Czech technical university in Prague
Faculty of mechanical engineering

MASTER THESIS

Parametric Optimization of Valve Lift Curve

2016/2017 **Michal Aschermann**

ČESKÉ
VYSOKÉ
UČENÍ
TECHNICKÉ
V PRAZE

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Aschermann** Jméno: **Michal** Osobní číslo: **393131**

Fakulta/ústav: **Fakulta strojní**

Zadávající katedra/ústav: **Ústav mechaniky, biomechaniky a mechatroniky**

Studijní program: **Strojní inženýrství**

Studijní obor: **Aplikovaná mechanika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Parametrická optimalizace křivky zdvihu ventilu motoru

Název diplomové práce anglicky:

Parametric Optimization of Valve Lift Curve

Pokyny pro vypracování:

1) Seznamte se s návrhem a tvorbou profilu vačky pro zajištění zdvihu ventilu motoru.
2) Seznamte se s návrhovým prostředím Valkin, Valdyn a Isight používané pro modelování a analýzu motorových systémů.
3) Proveďte návrh a optimalizaci křivky zdvihu ventilu s ohledem na stanovené limity.

Seznam doporučené literatury:

[1] Lederer P., Brousil J.: Teorie a optimalizace mechanických systémů I. Praha: ČVUT v Praze, 1989.
[2] Norton L. R.: Cam Design and Manufacturing Handbook. Industrial Press Inc., 2009.ISBN 9780831133672.
[3] Dokumentace k software Valkin a Valdyn [https://www.software.ricardo.com/Products/VALKIN, https://www.software.ricardo.com/Products/VALDYN]

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Jan Zavřel Ph.D., ústav mechaniky, biomechaniky a mechatroniky FS

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **19.04.2017** Termín odevzdání diplomové práce: **18.08.2017**

Platnost zadání diplomové práce: ____________

Podpis vedoucí(ho) práce

Podpis vedoucí(ho) ústavu/katedry

Podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

25 . 4 .2017
Datum převzetí zadání

Podpis studenta

Annotation sheet

Name:	**Michal Aschermann**
Title English:	**Parametric Optimization of Valve Lift Curve**
Title Czech:	**Parametrická optimalizace křivky zdvihu ventilu motoru**
Academic year:	2016/2017
Specialization:	Applied mechanics
Department:	Department of mechanics, biomechanics and mechatronics Department of mechanics and mechatronics
Supervisor:	Ing. Jan Zavřel, Ph.D
Scope of work:	pages: 57 number of figures: 32 number of tables: 15 number of apendices: 13
Key words:	optimization, valvetrain, Isight, Valkin, Valdyn, DOE, valve lift curve
Klíčová slova:	optimalizace, rozvodový mechanizmus, Isight, Valkin, Valdyn, DOE, zdvihová křivka ventilu

Abstract: This master thesis describes the process of parametric optimization of valve lift curve that is used in internal combustion engine. By this process, it was attempted to replace engineer's experience and sense by computational power. The valvetrain mechanisms, optimization techniques and software tools are introduced in the beginning. In further sections, the model for the optimization of valve lift curve is assembled using Isight. After testing, the Latin hypercube method is chosen to drive the DOE and simplex method is implemented to drive the optimization. The course of the process is compared for different inputs. Additionally, the improvement of the valve lift curve produced by the optimization process is shown by a comparison with the manually designed one.

Abstrakt: Tato diplomová práce se zabývá procesem parametrické optimalizace křivky zdvihu ventilu spalovacího motoru. Tímto způsobem by mohla výpočetní síla počítače nahradit cit a zkušenosti inženýra. Nejprve jsou popsány typy rozvodových mechanismů, optimalizační metody a softwarové nástroje. Dále je sestaven optimalizační model v programu Isight. Po otestování byla pro průzkum návrhového prostoru vybrána metoda ”Latin hypercube”a pro řízení optimalizačního procesu byla zvolena simplexová metoda. Průběh optimalizačního procesu je porovnán pro různé vstupy. Nakonec je ukázáno zlepšení pomocí porovnání ručně navrhnuté zdvihové křivky a křivky, která je produktem tohoