\ Generations)$
for $i = 1{:}\,n$ *(all n fireflies)*
for $j = 1{:}\,n$ *(all n fireflies (inner loop))*
Light intensity I_i *at* x_i *is determined by* $f(x_i)$
if $(I_i < I_j)$
Move firefly i towards j in all d dimensions
else
Move firefly i randomly
end If
Attractiveness changes with distance r via $exp[-\gamma^* r]$
Evaluate new solutions and update light intensity
end for *j*
end for *i*
Rank the fireflies according to light intensity and find the current best
end while
Postprocess results and visualization

Figure 21: Code of the Firefly Algorithm (FA)

Firefly –inspired algorithm is developed by idealizing the flashing characteristics of fireflies [13]. The following three idealized rules are considered:

- All fireflies are unisex that fireflies will be attracted to each other regardless their sex.
- Attractiveness is proportional to brightness, hence; for any two flashing fireflies, the less bright will move in the direction of the brighter one. Both the attractiveness and the brightness decrease as the distance between fireflies increases. If there is no brighter firefly to attract that specific one, it will move randomly.

- The brightness or the attractiveness of a firefly is determined by the objective function.

In firefly algorithm, there are two important subjects: the change of light intensity and formulation of the attractiveness (brightness). The attractiveness of a firefly, β^*, is measured by the brightness, $I(x)$, that is related to the objective function. However, the attractiveness varies with the distance, r_{ij}, considered between firefly i and firefly j. In addition, as the distance from the source increases, the light intensity decreases [14].

The variation of light intensity $I(r)$ is according to the inverse square law, as follows:

$$I(r) = \frac{I_s}{r^2} \tag{50}$$

where I_s is the intensity of the source.

For any medium with a fixed light absorption coefficient γ^*, the light intensity I changes with the distance, r, as the following Gaussian form:

$$I = I_0 e^{-\gamma^* r^2} \tag{51}$$

where I_0 is the initial light intensity.

Whereas the attractiveness of a firefly is proportional to the light intensity and can be defined as:

$$\beta^* = \beta_0 e^{-\gamma^* r^2} \tag{52}$$

where β_0 is the attractiveness at $r = 0$,

In actual implementation, the attractiveness function $\beta^*(r)$ is as follows:

$$\beta^*(r) = \beta_0 e^{-\gamma^* r^m}, \quad m \geq 1 \tag{53}$$

The distance between any two flies i and j at x_i and x_j is the Cartesian distance:

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2} \tag{54}$$

where $x_{i,k}$ is the k^{th} component of spatial coordinate x_i of the i^{th} firefly.

In the 2-Dimensional case:

$$r_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \tag{55}$$

The attraction movement of a firefly i towards another more attractive firefly j is determined by:

$$x_i = x_i + \beta_0 e^{-\gamma^* r_{ij}^2}\left(x_j - x_i\right) + \alpha\epsilon_i \tag{56}$$

where the second term of the equation is due to attraction, whereas; the third term is due to the randomization with α, which is randomization parameter, and ϵ_i is a vector of random numbers obtained from either Gaussian distribution or uniform distribution. The parameter γ^* characterizes the variation attractiveness, where its value is essential in determining the behaviour of the FA algorithm and the speed of convergence. In theory, $\gamma^* \in [0, \infty)$, however; in practice for most applications, it varies from 0.1 to 10.

Moreover, the distance r is not restricted to the Euclidean distance, but it can be defined depending on the kind of the problem of interest. Actually, any measure that can characterize the quantity of interest in the optimization problem can be used as the distance r, [14].

4.3 Firefly Algorithm Implementation on Proposed Mechanism

In this work, the firefly (FA) algorithm is implemented, as seen in the Appendix, to optimize the length of the proposed valve variable lift mechanism. The parameters to be optimized are the lengths a_1, a_2, a_3, a_4, a_5, a_6 and, r_6; whereas, a_7, a_8, $b_1$and $d_1$are

assumed constants. Therefore, the fireflies in this case are the lengths of the links and upper and lower bounds are assigned for each parameter, as shown in table 13.

Table 13: Parameters Upper and Lower Bounds

Link	Lower Bound (mm)	Upper Bound (mm)
a_1	20	25
a_2	45	55
a_3 , a_4 , a_5 , a_6	35	45
r_6	80	85

The population of fireflies is considered five fireflies for each parameter (total of 35 fireflies) and not higher due to the long calculation time, as there are 114 points describe the output valve lift, S_v, (from INVENTOR) correspond to the input angle , θ_1, as seen in Appendix, to be calculated in e ach iteration to get the output profile. However, since there are 200 points of valve lift, S, on the objective function profile, an interpolation has been used to locate the 114 points of valve lift, S_n, calculated on the objective function profile, as follows:

$$S_n = S^* - (S^* - S) * \frac{(t^* - t_v)}{(t^* - t)} \quad (57)$$

where, S^*, is the objective valve lift at time t^*, S, is the objective valve lift at time t and S_n, is the interpolated objective valve lift at time t_v of the calculated valve lift.

Thus, the error, ε, between the calculated valve lifts, S_v, and the interpolated objective valve lifts, S_n, has been calculated, as follows:

$$\varepsilon = |S_n - S_v| \quad (58)$$

Firstly, the 35 fireflies are uniformly distributed within the lower and the upper bounds of links lengths: a_1, a_2, a_3, a_4, a_5, a_6 and, r_6. Afterword, the error is calculated for each point of cam rotation, φ. There are 200 points, appoints 200 degrees of cam rotation (rise and fall), before the cam dwell occurred. Since INVENTOR simulation gives time in seconds and not in cam rotation angle, it has been changed to degrees. Therefore, the error, ε, is calculated for each degree of cam rotation, and the total error for the 114 points is then determined. The attractiveness (brightness) of the fireflies is related to the smallest total error, and the new links lengths are updated and attracted to fireflies that give the smallest total error. Due to the long-time calculations, the number of calculated points is reduced to 57 points, and this does not have an effect on the optimization results, as it has been checked with 114 points, as shown in the second optimization, table 16. Moreover, a comparison has been done between the calculated profiles of the first and second optimizations and it shows a total consistency, as shown in fig. (27). Hence, it means that 57 points are enough to perform the optimization.

The attraction movement of a firefly (new link length), i, to another more attractive firefly (the link length that gives the smallest total error), j, is determined by:

$$new\ link\_length_i = link\_length_i + \beta_0 e^{-\gamma^* r_{ij}^2}\left(link\_length_j - link\_length_i\right) + \alpha\epsilon_i \tag{59}$$

$$\epsilon_i = e - 0.5 \qquad e: is\ \ a\ random\ numbers\ by\ uniform\ distribution \quad e \in [0, +1] \tag{60}$$

Since r_{ij} is not a Cartesian distance, so that the measure that can well characterize the quantities of interest (length of links) in this optimization problem is:

$$r_{ij} = \left\| link\_length_j - link\_length_i \right\| / d_{max} \quad (61)$$

$$d_{max} = \| U_b - U_l \| \quad (62)$$

where,

$$U_b: Upper\ bound\ for\ all\ lingths \qquad U_b = 85\ mm$$

$$U_l: Lower\ bound\ for\ all\ lingths \qquad U_l = 20\ mm$$

The FIREFLY algorithm parameters are taken from the work of *Xin-She Yang* (2010) [14], as in table (14).

Table 14: Firefly Algorithm parameters

Parameter	Description	Value	Range
γ^*	Attractiveness variation coefficient	1.0	[0.1-10]
β_0	Attraction coefficient at $r = 0$	0.4	
α	Randomization parameter	0.25	[0.0-1.0]
α_{damp}	Randomization parameter damping ratio	0.98	(0-1.0)

However, randomization parameter, α, can be reduced each iteration as to reduce the randomness, this is called damping.

$$(\alpha)_{new} = \alpha * \alpha_{damp} \quad (63)$$

The objective function is the total error, $\varepsilon_R = f(a_1, a_2, a_3, a_4, a_5, a_6 and\ r_6)$:

$$\varepsilon_R = sum(\varepsilon) \quad (64)$$

which is the error sum of all the 57 points.

The code of firefly optimization works as follows; as shown in the flowchart figure (22). Firstly, the algorithm parameters and the upper and lower bounds of the required mechanism parameters (lengths of links) are entered. At that point, the 35 fireflies are randomly initialized by using uniform distribution, and the total error of each population of 7 fireflies correspond to the seven required parameters (lengths) is calculated by calling a function of our loop closure equation solution. Then the population of fireflies of the smallest total error is assigned initially as the best solution.

Finally, In every iteration loop of optimization, the total error of each population of fireflies (7 fireflies) is compared to the total errors of the other four populations (28 fireflies) and the largest errors are located and their fireflies are scaled (moved) towards the one which is compared to with the relation (59), then their total error is again calculated, by calling the function of LCE, and compared to the memorized smallest error and the smallest of them (population of 7 fireflies) is selected as the best solution. However, the population (7 fireflies) of the smallest total error is considered a new population of fireflies and, at the end of the loop there will be 5 new populations of fireflies (35 fireflies), which will be added to the previous 5 old populations (35 fireflies), then all will be sorted according to the minimum error and only the ones corresponding the smallest five total errors are selected; therefore, entered the next loop. However, at the end of each iteration the best minimum total error, the best optimized lengths and the time required to execute that optimization loop are printed. Additionally, the damping randomization parameter, α, is updated to get the smallest new value.

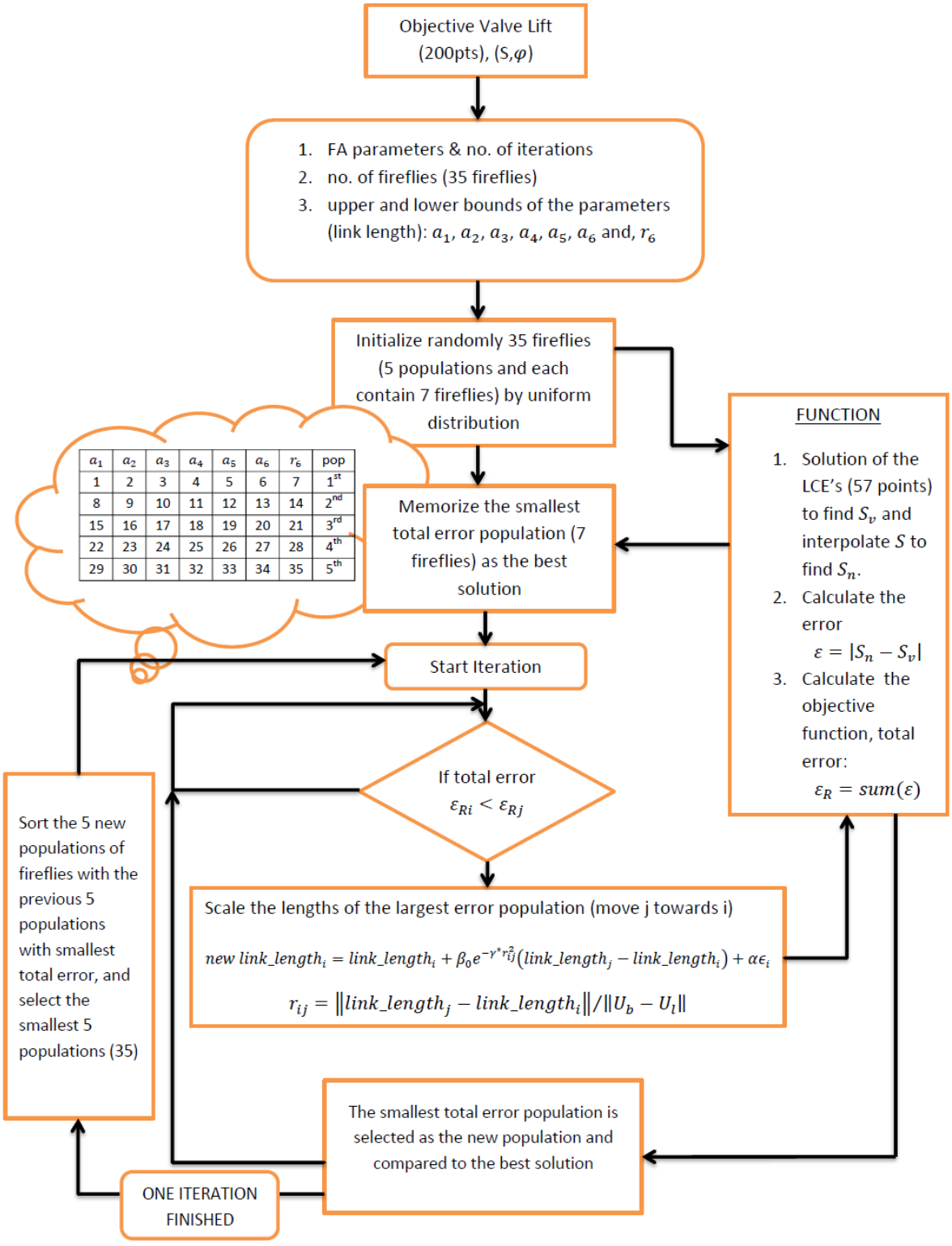


Figure 22: Flow Chart of the Firefly Algorithm

4.4 Optimization Results

Various executions of the firefly MATLAB script M-file for the optimization of links have been carried out. The results of the mechanism lengths optimization, number of iterations and the required run time are obtained. Moreover, a graph showing the reduction of the total error with the number of iteration for each optimization is plotted. Also, the main MATLAB code for the kinematic solution is executed using the obtained optimized lengths of links and the profile of the valve lift, S, is plotted and compared to the objective valve lift profile of 10mm.

Two optimizations are carried out for the same firefly algorithm parameters, but with different number of angle , θ_1, 114 and 57 points, as follows:

1. First Optimization for the full 114 points of valve lift, as shown in table (15).

Table 15: First Optimization and Parameters Results

Parameters	Value	Link	Optimized Length (mm)
γ^*	1.0	a_1	23.6140
β_0	0.4	a_2	53.3428
α	0.25	a_3	43.6912
α_{damp}	0.98	a_4	40.8299
Iteration no.	14	a_5	37.8992
Min. Total Error (mm)	7.4959	a_6	37.0149
Run Time (s)	268763.74	r_6	81.2611

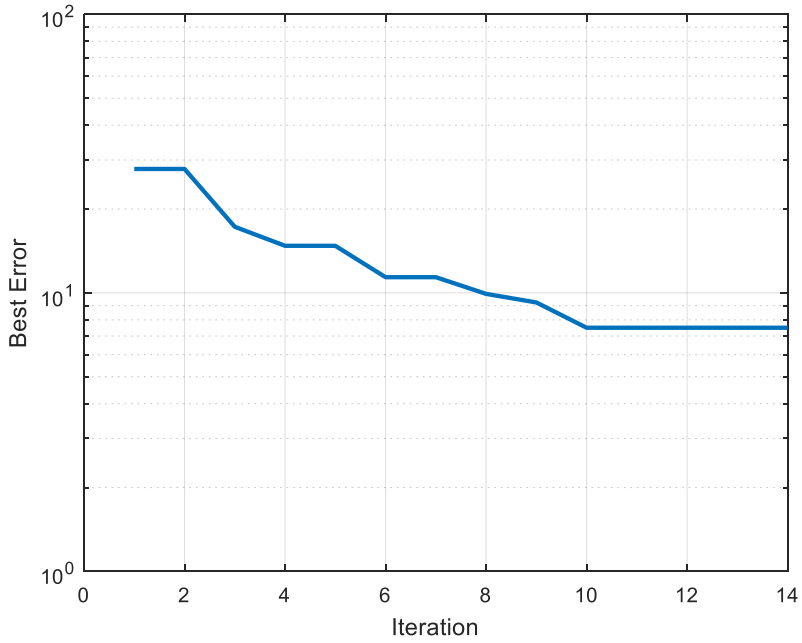


Figure 23: Total Error Reduction with Number of Iterations (1st Optimization)

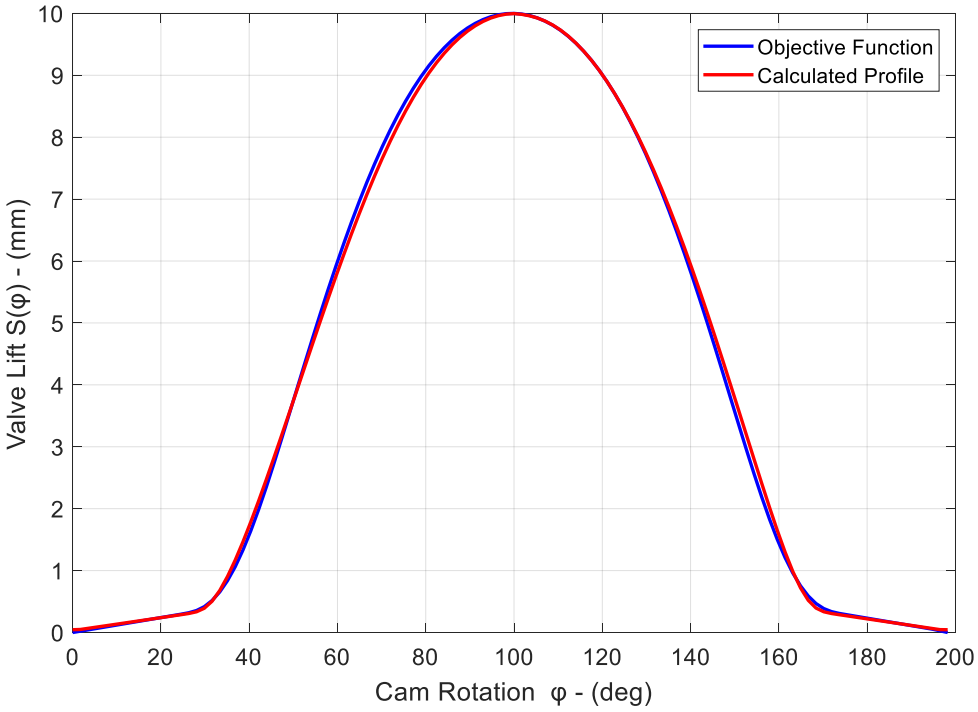


Figure 24: Valve Lift Profile Comparison (1st Optimization)

2. Second Optimization for only 57 points of valve lift, as shown in table (16).

Table 16: Second Optimization Parameters and Results

Parameters	Value	Link	Optimized Length (mm)
γ^*	1.0	a_1	23.6196
β_0	0.4	a_2	53.3117
α	0.25	a_3	43.8314
α_{damp}	0.98	a_4	40.7127
Iteration no.	19	a_5	37.9492
Min. Total Error (mm)	3.7879	a_6	36.9320
Run Time (s)	121377.51	r_6	81.2476

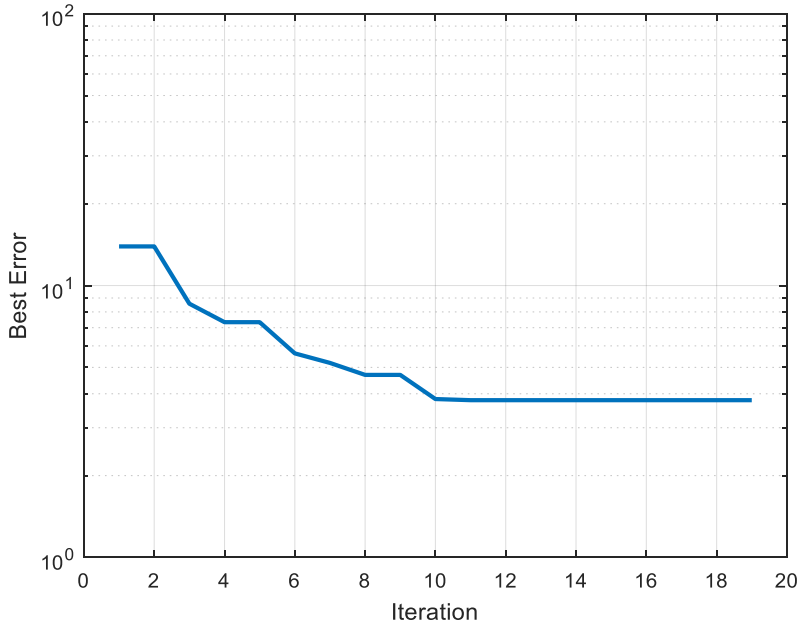


Figure 25: Total Error Reduction with Number of Iterations (2nd Optimization)

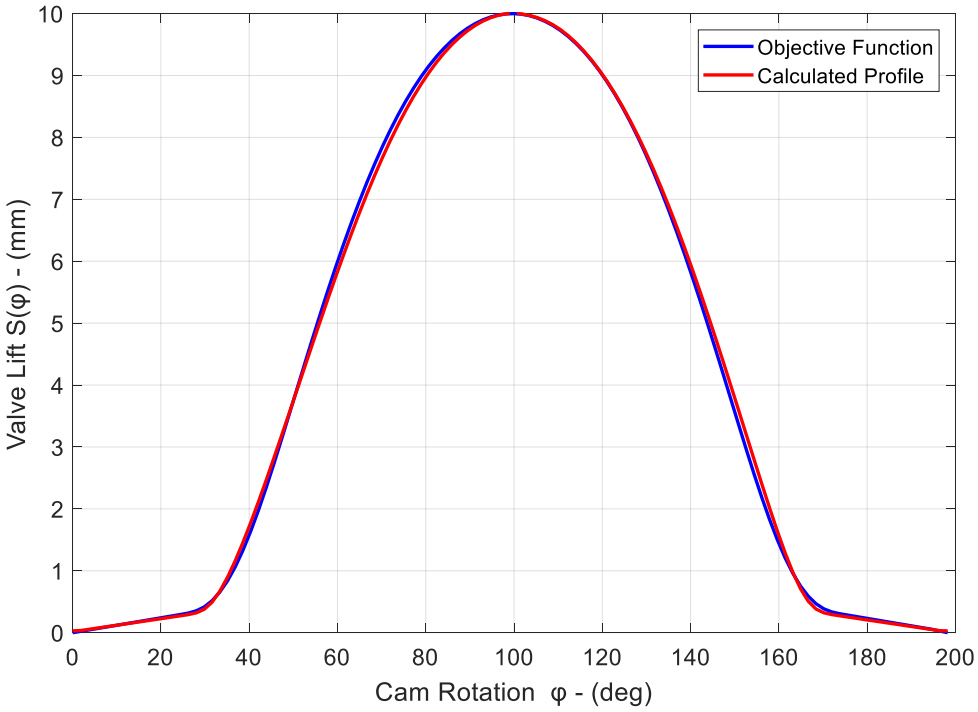


Figure 26: Valve Lift Profile Comparison (2nd Optimization)

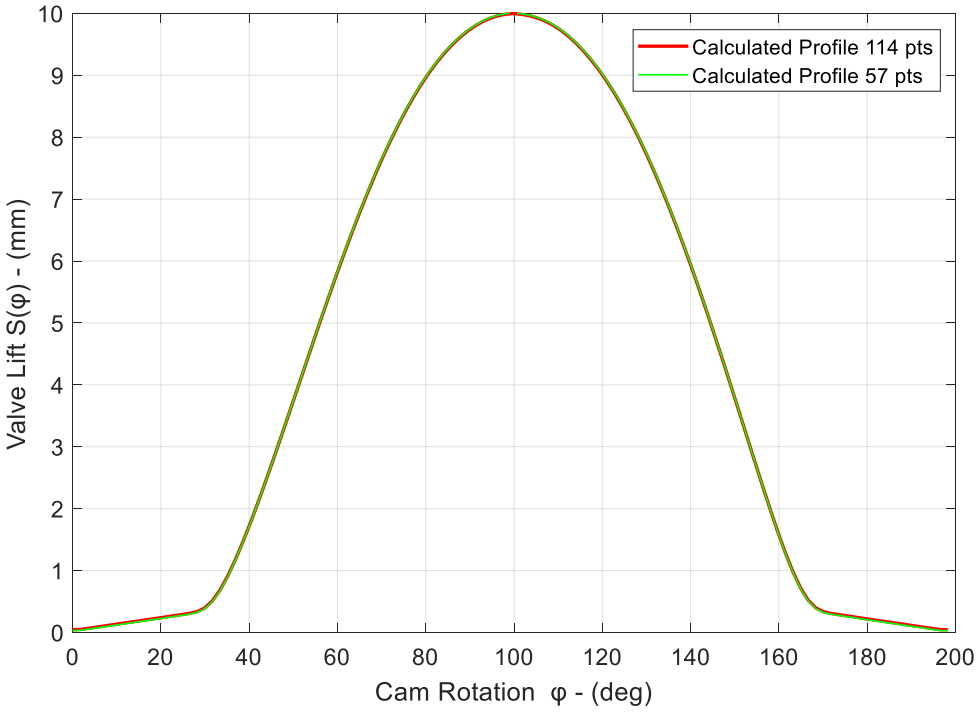


Figure 27: Valve Lift Profile Comparison of 1st and 2nd Optimizations

From the results, the minimum total error is $3.7879\ mm$ for the 57 points corresponds to a total error of about 7.4959 mm for the 114 points, which is logical as it is double the quantity.

Finally, the optimized Lengths of links are shown in table (17).

Table 17: The Optimized Lengths of Links

Link	Optimized Length (mm)
a_1	23.6196
a_2	53.3117
a_3	43.8314
a_4	40.7127
a_5	37.9492
a_6	36.9320
r_6	81.2476

CHAPTER 5

RESULTS AND DISCUSSION

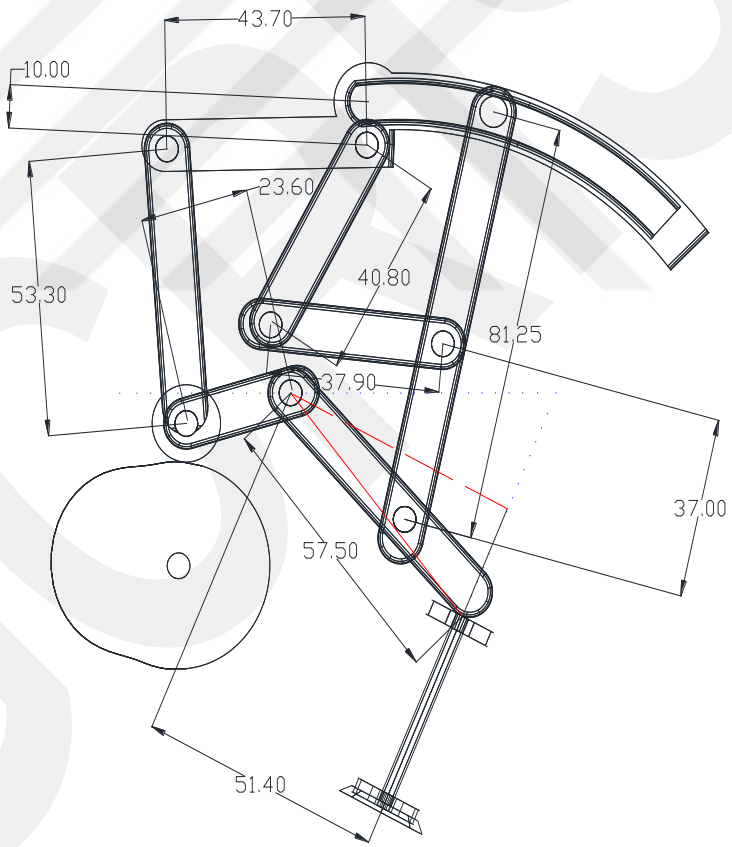


Figure 28: The Proposed Optimized Mechanism

Firstly, family of various valve lifts have been calculated by MATLAB, as seen in Appendix, for the new optimized lengths, and the results are plotted in figure (29).

Actually, the valve lifts profiles at specific push rod positions (γ angles) are calculated by MATLAB in order to get several maximum valve lifts, as shown in table (18).

Table 18: Maximum Valve Lifts at Different Values of γ

Auxiliary mechanism angle (γ)	Maximum Valve Lift (mm)
28.55°	10.0
30.95°	9.0
33.44°	8.0
35.98°	7.0
38.63°	6.0
41.35°	5.0
44.20°	4.0
47.14°	3.0

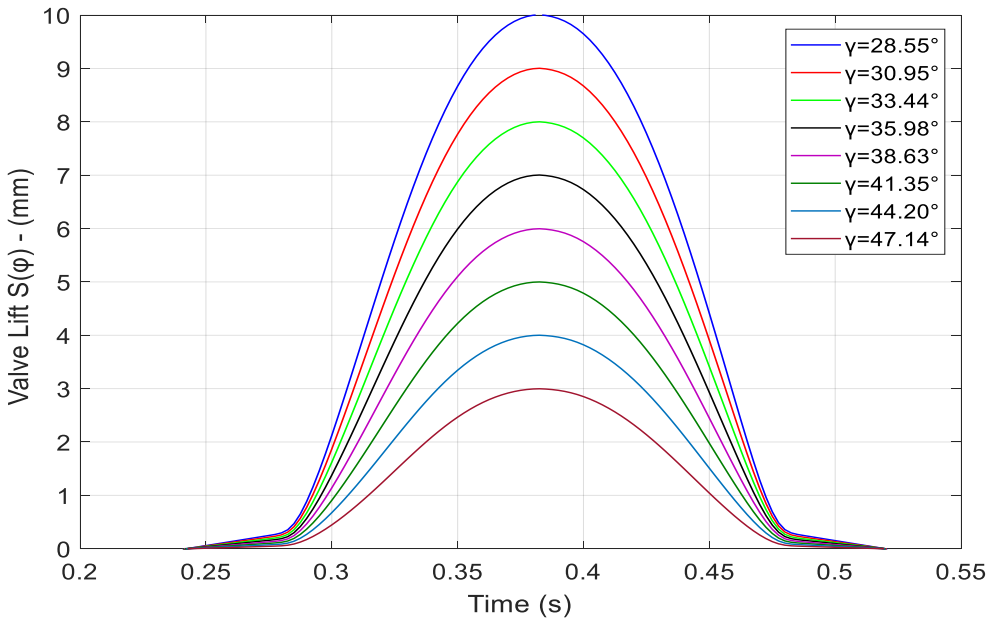


Figure 29: The Family of Valve Lifts at Various Push Rod Positions, γ, by Matlab for Optimized Mechanism

Since the optimization was done at only the $10\ mm$ valve lift at the angle, $\gamma = 28.55°$, a check must be done to see whether the other valve lifts are valid or not. Therefore, a comparison of the various valve lifts $(10\ mm, 9\ mm, 8\ mm\ \&7\ mm)$ with the objective values has been performed, as shown in fig. (30). It can be noticed that the eror is almost the same and the profiles are totally consistent.

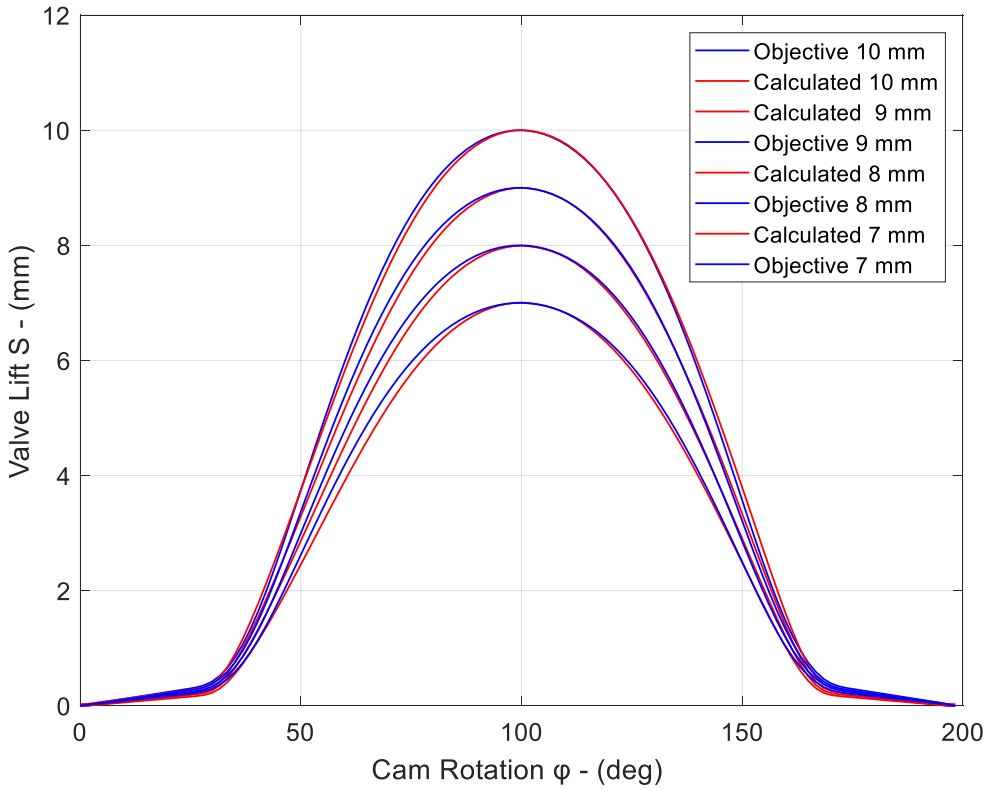


Figure 30: Family of Valve Lifts Comparison with the Objective Functions

Additionally, a comparison of the various valve velocities and accelerations at the lifts $(10\ mm, 9\ mm, 8\ mm\ \&7\ mm)$ with the objective values has been performed, as shown in fig. (31 & 32).

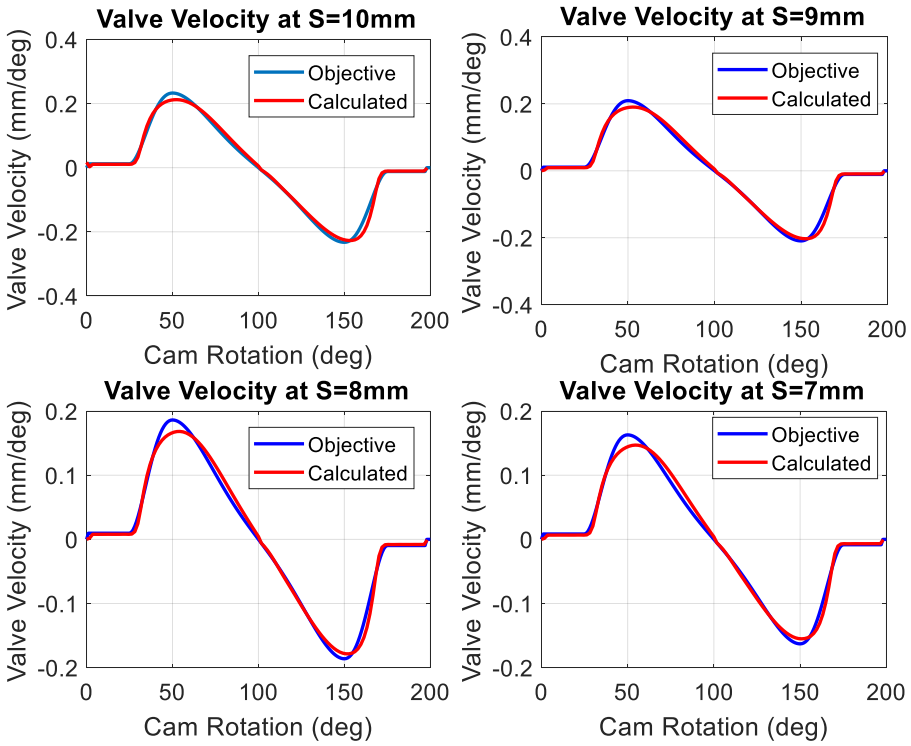


Figure 31: Family of Valve Velocity Comparison with the Objective Functions

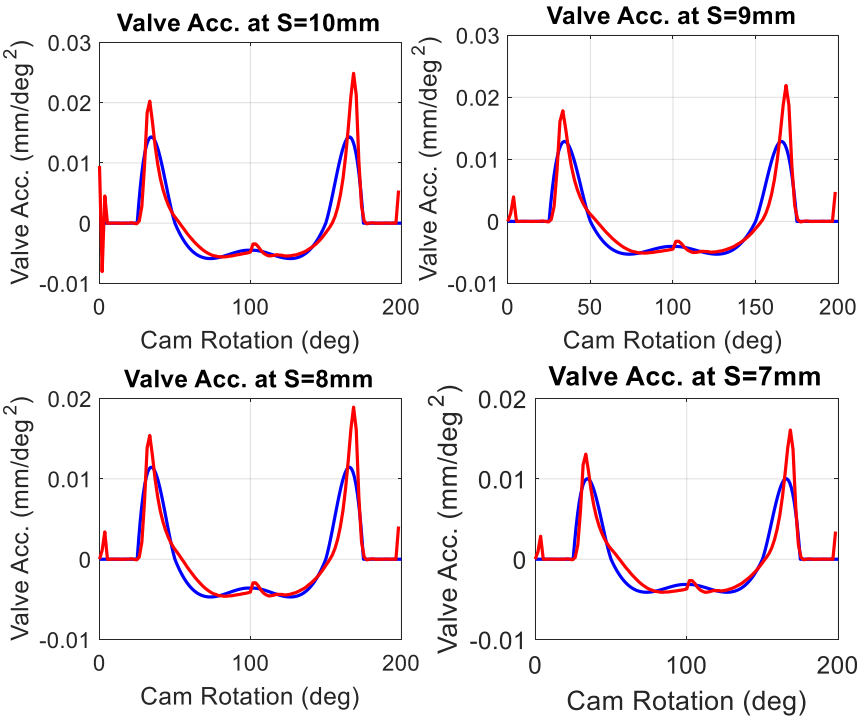


Figure 32: Family of Valve Acceleration Comparison with the Objective Functions

The following table (19) shows the total error of each valve lift.

Table 19: The Total Error at Various Valve Lift

γ	Maximum Valve Lift (mm)	Total Error for 114 pts (mm)
28.55°	10.0	7.5868
30.95°	9.0	7.5792
33.44°	8.0	8.1135
35.98°	7.0	8.5209

Further, the error variation of the valve lifts with the cam rotation angle, φ, has been plotted in a family, as seen in fig. (33).

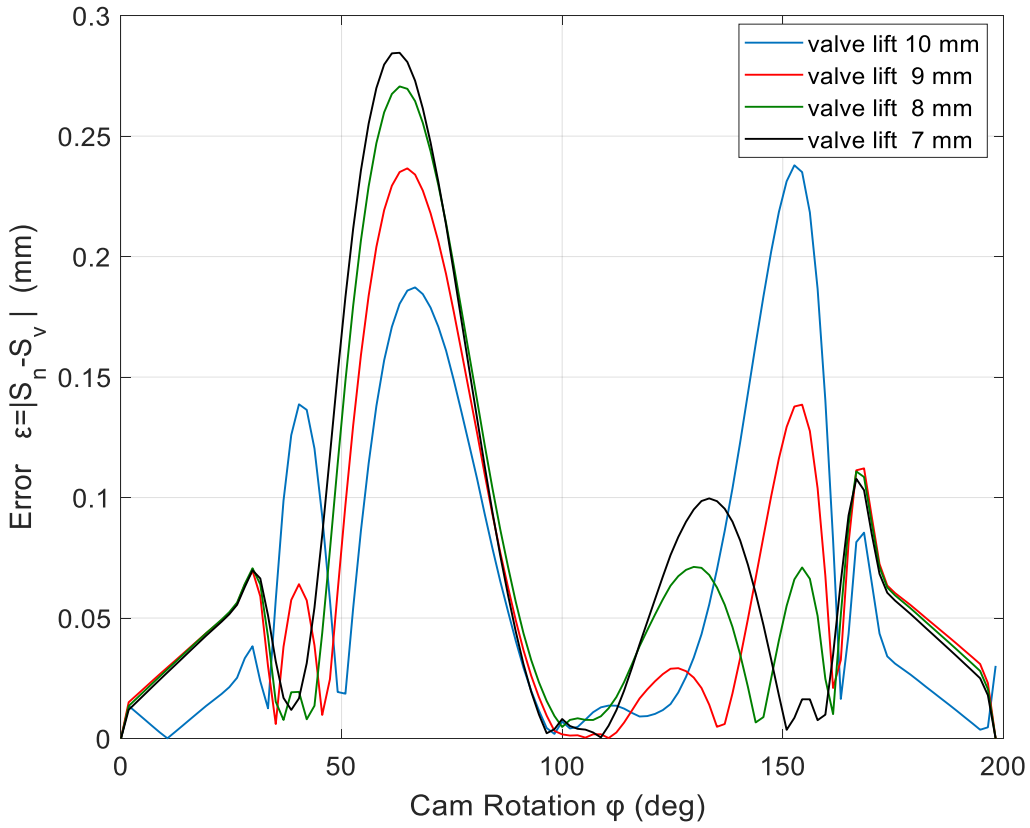


Figure 33: Family of Error Variation of the Valve Lifts with Cam Rotation Angle, φ

The mechanism is constructed again in INVENTOR with the obtained optimized lengths, as shown in figure (34). All the required constraints and joints have been selected and the mechanism with optimized lengths has been constructed. Consequently, several simulations have been performed with varying the push rod positions, γ, by using the auxiliary mechanism each time. To do so, each time link 4 is ungrounded and rotated manually to obtain the desired value of angle γ, then grounded again and fixed.

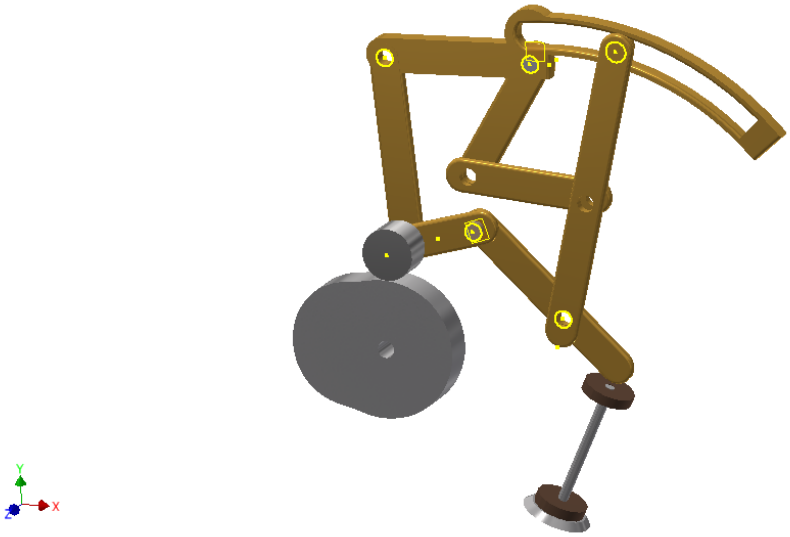


Figure 34: The Optimized Proposed VVL Mechanism Built in INVETOR

Eight simulations are carried out according to γ values and the results of valve lifts, horizontal displacement of rocker arm are plotted, as shown in figures (35& 36).

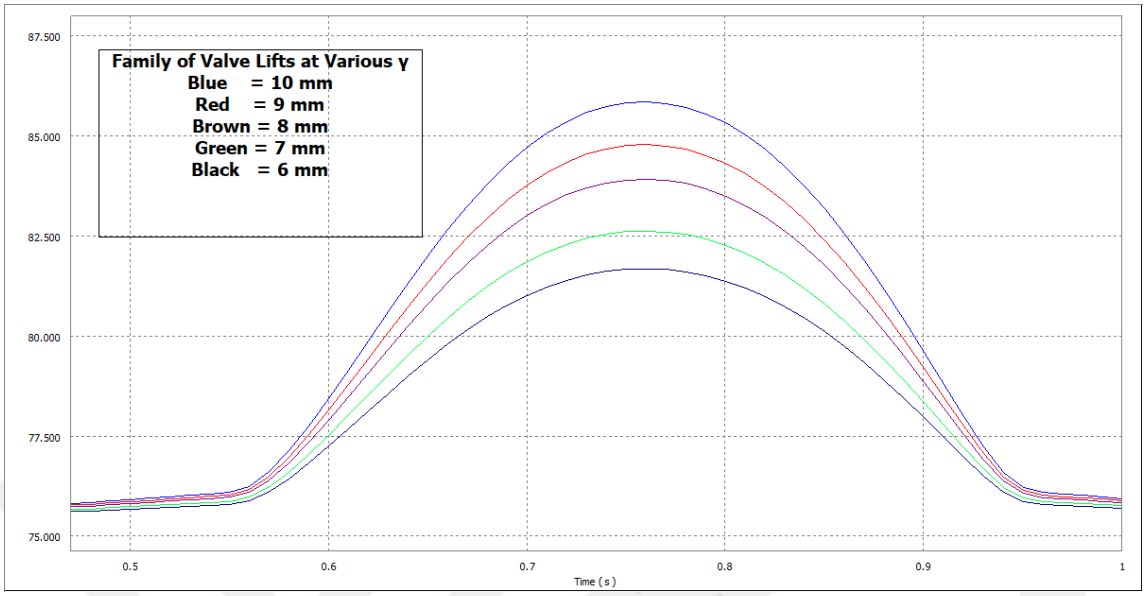


Figure 35: The Family of Valve Lifts at Various Push Rod Positions, γ, by Inventor

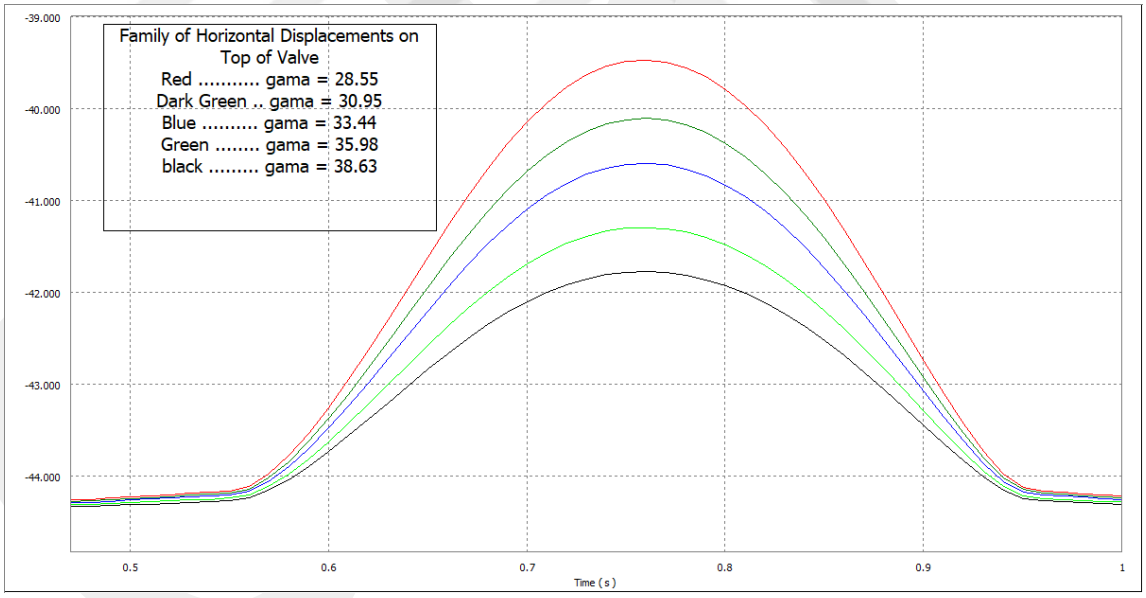


Figure 36: The Family of The Horizontal Displacement of Rocker Arm at Various Positions of γ by Inventor

Moreover, the valve velocities and accelerations are plotted in families, as shown in figures (37 & 38), consequently.

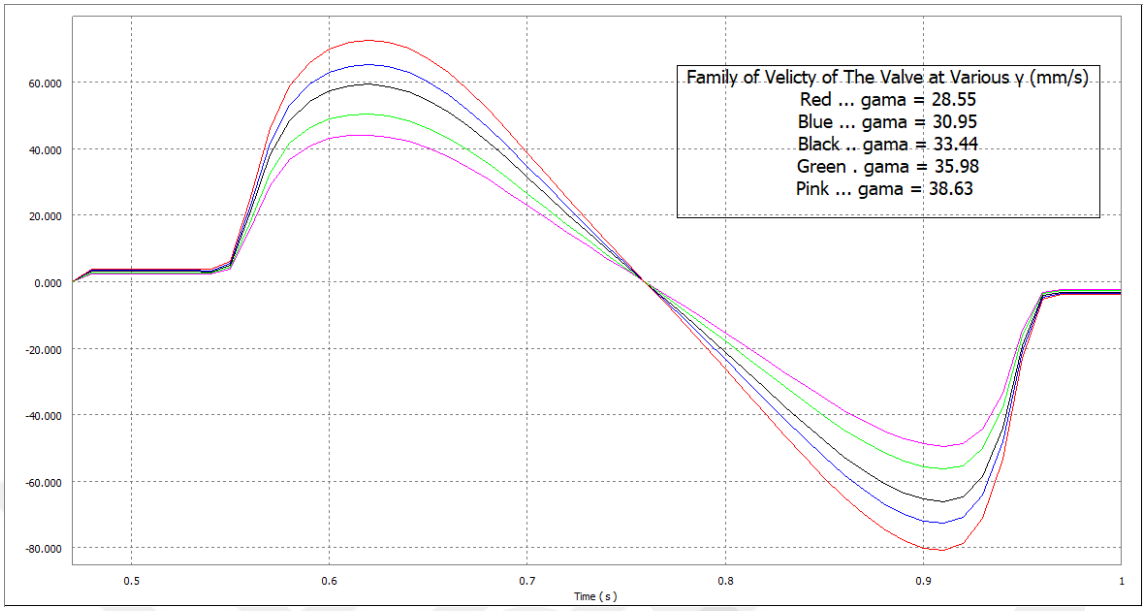


Figure 37: The Family of The Velocity of Valve at Various Positions of γ by Inventor

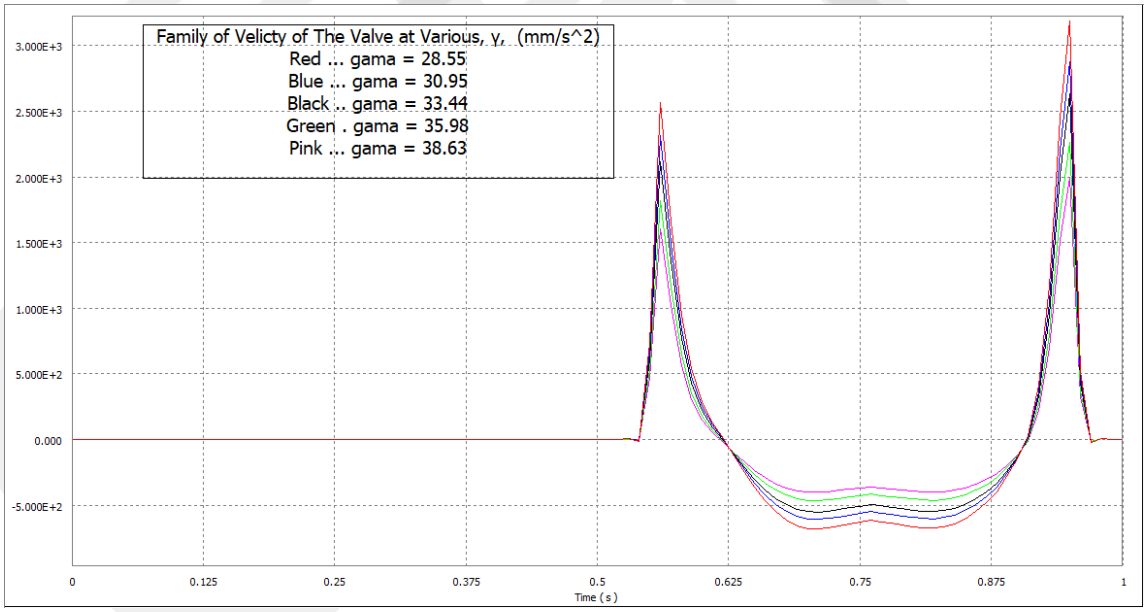


Figure 38: The Family of The Acceleration of Valve at Various positions of γ by Inventor

In addition, the velocity and acceleration of the valve at the maximum opening of 10 mm at $\gamma = 28.55°$, can be compared to the objective. Therefore in order to do so, the

data of the valve velocity and acceleration, as show in tables (21 & 22) in Appendix, at the maximum opening of 10 mm have been taken from the INVENTOR and drawn in MATLAB, whereas the units have been changed from mm/s and mm/s^2 to mm/deg and mm/deg^2 to ease the comparison.

On the other hand, the velocity and the acceleration has been calculated in MATLAB from the valve lift values already obtained by the kinematic analysis at the maximum opening of 10 mm while considering constant speed of rotation, and can be found from the following equations:

$$V = \frac{S_i - S_{i-1}}{t} \quad (65)$$

$$a = \frac{V_i - V_{i-1}}{t} \quad (66)$$

Where, S_i is the valve lift at the calculated instant in mm.

t, is the time interval between two consecutive points $(i - 1\ and\ i)$, where

$$t = \frac{200\ degrees}{114\ points} = 1.754385965\ degree$$

The results of comparison have been drawn in figures (39 & 40).

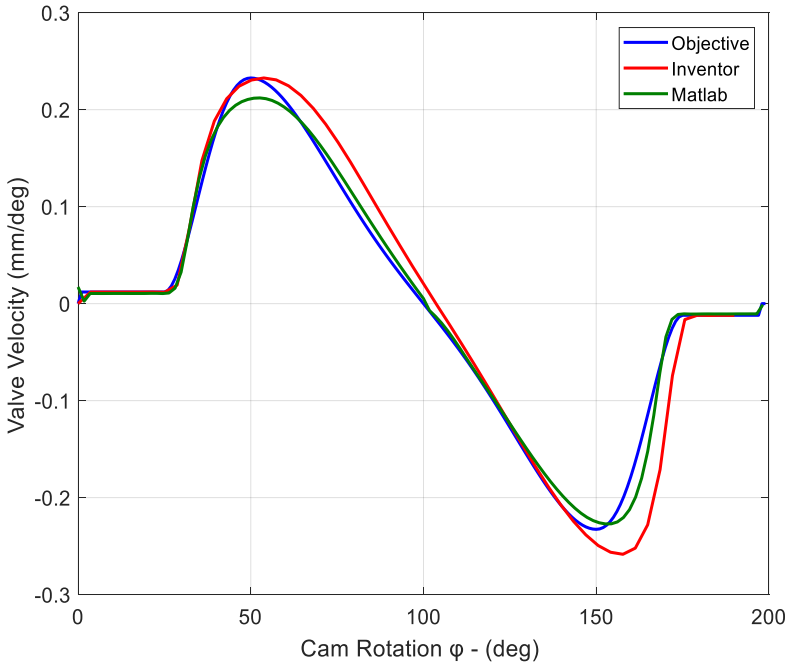


Figure 39: Comparison between Objective, Inventor and Matlab Valve Velocity at Valve Lift of 10 mm

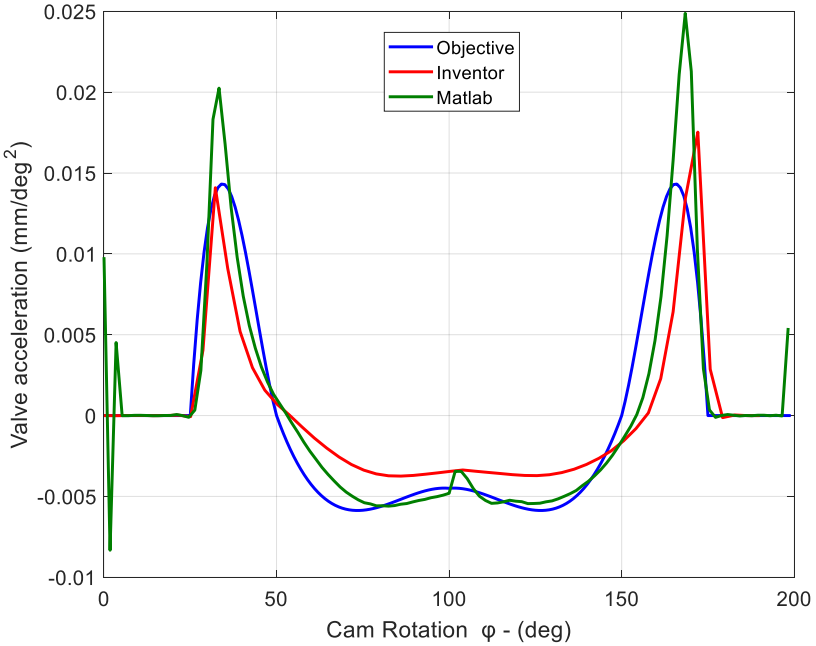


Figure 40: Comparison between Objective, INVENTOR and MATLAB Valve Acceleration at Valve Lift of 10 mm

Clearly, there is a consistency in the valve opening velocity and closing velocity between Matlab and the objective velocity, but the Inventor result show a little difference in the closing phase of the valve. Whereas, the acceleration comparison show that the Inventor results are better than Matlab, as the Matlab show a large difference in the maximum values of acceleration. However, MATLAB acceleration results are closer to the objective values than the INVENTOR in the minimum part.

A further detailed result is the comparison between the valve lift results of MATLAB and INVENTOR with the objective values of valve lift at the maximum height of 10mm, as shown in fig. (41). Obviously, MATLAB results agreed appropriately with the objective values of valve lift, while the INVENTOR results shows some deviation from the objective values in the rise and fall phase of the cam.

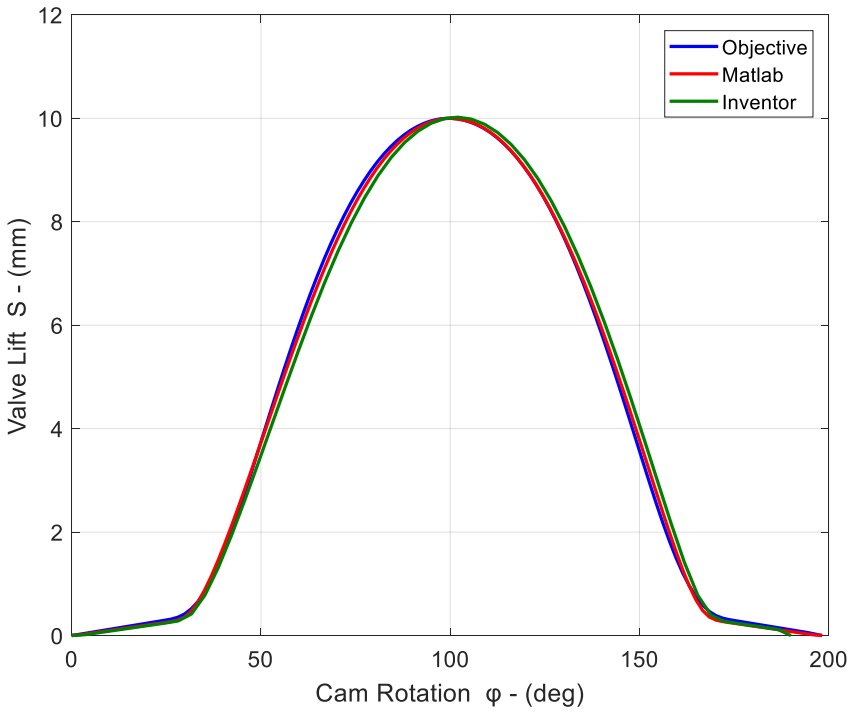
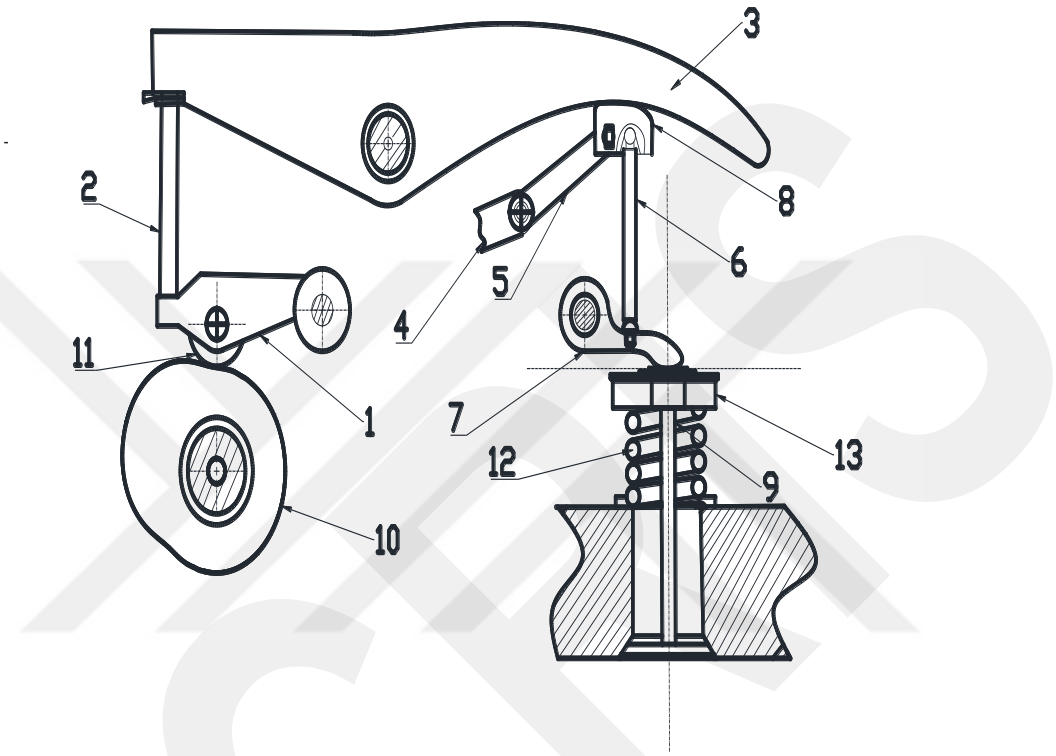


Figure 41: Comparison between Objective, INVENTOR and MATLAB Valve Lift Profile of 10 mm

Moreover, an ideation of OHC mechanism has been drawn without specified dimension, as shown in fig. (42). The idea is based on the proposed mechanism and devised from the work of Mihalcea Stelian on the OHV mechanism, [1].



No.	Description
1	Link 1
2	Link 2
3	Link 3 (Circular Arc for VVL System)
4	Auxiliary Link 4 (any actuation) – VVL
5	Auxiliary Link 5- VVL
6	Push Rod
7	Rocker Arm
8	Sliding Joint of Push Rod – VVL
9	Intake Valve
10	Cam with 8mm Valve Lift
11	Follower
12	Spring
13	Washer with Spring Holder

Figure 42: Ideation of the OHC Mechanism

CHAPTER 6

CONCLUSION

The variable valve lift system, commonly used for varying the opening of the intake valve in automobile engines, has been studied in this work. The schematic proposed mechanism has been built in the INVENTOR software. However, the dimensions were assumed at the beginning to check the workability of the mechanism. Hence, different simulations have been performed and good consistent results have been found with a small error as compared to the solution of the kinematic synthesis performed by MATLAB. It is important to mention that the cam was designed based on an objective valve lift profile obtained from the work of Professor Gordon Blair on race engines [10], but the cam was designed on a maximum valve lift of 8 mm and not 10 mm in order to vary the valve lift and obtain higher valve lifts up to 10 mm and lesser up to 3 mm. Therefore, this shows one of the economic advantages of using the VVL system in reducing the dimension of the cam and also the amount material used, especially if mass production has been considered.

Moreover, the Firefly optimization has been used to optimize the lengths of links of the proposed mechanism. The minimum error between the objective function of valve lift of maximum opening of 10 mm and the valve lift calculated from the solution of the

kinematic loop closure equations is considered as the attractiveness of the fireflies. Two optimizations have been performed. The first one was time consuming operation since there are 114 points to be calculated for every iteration with various fireflies. While the second one was carried out with only 57 points and the results were almost the same and the optimized lengths have been selected.

Thus, the value of the auxiliary mechanism angle γ which is responsible for varying the valve lift has be selected, using the optimized lengths found by FA algorithm, in such a way to give varying maximum valve lifts, 3, 4, 5, 6,7,8,9 and 10 mm. Consequently, the mechanism was built again in INVENTOR with the optimized lengths and various simulations, with varying γ angle to get the various valve lift profiles, have been carried out. Both MATLAB and INVENTOR results of families of valve lifts have shown good consistency. However, the velocity and the acceleration of the valve have been calculated in INVENTOR and the family of velocity and acceleration curves have been plotted. Since there is the objective velocity and acceleration only at a valve lift of maximum opening of 10 mm, a comparison has been done between MATLAB and INVENTOR with the objective values, and good results obtained at the valve opening phase and reasonable results are found at the closing phase. Furthermore, a comparison between the valve lift profiles at the height of 10 mm has been carried out. It showed that the MATLAB results are very close to the objective values whereas Inventor results shows a little deviation in the valve opening and closing phase, and this could be due to an error in INVENTOR constraints.

Finally, this work has proved the workability of the proposed schematic mechanism but a final real mechanism could be a future work based on the mechanical idea of this VVL system and using a mechanical, hydraulic or electrical actuation to select the best angle γ that will give the required maximum intake valve opening that is suitable for the demand of the engine. In addition, an ideation of OHC mechanism has been drawn without specified dimension.

REFERENCES

1. **Mihalcea Stelian**, (2010), "*A Kinematic Analysis of the Variable Valve Timing Mechanism with Three Elements and Continuous Valve Lift*", ANALELE UNIVERSITĂTII , Anul XVII, No. 1, ISSN 1453 – 7397.
2. **Xiaozhen Q., Dojoong K.,** (2012), *"Kinematic Design and Analysis of a Four-Bar Linkage-Type Continuously Variable Valve Actuation Mechanism",* J Mechanism and Machine Theory, Vol. 57, pp. 111-125.
3. **G. Dritsas, P. Nikolakopoulos, C. Papadopoulos,** (2013) *"Design Evaluation of a Follower Cam with Variable Valve Lift Mechanism"*, International Journal of Structural Integrity, Vol. 4 Issue: 1, pp.7-22.
4. **Albatlan et al.,** (2014), *"Dynamic Analysis and Experimental Evaluation of Variable valve Lift System for Internal Combustion Engine with Double overhead Camshaft",* Int. J. Vehicle Structures & Systems, vol. 6(1-2), pp. 24-31.
5. **S. Mihalcea, N. Stanescu, D. Popa,** (2015), *"Synthesis and Kinematic and Dynamic Analysis of a Variable Valve Lift Mechanism with General Contact Curve",* J Multi-body Dynamics, Vol. 229(1), pp. 65–83.
6. **Carmelina Abagnale et. al.,** (2015), *"New Mechanical Variable Valve Actuation Systems for Motorcycle Engines",* J. Mechanical Engineering Science, Vol. 229(4), pp. 716–734.
7. **Megha M., Eknath A., Santosh T.,** (2016), "*Numerical Analysis of Continuous Variable Valve Lift (CVVL) Mechanisms for Throttle Free Load Control of SI Engine",* International Journal of Engineering Research and Development, Vol.12, Issue 12, pp. 45-55.
8. **Adrian C., Vasile H., Nioclae-Doru S., Adrian B., Rodica N.,** (2017), *"Analysis Synthesis and Computer-Aided Kinematic Analysis of a Continuously*

9. *Variable Lift Mechanism",* J Mechanical Engineering Science, Vol. 231(2), pp. 309–325.
10. **National Research Council,** (2015), "*Cost, Effectiveness, and Development of Fuel Economy Technologies for Light-Duty Vehicles*". The National Academies Press, Washington, DC.
11. **G. P. Blair, Charlie D. McCartan, H Hermann,** (2005), *"The Right Lift",* Race Engine Technology, Issue 009.
12. **Robert L. Norton**, (1999), "Design of Machinery: *An Introduction to The Synthesis and Analysis of Mechanisms and Machines* ", McGraw-Hill, New York.
13. **Joseph E. Shigley, John J. Uicker,** (1980), "Theory of Machines and Mechanisms ", McGraw-Hill, New York.
14. **Xin-She Yang,** (2010), "Nature-Inspired Metaheuristic Algorithms ", Luniver Press, United Kingdom.
15. **Xin-She Yang,**(2010), *"Firefly Algorithm, Stochastic Test Functions and Design Optimisation"* Int. J. Bio-Inspired Computation, Vol. 2, No. 2, pp.78–84.
16. **Xin -She Yang,** (2009) *"Firefly algorithms for multimodal optimization"*, in: Stochastic Algorithms: Foundations and Applications, SAGA 2009, Lecture Notes in Computer Sciences, Vol. 5792, pp. 169-178.

APPENDIX

Table 20: The Angle of Rotation of Link 1 Obtained from INVENTOR

No.	Time (s)	Revolution in degrees From INVENTOR	$\theta_1\ (degrees)$
1	0.24000	86.94640	193.830
2	0.24250	86.93640	193.820
3	0.24500	86.89566	193.779
4	0.24750	86.85500	193.739
5	0.25000	86.81443	193.698
6	0.25250	86.77380	193.657
7	0.25500	86.73310	193.617
8	0.25750	86.69240	193.576
9	0.26000	86.65180	193.535
10	0.26250	86.61130	193.495
11	0.26500	86.57080	193.454
12	0.26750	86.53030	193.414
13	0.27000	86.48940	193.373
14	0.27250	86.44857	193.332
15	0.27500	86.40850	193.292
16	0.27750	86.36620	193.250
17	0.28000	86.30520	193.189
18	0.28250	86.18110	193.065
19	0.28500	85.93510	192.819
20	0.28750	85.55640	192.440
21	0.29000	85.07080	191.954
22	0.29250	84.50710	191.391
23	0.29500	83.88770	190.771
24	0.29750	83.22980	190.113
25	0.30000	82.54620	189.430
26	0.30250	81.84650	188.730
27	0.30500	81.13830	188.022
28	0.30750	80.42733	187.311
29	0.31000	79.71810	186.602

No.	Time (s)	Revolution in degrees From INVENTOR	$\theta_1\ (degrees)$
30	0.31250	79.01380	185.897
31	0.31500	78.31740	185.201
32	0.31750	77.63200	184.516
33	0.32000	76.96030	183.844
34	0.32250	76.30510	183.189
35	0.32500	75.66880	182.552
36	0.32750	75.05340	181.937
37	0.33000	74.46090	181.345
38	0.33250	73.89290	180.777
39	0.33500	73.35080	180.234
40	0.33750	72.83590	179.720
41	0.34000	72.34920	179.233
42	0.34250	71.89140	178.775
43	0.34500	71.46310	178.347
44	0.34750	71.06470	177.948
45	0.35000	70.69610	177.580
46	0.35250	70.35750	177.241
47	0.35500	70.04850	176.932
48	0.35750	69.76890	176.653
49	0.36000	69.51825	176.402
50	0.36250	69.29590	176.180
51	0.36500	69.10144	175.985
52	0.36750	68.93420	175.818
53	0.37000	68.79360	175.677
54	0.37250	68.67920	175.563
55	0.37500	68.59040	175.474
56	0.37750	68.52680	175.410
57	0.38000	68.48790	175.372
58	0.38250	68.47310	175.357
59	0.38500	68.49360	175.377
60	0.38750	68.53140	175.415
61	0.39000	68.58880	175.472
62	0.39250	68.66890	175.553
63	0.39500	68.77410	175.658
64	0.39750	68.90580	175.789
65	0.40000	69.06500	175.949
66	0.40250	69.25170	176.135
67	0.40500	69.46570	176.349
68	0.40750	69.70670	176.590
69	0.41000	69.97520	176.859

No.	Time (s)	Revolution in degrees From INVENTOR	$\theta_1\ (degrees)$
70	0.41250	70.27160	177.155
71	0.41500	70.59680	177.480
72	0.41750	70.95110	177.835
73	0.42000	71.33480	178.218
74	0.42250	71.74790	178.632
75	0.42500	72.19060	179.074
76	0.42750	72.66280	179.546
77	0.43000	73.16430	180.048
78	0.43250	73.69490	180.579
79	0.43500	74.25430	181.138
80	0.43750	74.84170	181.725
81	0.44000	75.45650	182.340
82	0.44250	76.09780	182.981
83	0.44500	76.76450	183.648
84	0.44750	77.45517	184.339
85	0.45000	78.16800	185.052
86	0.45250	78.90070	185.784
87	0.45500	79.65070	186.534
88	0.45750	80.41520	187.299
89	0.46000	81.19060	188.074
90	0.46250	81.97110	188.855
91	0.46500	82.74780	189.631
92	0.46750	83.50770	190.391
93	0.47000	84.23250	191.116
94	0.47250	84.89620	191.780
95	0.47500	85.46520	192.349
96	0.47750	85.90062	192.784
97	0.48000	86.17330	193.057
98	0.48250	86.30440	193.188
99	0.48500	86.36660	193.250
100	0.48750	86.40940	193.293
101	0.49000	86.44990	193.334
102	0.49250	86.49120	193.375
103	0.49500	86.53250	193.416
104	0.49750	86.57340	193.457
105	0.50000	86.61420	193.498
106	0.50250	86.65520	193.539
107	0.50500	86.69620	193.580
108	0.50750	86.73730	193.621
109	0.51000	86.77830	193.662

No.	Time (s)	Revolution in degrees From INVENTOR	θ_1 $(degrees)$
110	0.51250	86.81920	193.703
111	0.51500	86.86020	193.744
112	0.51750	86.90110	193.785
113	0.52000	86.94220	193.826
114	0.52250	86.94640	193.830

Table 21: Valve Velocity Data Obtained from INVENTOR

No.	Time (s)	Velocity (mm/s)
1	0.47000	0.00000
2	0.48000	3.72202
3	0.49000	3.71816
4	0.50000	3.72547
5	0.51000	3.72349
6	0.52000	3.71840
7	0.53000	3.75513
8	0.54000	3.67039
9	0.55000	5.97046
10	0.56000	24.51620
11	0.57000	46.06870
12	0.58000	58.76800
13	0.59000	65.98460
14	0.60000	70.01720
15	0.61000	72.00900
16	0.62000	72.67090
17	0.63000	72.05380
18	0.64000	70.12890
19	0.65000	67.04187
20	0.66000	62.86770
21	0.67000	57.76535
22	0.68000	51.90950
23	0.69000	45.56221
24	0.70000	38.86950
25	0.71000	32.06460
26	0.72000	25.28710
27	0.73000	18.63020
28	0.74000	12.08030
29	0.75000	5.73980

No.	Time (s)	Velocity (mm/s)
30	0.76000	-0.59904
31	0.77000	-6.79091
32	0.78000	-13.10753
33	0.79000	-19.55720
34	0.80000	-26.14370
35	0.81000	-32.84790
36	0.82000	-39.59770
37	0.83000	-46.30100
38	0.84000	-52.86920
39	0.85000	-59.10310
40	0.86000	-64.88290
41	0.87000	-70.05850
42	0.88000	-74.43680
43	0.89000	-77.88590
44	0.90000	-80.04620
45	0.91000	-80.71080
46	0.92000	-78.72490
47	0.93000	-71.21790
48	0.94000	-53.46070
49	0.95000	-23.18350
50	0.96000	-5.12536
51	0.97000	-3.74884
52	0.98000	-3.79870
53	0.99000	-3.75822
54	1.00000	-3.78164

Table 22: Valve Acceleration Data Obtained from INVENTOR

No.	Time (s)	Acceleration (mm/s^2)
1	0.47000	0.00000
2	0.48000	-1.33757
3	0.49000	0.71257
4	0.50000	1.04225
5	0.51000	-1.29242
6	0.52000	0.59395
7	0.53000	4.57658
8	0.54000	-12.45210
9	0.55000	735.24500
10	0.56000	2563.37000

No.	Time (s)	Acceleration (mm/s^2)
11	0.57000	1656.07000
12	0.58000	945.30249
13	0.59000	538.69100
14	0.60000	285.42800
15	0.61000	126.58500
16	0.62000	3.79268
17	0.63000	-127.38400
18	0.64000	-252.84400
19	0.65000	-365.62900
20	0.66000	-466.53400
21	0.67000	-550.76200
22	0.68000	-614.02600
23	0.69000	-656.15700
24	0.70000	-677.70100
25	0.71000	-681.08800
26	0.72000	-673.53400
27	0.73000	-659.51200
28	0.74000	-643.99300
29	0.75000	-627.41700
30	0.76000	-612.30700
31	0.77000	-625.99100
32	0.78000	-638.82200
33	0.79000	-651.62800
34	0.80000	-664.63400
35	0.81000	-673.62800
36	0.82000	-675.61300
37	0.83000	-665.54400
38	0.84000	-641.49100
39	0.85000	-603.49100
40	0.86000	-549.78600
41	0.87000	-481.13200
42	0.88000	-396.51000
43	0.89000	-284.20500
44	0.90000	-146.18000
45	0.91000	29.47840
46	0.92000	415.36700
47	0.93000	1166.47000
48	0.94000	2450.58000
49	0.95000	3188.56000
50	0.96000	520.76900

No.	Time (s)	Acceleration (mm/s^2)
51	0.97000	-23.45940
52	0.98000	6.57884
53	0.99000	-0.76339
54	1.00000	-0.60794

MATLAB Script M-file for The Kinematic Solution Calculations of Chapter 4

```
% Chapter 4 Calculations by MATLAB of Kinematic solution of the proposed
% mechanism using LCE's, where the dimensions are initially assumed
% (parameters).
clc; close all; clear
syms theta2 theta3

% parameters
b1=17;d1=48;rc=5;rf=5;r6=85;a1=25;a2=50;a3=46;a4=40;a5=40;a6=35;
a7=57.5;a8=35;a9=51.4;

% theta1 is input and found from the initial geometry
theta1=193.83*pi/180;

% LCE1 solution to find theta2 and theta3 and considering the initial value
% of theta3= 185.23 deg from geometry.
[theta2,theta3]=solve(b1+a3*cos(theta3)+a2*cos(theta2)-a1*cos(theta1)==0,...
   d1+a3*sin(theta3)+a2*sin(theta2)-a1*sin(theta1)==0, [theta2,theta3]);
theta2=vpa(theta2(2)*180/pi)+360
theta3=vpa(theta3(2)*180/pi)+360

% LCE2 solution to find psi, theta5  and theta6 using using various gama to
% vary the maximum valve lift.
syms theta5 psi
alfa=90.5*pi/180;
theta3in=185.22*pi/180;
theta3=theta3*pi/180;
dtheta=theta3in-theta3;
gama=45*pi/180;
theta6=(theta3-alfa-psi);
Sf=2*r6*sin(psi/2);
Sfx=Sf*cos(psi/2)*sin(pi-(theta3-alfa))+Sf*sin(psi/2)*cos((theta3-alfa)+pi);
Sfy=Sf*cos(psi/2)*cos(pi-(theta3-alfa))+Sf*sin(psi/2)*sin((theta3-alfa)+pi);

[theta5,psi]=solve([(rc+rf)*cos(theta3-alfa)+Sfx+a4*sin(gama)-a5*...
   cos(theta5)-(r6-a6)*cos(theta6)==0,(rc+rf)*sin(theta3-alfa)+Sfy+a4*...
   cos(gama)-a5*sin(theta5)-(r6-a6)*sin(theta6)==0],[theta5,psi]);
psi=vpa(psi-dtheta);

% finding theta6 by back substitution in LCE2.
Sf=2*r6*sin(psi/2);
Sfx=Sf*cos(psi/2)*sin(pi-(theta3-alfa))+Sf*sin(psi/2)*cos((theta3-alfa)+pi);
Sfy=Sf*cos(psi/2)*cos(pi-(theta3-alfa))+Sf*sin(psi/2)*sin((theta3-alfa)+pi);
```

```
theta6=atan2((rc+rf)*sin(theta3-alfa)+Sfy+a4*cos(gama)-a5*sin(theta5),...
  (rc+rf)*cos(theta3-alfa)+Sfx+a4*sin(gama)-a5*cos(theta5));

 % LCE3 solution to find theta7
theta7=atan2(d1-a4*cos(gama)+a5*sin(theta5)-a6*sin(theta6),...
   b1-a4*sin(gama)+a5*cos(theta5)-a6*cos(theta6));

beta=acos(a9/a7);
theta7in=310.902345*pi/180;
theta5=vpa(theta5*180/pi)+360
theta6=vpa(theta6*180/pi)
psi=vpa(psi*180/pi)

% LCE4 solution to find valve lift sV(mm) and rocker arm displacement
% sH(mm) and using the initial value of theta7.
sV=vpa(cos((theta7in-theta7)/2+beta)*2*a7*sin((theta7in-theta7)/2))
sH=vpa(sin((theta7in-theta7)/2+beta)*2*a7*sin((theta7in-theta7)/2))

theta7=vpa(theta7*180/pi)+360
```

Main MATLAB Script M-file for The Kinematic Solution

```
% Calculation of the objective functions and the cam profile based on
% the work of Prof. Gordon Blair on race engines.
clc; close all; clear
Rp=20;

A1=[1 1/4 1/4^2 1/4^3 1/4^4 1/4^5;...
   0 1 2/4 3/4^2 4/4^3 5/4^4;...
   0 0 2 6/4 12/4^2 20/4^3;...
   1 1/2 1/2^2 1/2^3 1/2^4 1/2^5;...
   0 1 2/2 3/2^2 4/2^3 5/2^4;...
   0 0 2 6/2 12/2^2 20/2^3];
B1=[0.3 1.2 0 3.7 23.27 0]';
C1=inv(A1)*B1;
A2=[1 1/2 1/2^2 1/2^3 1/2^4 1/2^5;...
   0 1 2/2 3/2^2 4/2^3 5/2^4;...
   0 0 2 6/2 12/2^2 20/2^3;...
   1 1 1 1 1 1;...
   0 1 2 3 4 5;...
   0 0 2 6 12 20];
B2=[3.7 23.27 0 10 0 -45]';
C2=inv(A2)*B2;
theta=0;beta=100;
j=0;
for i=1:361
   if i==1 | i==199
      v(i)=0;
      a(i)=0;
   end
   if i>1 & i<=100
      if theta <=25
         s(i)=1.2*(theta/beta);
         v(i)=1.2/beta;
         a(i)=0;
```

```
        else if theta <=50
              s(i)=C1(1)+C1(2)*theta/beta+C1(3)*(theta/beta)^2+C1(4)*...
                 (theta/beta)^3+C1(5)*(theta/beta)^4+C1(6)*(theta/beta)^5;
              v(i)=1/beta*(C1(2)+2*C1(3)*(theta/beta)+3*C1(4)*(theta/...
                 beta)^2+4*C1(5)*(theta/beta)^3+5*C1(6)*(theta/beta)^4);
              a(i)=1/(beta^2)*(2*C1(3)+6*C1(4)*(theta/beta)+12*C1(5)...
                 *(theta/beta)^2+20*C1(6)*(theta/beta)^3);
           else
              s(i)=C2(1)+C2(2)*theta/beta+C2(3)*(theta/beta)^2+C2(4)*...
                 (theta/beta)^3+C2(5)*(theta/beta)^4+C2(6)*(theta/beta)^5;
              v(i)=1/beta*(C2(2)+2*C2(3)*(theta/beta)+3*C2(4)*(theta/...
                 beta)^2+4*C2(5)*(theta/beta)^3+5*C2(6)*(theta/beta)^4);
              a(i)=1/(beta^2)*(2*C2(3)+6*C2(4)*(theta/beta)+12*C2(5)...
                 *(theta/beta)^2+20*C2(6)*(theta/beta)^3);
           end
        end
     end
     if  i>100 & i<199
        if theta <=25
           s(i)=1.2*(theta/beta);
           v(i)=-1.2/beta;
           a(i)=0;
        else if theta <=50
              s(i)=C1(1)+C1(2)*theta/beta+C1(3)*(theta/beta)^2+C1(4)*...
                 (theta/beta)^3+C1(5)*(theta/beta)^4+C1(6)*(theta/beta)^5;
              v(i)=-1/beta*(C1(2)+2*C1(3)*(theta/beta)+3*C1(4)*(theta/...
                 beta)^2+4*C1(5)*(theta/beta)^3+5*C1(6)*(theta/beta)^4);
              a(i)=1/(beta^2)*(2*C1(3)+6*C1(4)*(theta/beta)+12*C1(5)...
                 *(theta/beta)^2+20*C1(6)*(theta/beta)^3);
           else
              s(i)=C2(1)+C2(2)*theta/beta+C2(3)*(theta/beta)^2+C2(4)*...
                 (theta/beta)^3+C2(5)*(theta/beta)^4+C2(6)*(theta/beta)^5;
              v(i)=-1/beta*(C2(2)+2*C2(3)*(theta/beta)+3*C2(4)*(theta/...
                 beta)^2+4*C2(5)*(theta/beta)^3+5*C2(6)*(theta/beta)^4);
              a(i)=1/(beta^2)*(2*C2(3)+6*C2(4)*(theta/beta)+12*C2(5)...
                 *(theta/beta)^2+20*C2(6)*(theta/beta)^3);
           end
        end
     end
     phi(i)=j;
     j=j+1;
     if i <=100
        theta=theta+1;
     else
        theta=theta-1;
     end
     if i>=200
        s(i)=0;
        v(i)=0;
        a(i)=0;
     end
  end
  for i=1:361
     rho(i)=((((Rp+s(i))^2+v(i)^2))^(3/2))/((Rp+s(i))^2+2*v(i)^2-a(i)*(Rp+s(i)));
  end
  j=1;
```

```
for i=1:5:361
   phin(j)=phi(i);
   sn(j)=s(i);
   j=j+1;
end
T=0.24;
T1=0.24;
for j=1:200
   S(j)=s(j);
   t(j)=T;
   T=T+0.00141959799;
end
% valve lift objective function
figure
plot(phi,s)
% valve speed objective function
figure
plot(phi,v)
% valve acceleration objective function
figure
plot(phi,a)
% cam Profile
figure
polar(phi*pi/180,rho)
% valve lift objective function varied in time (s)
figure
plot(t,S)

% Kinematic solution of the proposed mechanism using LCE's, where the
% dimensions are initially assumed (parameters).
syms theta2 theta3
 % parameters
b1=17;d1=48;rc=5;rf=5;r6=85;a1=25;a2=50;a3=45;a4=40;a5=40;a6=35;
a7=57.5;a8=35;a9=51.4;
% theta1 is input and found from the cam rotation using INVENTOR.
theta1=[193.83 193.82000 193.77926 193.73860 193.69803 193.65740 193.61670...
   193.57600 193.53540 193.49490 193.45440 193.41390 193.37300 193.33217...
   193.29210 193.24980 193.18880 193.06470 192.81870 192.44000 191.95440...
   191.39070 190.77130 190.11340 189.42980 188.73010 188.02190 187.31093...
   186.60170 185.89740 185.20100 184.51560 183.84390 183.18870 182.55240...
   181.93700 181.34450 180.77650 180.23440 179.71950 179.23280 178.77500...
   178.34670 177.94830 177.57970 177.24110 176.93210 176.65250 176.40185...
   176.17950 175.98504 175.81780 175.67720 175.56280 175.47400 175.41040...
   175.37150 175.35670 175.37720 175.41500 175.47240 175.55250 175.65770...
   175.78940 175.94860 176.13530 176.34930 176.59030 176.85880 177.15520...
   177.48040 177.83470 178.21840 178.63150 179.07420 179.54640 180.04790...
   180.57850 181.13790 181.72530 182.34010 182.98140 183.64810 184.33877...
   185.05160 185.78430 186.53430 187.29880 188.07420 188.85470 189.63140...
   190.39130 191.11610 191.77980 192.34880 192.78422 193.05690 193.18800...
   193.25020 193.29300 193.33350 193.37480 193.41610 193.45700 193.49780...
   193.53880 193.57980 193.62090 193.66190 193.70280 193.74380 193.78470...
   193.82580 193.83000 ]*pi/180;

% LCE1 solution to find theta2 and theta3 and asuming the initial value of
% theta3= 185.23 deg.
for i=1:114
```

```
    [sol1,sol2]=solve(b1+a3*cos(theta3)+a2*cos(theta2)-a1*cos(theta1(i))==0,...
       d1+a3*sin(theta3)+a2*sin(theta2)-a1*sin(theta1(i))==0, [theta2,theta3]);
    Theta2(i)=sol1(2);
    Theta3=vpa(sol2(2)*180/pi);

    % LCE2 solution to find psi, theta5  and theta6 using using gama=28.55
    % deg., as it corresponds to the maximum valve lift of 10mm.
    syms theta5 psi
    alfa=90*pi/180;
    theta3in=185.23*pi/180;
    Theta3(i)=Theta3*pi/180;
    dtheta(i)=theta3in-Theta3(i);
    gama=28.55*pi/180;
    Theta6(i)=(Theta3(i)-alfa-psi);
    Sf=2*r6*sin(psi/2);
    Sfx(i)=Sf*cos(psi/2)*sin(pi-(Theta3(i)-alfa))+Sf*sin(psi/2)*...
       cos((Theta3(i)-alfa)+pi);
    Sfy(i)=Sf*cos(psi/2)*cos(pi-(Theta3(i)-alfa))+Sf*sin(psi/2)*...
       sin((Theta3(i)-alfa)+pi);
    [sol1,sol2]=solve([(rc+rf)*cos(Theta3(i)-alfa)+Sfx(i)+a4*sin(gama)-a5*...
       cos(theta5)-(r6-a6)*cos(Theta6(i))==0,(rc+rf)*sin(Theta3(i)-alfa)+...
       Sfy(i)+a4*cos(gama)-a5*sin(theta5)-(r6-a6)*sin(Theta6(i))==0],...
       [theta5,psi]);
    Psi(i)=sol2-dtheta(i);
    Theta5(i)= sol1;

    % finding theta6 by back substitution in LCE2.
    SF(i)=2*r6*sin(Psi(i)/2);
    SFx(i)=SF(i)*cos(Psi(i)/2)*sin(pi-(Theta3(i)-alfa))+SF(i)*sin(Psi(i)/2)*...
       cos((Theta3(i)-alfa)+pi);
    SFy(i)=SF(i)*cos(Psi(i)/2)*cos(pi-(Theta3(i)-alfa))+SF(i)*sin(Psi(i)/2)*...
       sin((Theta3(i)-alfa)+pi);
    Theta6(i)=atan2((rc+rf)*sin(Theta3(i)-alfa)+SFy(i)+a4*cos(gama)-a5*...
       sin(Theta5(i)),(rc+rf)*cos(Theta3(i)-alfa)+SFx(i)+a4*sin(gama)-a5*...
       cos(Theta5(i)));

    % LCE3 solution to find theta7
    Theta7(i)=atan2(d1-a4*cos(gama)+a5*sin(Theta5(i))-a6*sin(Theta6(i)),...
       b1-a4*sin(gama)+a5*cos(Theta5(i))-a6*cos(Theta6(i)));

    % LCE4 solution to find valve lift Sv(mm) and rocker arm          % displacement SH (mm). Using the
initial value of theta7.

    %    beta=acos(a9/a7);
    theta7in=311.788859*pi/180;
    sV(i)=vpa(cos((theta7in-Theta7(i))/2+beta)*2*a7*sin((theta7in-Theta7(i))/2));
    sH(i)=vpa(sin((theta7in-Theta7(i))/2+beta)*2*a7*sin((theta7in-Theta7(i))/2));
end
theta2=vpa(Theta2*180/pi);
theta3=vpa(Theta3*180/pi);
theta5=vpa(Theta5*180/pi)+360;
theta6=vpa(Theta6*180/pi);
psi=vpa(Psi*180/pi);
theta7=vpa(Theta7*180/pi)+360;
for i=1:114
    t1(i)=T1;
```

```
   T1=T1+0.0025;
end

% Plotting a comparison graph between the valve lift objective function (S)
% and the calculated valve lift of the proposed mechainsm (SV)
figure
plot(t,S,t1,sV)
```

FIREFLY MATLAB Script M-file for The Optimization of Links

```
%  Firefly Algorithm for Optimization of Mechanism's Links Lengths
clc;
clear;
close all;
% Calculation of  the objective functions and the cam profile based on
% the work of Prof. Gordon Blair on race engines.
Rp=20;

A1=[1 1/4 1/4^2 1/4^3 1/4^4 1/4^5;...
   0 1 2/4 3/4^2 4/4^3 5/4^4;...
   0 0 2 6/4 12/4^2 20/4^3;...
   1 1/2 1/2^2 1/2^3 1/2^4 1/2^5;...
   0 1 2/2 3/2^2 4/2^3 5/2^4;...
   0 0 2 6/2 12/2^2 20/2^3];
B1=[0.3 1.2 0 3.7 23.27 0]';
C1=inv(A1)*B1;
A2=[1 1/2 1/2^2 1/2^3 1/2^4 1/2^5;...
   0 1 2/2 3/2^2 4/2^3 5/2^4;...
   0 0 2 6/2 12/2^2 20/2^3;...
   1 1 1 1 1 1;...
   0 1 2 3 4 5;...
   0 0 2 6 12 20];
B2=[3.7 23.27 0 10 0 -45]';
C2=inv(A2)*B2;
theta=0;beta=100;
j=0;
for i=1:361
   if i==1 | i==199
      v(i)=0;
      a(i)=0;
   end
   if i>1 & i<=100
      if theta <=25
         s(i)=1.2*(theta/beta);
         v(i)=1.2/beta;
         a(i)=0;
      else if theta <=50
            s(i)=C1(1)+C1(2)*theta/beta+C1(3)*(theta/beta)^2+C1(4)*...
               (theta/beta)^3+C1(5)*(theta/beta)^4+C1(6)*(theta/beta)^5;
            v(i)=1/beta*(C1(2)+2*C1(3)*(theta/beta)+3*C1(4)*(theta/...
               beta)^2+4*C1(5)*(theta/beta)^3+5*C1(6)*(theta/beta)^4);
            a(i)=1/(beta^2)*(2*C1(3)+6*C1(4)*(theta/beta)+12*C1(5)...
               *(theta/beta)^2+20*C1(6)*(theta/beta)^3);
         else
```

```
            s(i)=C2(1)+C2(2)*theta/beta+C2(3)*(theta/beta)^2+C2(4)*...
               (theta/beta)^3+C2(5)*(theta/beta)^4+C2(6)*(theta/beta)^5;
            v(i)=1/beta*(C2(2)+2*C2(3)*(theta/beta)+3*C2(4)*(theta/...
               beta)^2+4*C2(5)*(theta/beta)^3+5*C2(6)*(theta/beta)^4);
            a(i)=1/(beta^2)*(2*C2(3)+6*C2(4)*(theta/beta)+12*C2(5)...
               *(theta/beta)^2+20*C2(6)*(theta/beta)^3);
         end
      end
   end
   if  i>100 & i<199
      if theta <=25
         s(i)=1.2*(theta/beta);
         v(i)=-1.2/beta;
         a(i)=0;
      else if theta <=50
            s(i)=C1(1)+C1(2)*theta/beta+C1(3)*(theta/beta)^2+C1(4)*...
               (theta/beta)^3+C1(5)*(theta/beta)^4+C1(6)*(theta/beta)^5;
            v(i)=-1/beta*(C1(2)+2*C1(3)*(theta/beta)+3*C1(4)*(theta/...
               beta)^2+4*C1(5)*(theta/beta)^3+5*C1(6)*(theta/beta)^4);
            a(i)=1/(beta^2)*(2*C1(3)+6*C1(4)*(theta/beta)+12*C1(5)...
               *(theta/beta)^2+20*C1(6)*(theta/beta)^3);
         else
            s(i)=C2(1)+C2(2)*theta/beta+C2(3)*(theta/beta)^2+C2(4)*...
               (theta/beta)^3+C2(5)*(theta/beta)^4+C2(6)*(theta/beta)^5;
            v(i)=-1/beta*(C2(2)+2*C2(3)*(theta/beta)+3*C2(4)*(theta/...
               beta)^2+4*C2(5)*(theta/beta)^3+5*C2(6)*(theta/beta)^4);
            a(i)=1/(beta^2)*(2*C2(3)+6*C2(4)*(theta/beta)+12*C2(5)...
               *(theta/beta)^2+20*C2(6)*(theta/beta)^3);
         end
      end
   end
   phi(i)=j;
   j=j+1;
   if i <=100
      theta=theta+1;
   else
      theta=theta-1;
   end
   if i>=200
      s(i)=0;
      v(i)=0;
      a(i)=0;
   end
end
for i=1:361
   rho(i)=((((Rp+s(i))^2+v(i)^2))^(3/2))/((Rp+s(i))^2+2*v(i)^2-a(i)*(Rp+s(i)));
end
j=1;
for i=1:5:361
   phin(j)=phi(i);
   sn(j)=s(i);
   j=j+1;
end
T=0.24;
T1=0.24;
th=193.83;
```

```
for j=1:200
   S(j)=s(j);
   t(j)=T;
   T=T+0.00141959799;
   PHI(j)=phi(j);
   if j<=100
      TH1(j)=th;
      th=th-.1864;
   else
      th=th+.1864;
      TH1(j)=th;
   end
end

nVar=7;              % Number of parameters

VarSize=[1 nVar];      % Parameters Matrix Size

VarMin=20;            % parameters Lower Bound
VarMax=85;            % Parameters Upper Bound
VarMin1=20;           % a1 Lower Bound
VarMax1=25;           % a1 Upper Bound
VarMin2=45;           % a2 Lower Bound
VarMax2=55;           % a2 Bound
VarMin3=35;           % a3,a4,a5 & a6 Lower Bound
VarMax3=45;           % a3,a4,a5 & a6 Upper Bound
VarMin4=80;           % r6 Lower Bound
VarMax4=85;           % r6 Upper Bound
% Firefly Algorithm Parameters

MaxIt=30;       % Maximum Number of Iterations
nPop=5;          % Number of Fireflies
gamma=1;         % attractiveness Variation Coefficient
beta0=0.4;       % Attraction Coefficient
alpha=0.25;      % Randomization Parameter
alpha_damp=0.98;  % Randomization Parameter Damping Ratio
m=2;
dmax = (VarMax-VarMin)*sqrt(nVar);


% Initialization

% Empty Firefly Structure
firefly.linkleng=[];
firefly.error=[];

% Initialize Population Array
pop=repmat(firefly,nPop,1);

% Initialize Best Solution Ever Found
BestSol.error=inf;

% Create Initial Fireflies
for i=1:nPop
   tic
 pop(i).linkleng=unifrnd(VarMin,VarMax,VarSize);
```

```
  pop(i).linkleng(1)=unifrnd(VarMin1,VarMax1,1);
  pop(i).linkleng(2)=unifrnd(VarMin2,VarMax2,1);
  pop(i).linkleng(3)=unifrnd(VarMin3,VarMax3,1);
  pop(i).linkleng(4)=unifrnd(VarMin3,VarMax3,1);
  pop(i).linkleng(5)=unifrnd(VarMin3,VarMax3,1);
  pop(i).linkleng(6)=unifrnd(VarMin3,VarMax3,1);
  pop(i).linkleng(7)=unifrnd(VarMin4,VarMax4,1);
  pop(i).linkleng
  pop(i).error=mainopt(pop(i).linkleng,S,t);

  if pop(i).error<=BestSol.error
     BestSol=pop(i);
  end
  toc
end

% Array to Hold Best Values and Iteration Time
BestError=zeros(MaxIt,1);
BestLeng=zeros(MaxIt,nVar);
Time=zeros(MaxIt,1);

% Firefly Algorithm Loop
for it=1:MaxIt
  tic
   it
   if BestSol.error >= 10e-1
   newpop=repmat(firefly,nPop,1);
   for i=1:nPop
      newpop(i).error = inf;
      for j=1:nPop
         if pop(i).error < pop(j).error
            rij=norm(pop(j).linkleng-pop(i).linkleng)/dmax;
            beta=beta0*exp(-gamma*rij^m);
            e=unifrnd(0,+1,VarSize);
            newsol.linkleng = pop(i).linkleng ...
                     + beta*rand(VarSize).*(pop(j).linkleng-pop(i).linkleng) ...
                     + alpha*(e-0.5);

            newsol.linkleng=max(newsol.linkleng,VarMin);
            newsol.linkleng=min(newsol.linkleng,VarMax);

            newsol.error=mainopt(newsol.linkleng,S,t);

            if newsol.error <= newpop(i).error
               newpop(i) = newsol;
               if newpop(i).error<=BestSol.error
                  BestSol=newpop(i);
               end
            end
         end
      end
   end
   end

%     Fireflies New Population
   pop=[pop
```

```
        newpop];

%     Rank The New Fireflies Population
    [~, SortOrder]=sort([pop.error]);
    pop=pop(SortOrder);

%     Reduce The Population to nPop
    pop=pop(1:nPop);

%     Store Best Error and Length Ever Found
    BestError(it)=BestSol.error;
    BestLeng(it,:)=BestSol.linkleng;
%     Show Iteration Information
    disp(['Iteration ' num2str(it) ': Best Error = ' num2str(BestError(it))...
        ': Best Lengths = ' num2str(BestLeng(it,:))])

%     Damping Randomization Parameter to Reduce Randomness
   alpha = alpha*alpha_damp;
   toc
   Time(it)=toc;
%    Save The Results
   save('opt.mat','it','BestError','BestLeng','Time')

end
% Plot The Results
figure;
semilogy(BestError,'LineWidth',2);
xlabel('Iteration');
ylabel('Best Error');
grid on;
```

Total Error Function MATLAB Script M-file

```
function ER=mainopt(a,S,t)

a1=a(1);a2=a(2);a3=a(3);a4=a(4);a5=a(5);a6=a(6);r6=a(7);
% Kinematic solution of the proposed mechanism using LCE's, where the
% dimensions are initially assumed (parameters).
syms theta2 theta3
T1=0.24;
% parameters
b1=16.9999;d1=48;rc=5;rf=5;a7=57.5;a8=35;a9=51.4;
% theta1 is input and found from the cam rotation using INVENTOR.
theta1=[193.83  193.77926  193.69803  193.61670...
193.53540 193.45440 193.37300 ...
193.29210 193.18880 192.81870 191.95440...
190.77130 189.42980 188.02190 ...
186.60170 185.20100 183.84390 182.55240...
181.34450 180.23440 179.23280 ...
178.34670 177.57970 176.93210 176.40185...
175.98504 175.67720 175.47400 ...
175.37150 175.37720 175.47240 175.65770...
175.94860 176.34930 176.85880 ...
177.48040 178.21840 179.07420 180.04790...
181.13790 182.34010 183.64810 ...
```

```
185.05160 186.53430 188.07420 189.63140...
191.11610 192.34880 193.05690 ...
193.25020 193.33350 193.41610 193.49780...
193.57980 193.66190 193.74380 ...
193.82580 ]*pi/180;

% LCE1 solution to find theta2 and theta3 and asuming the initial value of
% theta3= 185.23 deg.
for i=1:57
[sol1,sol2]=solve(b1+a3*cos(theta3)+a2*cos(theta2)-a1*cos(theta1(i))==0,...
d1+a3*sin(theta3)+a2*sin(theta2)-a1*sin(theta1(i))==0, [theta2,theta3]);
Theta2(i)=sol1(2);
Theta3=vpa(sol2(2)*180/pi);

% LCE2 solution to find PSI, theta5  and theta6 using using gama=28.55
% deg., as it corresponds to the maximum valve lift of 10mm.
syms theta5 PSI
alfa=90*pi/180;
theta3in=185.23*pi/180;
Theta3(i)=Theta3*pi/180;
dtheta(i)=theta3in-Theta3(i);
gama=28.55*pi/180;
Theta6(i)=(Theta3(i)-alfa-PSI);
Sf=2*r6*sin(PSI/2);
Sfx(i)=Sf*cos(PSI/2)*sin(pi-(Theta3(i)-alfa))+Sf*sin(PSI/2)*...
cos((Theta3(i)-alfa)+pi);
Sfy(i)=Sf*cos(PSI/2)*cos(pi-(Theta3(i)-alfa))+Sf*sin(PSI/2)*...
sin((Theta3(i)-alfa)+pi);
[sol1,sol2]=solve([(rc+rf)*cos(Theta3(i)-alfa)+Sfx(i)+a4*sin(gama)-a5*...
cos(theta5)-(r6-a6)*cos(Theta6(i))==0,(rc+rf)*sin(Theta3(i)-alfa)+...
Sfy(i)+a4*cos(gama)-a5*sin(theta5)-(r6-a6)*sin(Theta6(i))==0],...
[theta5,PSI]);
Psi(i)=sol2-dtheta(i);
Theta5(i)= sol1;

% finding theta6 by back substitution in LCE2.
SF(i)=2*r6*sin(Psi(i)/2);
SFx(i)=SF(i)*cos(Psi(i)/2)*sin(pi-(Theta3(i)-alfa))+SF(i)*sin(Psi(i)/2)*...
cos((Theta3(i)-alfa)+pi);
SFy(i)=SF(i)*cos(Psi(i)/2)*cos(pi-(Theta3(i)-alfa))+SF(i)*sin(Psi(i)/2)*...
sin((Theta3(i)-alfa)+pi);
Theta6(i)=atan2((rc+rf)*sin(Theta3(i)-alfa)+SFy(i)+a4*cos(gama)-a5*...
sin(Theta5(i)),(rc+rf)*cos(Theta3(i)-alfa)+SFx(i)+a4*sin(gama)-a5*...
cos(Theta5(i)));

% LCE3 solution to find theta7
Theta7(i)=atan2(d1-a4*cos(gama)+a5*sin(Theta5(i))-a6*sin(Theta6(i)),...
b1-a4*sin(gama)+a5*cos(Theta5(i))-a6*cos(Theta6(i)));

% LCE4 solution to find valve lift Sv(mm) and rocker arm displacement
% Sh (mm). Using the initial value of theta7.
beta=acos(a9/a7);
theta7in=311.76936224*pi/180;
Sv(i)=vpa(cos((theta7in-Theta7(i))/2+beta)*2*a7*sin((theta7in-Theta7(i))/2));
Sh(i)=vpa(sin((theta7in-Theta7(i))/2+beta)*2*a7*sin((theta7in-Theta7(i))/2));
end
```

```
% Error Calculation
for i=1:57
t1(i)=T1;
T1=T1+2*0.0025;
end
for i=1:57
for j=1:199
if t1(i)>t(j)& t1(i)<t(j+1)
Sn(i)=S(j+1)-(S(j+1)-S(j))*(t(j+1)-t1(i))/(t(j+1)-t(j));
error(i)=abs(Sv(i)-Sn(i));
end
end
end
% Total Error Calculation
ER=sum(error)
end
```

MATLAB Script M-file to Compare the Valve Velocity and Acceleration

```
% The M-file to draw the velocity and acceleration obtained from inventor,
% objective, and MATLAB calculations
clc; close all; clear
ti=[0.47000 0.48000 0.49000 0.50000 0.51000 0.52000 0.53000 0.54000 0.55000...
0.56000 0.57000 0.58000 0.59000 0.60000 0.61000 0.62000 0.63000 0.64000 0.65000...
0.66000 0.67000 0.68000 0.69000 0.70000 0.71000 0.72000 0.73000 0.74000 0.75000...
0.76000 0.77000 0.78000 0.79000 0.80000 0.81000 0.82000 0.83000 0.84000 0.85000...
0.86000 0.87000 0.88000 0.89000 0.90000 0.91000 0.92000 0.93000 0.94000 0.95000...
0.96000 0.97000 0.98000 0.99000 1.00000];
vi=[0.00000 3.72202 3.71816 3.72547 3.72349 3.71840 3.75513 3.67039 5.97046...
24.51620 46.06870 58.76800 65.98460 70.01720 72.00900 72.67090 72.05380...
70.12890 67.04187 62.86770 57.76535 51.90950 45.56221 38.86950 32.06460...
25.28710 18.63020 12.08030 5.73980 -0.59904 -6.79091 -13.10753 -19.55720...
-26.14370 -32.84790 -39.59770 -46.30100 -52.86920 -59.10310 -64.88290...
-70.05850 -74.43680 -77.88590 -80.04620 -80.71080 -78.72490 -71.21790...
-53.46070 -23.18350 -5.12536 -3.74884 -3.79870 -3.75822 -3.78164]*.2327/72.6709;
ai=[0.00000 -1.33757 0.71257 1.04225 -1.29242 0.59395 4.57658 -12.45210...
735.24500 2563.37000 1656.07000 945.30249 538.69100 285.42800 126.58500...
3.79268 -127.38400 -252.84400 -365.62900 -466.53400 -550.76200 -614.02600...
-656.15700 -677.70100 -681.08800 -673.53400 -659.51200 -643.99300 -627.41700...
-612.30700 -625.99100 -638.82200 -651.62800 -664.63400 -673.62800 -675.61300...
-665.54400 -641.49100 -603.49100 -549.78600 -481.13200 -396.51000 -284.20500...
-146.18000 29.47840 415.36700 1166.47000 2450.58000 3188.56000 520.76900...
-23.45940 6.57884 -0.76339 -0.60794]*0.01409/2563;

 Rp=20;

A1=[1 1/4 1/4^2 1/4^3 1/4^4 1/4^5;...
   0 1 2/4 3/4^2 4/4^3 5/4^4;...
   0 0 2 6/4 12/4^2 20/4^3;...
   1 1/2 1/2^2 1/2^3 1/2^4 1/2^5;...
   0 1 2/2 3/2^2 4/2^3 5/2^4;...
   0 0 2 6/2 12/2^2 20/2^3];
B1=[0.3 1.2 0 3.7 23.27 0]';
C1=inv(A1)*B1
A2=[1 1/2 1/2^2 1/2^3 1/2^4 1/2^5;...
   0 1 2/2 3/2^2 4/2^3 5/2^4;...
   0 0 2 6/2 12/2^2 20/2^3;...
```

```
    1 1 1 1 1 1;...
    0 1 2 3 4 5;...
    0 0 2 6 12 20];
B2=[3.7 23.27 0 10 0 -45]';
C2=inv(A2)*B2
theta=0;beta=100;
j=0;
for i=1:361
   if i==1 | i==199
      v(i)=0;
      a(i)=0;
   end

       if i>1 & i<=100
          if theta <=25
             s(i)=1.2*(theta/beta);
            v(i)=1.2/beta;
            a(i)=0;
          else if theta <=50
             s(i)=C1(1)+C1(2)*theta/beta+C1(3)*(theta/beta)^2+C1(4)*(theta/beta)^3+...
              C1(5)*(theta/beta)^4+C1(6)*(theta/beta)^5;
             v(i)=1/beta*(C1(2)+2*C1(3)*(theta/beta)+3*C1(4)*(theta/beta)^2+4*C1(5)...
                *(theta/beta)^3+5*C1(6)*(theta/beta)^4);
             a(i)=1/(beta^2)*(2*C1(3)+6*C1(4)*(theta/beta)+12*C1(5)...
                *(theta/beta)^2+20*C1(6)*(theta/beta)^3);
             else
                s(i)=C2(1)+C2(2)*theta/beta+C2(3)*(theta/beta)^2+C2(4)*(theta/beta)^3+...
                  C2(5)*(theta/beta)^4+C2(6)*(theta/beta)^5;
                v(i)=1/beta*(C2(2)+2*C2(3)*(theta/beta)+3*C2(4)*(theta/beta)^2+4*C2(5)...
                *(theta/beta)^3+5*C2(6)*(theta/beta)^4);
               a(i)=1/(beta^2)*(2*C2(3)+6*C2(4)*(theta/beta)+12*C2(5)...
                *(theta/beta)^2+20*C2(6)*(theta/beta)^3);
          end
          end
       end
            if  i>100 & i<199
              if theta <=25
                 s(i)=1.2*(theta/beta);
                 v(i)=-1.2/beta;
                 a(i)=0;
              else if theta <=50
                      s(i)=C1(1)+C1(2)*theta/beta+C1(3)*(theta/beta)^2+C1(4)*(theta/beta)^3+...
                 C1(5)*(theta/beta)^4+C1(6)*(theta/beta)^5;
                    v(i)=-1/beta*(C1(2)+2*C1(3)*(theta/beta)+3*C1(4)*(theta/beta)^2+4*C1(5)...
                       *(theta/beta)^3+5*C1(6)*(theta/beta)^4);
                     a(i)=1/(beta^2)*(2*C1(3)+6*C1(4)*(theta/beta)+12*C1(5)...
                 *(theta/beta)^2+20*C1(6)*(theta/beta)^3);
                    else
                    s(i)=C2(1)+C2(2)*theta/beta+C2(3)*(theta/beta)^2+C2(4)*(theta/beta)^3+...
                   C2(5)*(theta/beta)^4+C2(6)*(theta/beta)^5;
                    v(i)=-1/beta*(C2(2)+2*C2(3)*(theta/beta)+3*C2(4)*(theta/beta)^2+4*C2(5)...
                       *(theta/beta)^3+5*C2(6)*(theta/beta)^4);
                     a(i)=1/(beta^2)*(2*C2(3)+6*C2(4)*(theta/beta)+12*C2(5)...
                 *(theta/beta)^2+20*C2(6)*(theta/beta)^3);
                 end
```

```
        end
    end

    phi(i)=j;
    j=j+1;
    if i <=100
        theta=theta+1;
    else
        theta=theta-1;
    end
    if i>=200
        s(i)=0;
        v(i)=0;
        a(i)=0;
    end

end
for i=1:361
    rho(i)=((((Rp+s(i))^2+v(i)^2))^(3/2))/((Rp+s(i))^2+2*v(i)^2-a(i)*(Rp+s(i)));
end
for i=1:5:361
    phin(j)=phi(i);
    sn(j)=s(i);
    j=j+1;
end


T=0.48;
for j=1:200
    S(j)=s(j);
    V(j)=v(j);
    A(j)=a(j);
    t(j)=T;
    T=T+.00265;
end

syms theta2 theta3

% parameters
b1=16.9999;d1=48;rc=5;rf=5;r6=81.2476;a1=23.6196;a2=53.3117;a3=43.8314
;a4=40.7127;a5=37.9492;a6=36.9320;
a7=57.5;a8=35;a9=51.4;
% theta1 is input and found from the cam rotation using INVENTOR.
theta1=[193.83 193.82000 193.77926 193.73860 193.69803 193.65740 193.61670...
    193.57600 193.53540 193.49490 193.45440 193.41390 193.37300 193.33217...
    193.29210 193.24980 193.18880 193.06470 192.81870 192.44000 191.95440...
    191.39070 190.77130 190.11340 189.42980 188.73010 188.02190 187.31093...
    186.60170 185.89740 185.20100 184.51560 183.84390 183.18870 182.55240...
    181.93700 181.34450 180.77650 180.23440 179.71950 179.23280 178.77500...
    178.34670 177.94830 177.57970 177.24110 176.93210 176.65250 176.40185...
    176.17950 175.98504 175.81780 175.67720 175.56280 175.47400 175.41040...
    175.37150 175.35670 175.37720 175.41500 175.47240 175.55250 175.65770...
    175.78940 175.94860 176.13530 176.34930 176.59030 176.85880 177.15520...
    177.48040 177.83470 178.21840 178.63150 179.07420 179.54640 180.04790...
    180.57850 181.13790 181.72530 182.34010 182.98140 183.64810 184.33877...
    185.05160 185.78430 186.53430 187.29880 188.07420 188.85470 189.63140...
```

```
    190.39130 191.11610 191.77980 192.34880 192.78422 193.05690 193.18800...
    193.25020 193.29300 193.33350 193.37480 193.41610 193.45700 193.49780...
    193.53880 193.57980 193.62090 193.66190 193.70280 193.74380 193.78470...
    193.82580 193.83000 ]*pi/180;

% LCE1 solution to find theta2 and theta3 and asuming the initial value of
% theta3= 185.23 deg.
for i=1:114
    tic
    [sol1,sol2]=solve(b1+a3*cos(theta3)+a2*cos(theta2)-a1*cos(theta1(i))==0,...
        d1+a3*sin(theta3)+a2*sin(theta2)-a1*sin(theta1(i))==0, [theta2,theta3]);
    Theta2(i)=sol1(2);
    Theta3=vpa(sol2(2)*180/pi);

    % LCE2 solution to find psi, theta5  and theta6 using using gama=28.55
    % deg., as it corresponds to the maximum valve lift of 10mm.
    syms theta5 psi
    alfa=90*pi/180;
    theta3in=185.23*pi/180;
    Theta3(i)=Theta3*pi/180;
    dtheta(i)=theta3in-Theta3(i);
    gama=28.5*pi/180;
    Theta6(i)=(Theta3(i)-alfa-psi);
    Sf=2*r6*sin(psi/2);
    Sfx(i)=Sf*cos(psi/2)*sin(pi-(Theta3(i)-alfa))+Sf*sin(psi/2)*...
        cos((Theta3(i)-alfa)+pi);
    Sfy(i)=Sf*cos(psi/2)*cos(pi-(Theta3(i)-alfa))+Sf*sin(psi/2)*...
        sin((Theta3(i)-alfa)+pi);
    [sol1,sol2]=solve([(rc+rf)*cos(Theta3(i)-alfa)+Sfx(i)+a4*sin(gama)-a5*...
        cos(theta5)-(r6-a6)*cos(Theta6(i))==0,(rc+rf)*sin(Theta3(i)-alfa)+...
        Sfy(i)+a4*cos(gama)-a5*sin(theta5)-(r6-a6)*sin(Theta6(i))==0],...
        [theta5,psi]);
    Psi(i)=sol2-dtheta(i);
    Theta5(i)= sol1;

    % finding theta6 by back substitution in LCE2.
    SF(i)=2*r6*sin(Psi(i)/2);
    SFx(i)=SF(i)*cos(Psi(i)/2)*sin(pi-(Theta3(i)-alfa))+SF(i)*sin(Psi(i)/2)*...
        cos((Theta3(i)-alfa)+pi);
    SFy(i)=SF(i)*cos(Psi(i)/2)*cos(pi-(Theta3(i)-alfa))+SF(i)*sin(Psi(i)/2)*...
        sin((Theta3(i)-alfa)+pi);
    Theta6(i)=atan2((rc+rf)*sin(Theta3(i)-alfa)+SFy(i)+a4*cos(gama)-a5*...
        sin(Theta5(i)),(rc+rf)*cos(Theta3(i)-alfa)+SFx(i)+a4*sin(gama)-a5*...
        cos(Theta5(i)));

    % LCE3 solution to find theta7
    Theta7(i)=atan2(d1-a4*cos(gama)+a5*sin(Theta5(i))-a6*sin(Theta6(i)),...
        b1-a4*sin(gama)+a5*cos(Theta5(i))-a6*cos(Theta6(i)));

    % LCE4 solution to find valve lift Sv(mm) and rocker arm displacement
    % Sh (mm). Using the initial value of theta7.
    beta=acos(a9/a7);
    theta7in=311.73574805592*pi/180;
    Sv(i)=vpa(cos((theta7in-Theta7(i))/2+beta)*2*a7*sin((theta7in-Theta7(i))/2));
    Sh(i)=vpa(sin((theta7in-Theta7(i))/2+beta)*2*a7*sin((theta7in-Theta7(i))/2));
end
```

```
theta2=vpa(Theta2*180/pi);
theta3=vpa(Theta3*180/pi);
theta5=vpa(Theta5*180/pi)+360;
theta6=vpa(Theta6*180/pi);
psi=vpa(Psi*180/pi);
theta7=vpa(Theta7*180/pi)+360;
ss=0;vv=0;T1=.48;
for i=1:114
   t1(i)=T1;
   T1=T1+0.0046;
   Vv(i)=(Sv(i)-ss)/0.0046;
   Av(i)=(Vv(i)-vv)/0.0046;
   ss=Sv(i);
   vv=Vv(i);
end
Vv=Vv*.2327/max(Vv);
Av=Av*0.01409/2952
for i=1:114
   for j=1:199
      if t1(i)>t(j)& t1(i)<t(j+1)
         Sn(i)=S(j+1)-(S(j+1)-S(j))*(t(j+1)-t1(i))/(t(j+1)-t(j));
         error(i)=abs(Sv(i)-Sn(i));
      end
   end
   toc
end

% Plotting a comparison graph between the valve lift objective function (S)
% and the calculated valve lift of the proposed mechainsm (SV)
figure
plot(t,S,t1,Sv)
figure
plot(t1,Sn,'b',t1,Sv,'r')
% Plotting a comparison graph between the Objective, INVENTOR and MATLAB
% velocities and accelerations
figure
plot(t,V,'b',ti,vi,'r',t1,Vv,'g')
figure
plot(t,A,'b',ti, ai,'r',t1,Av,'g')
```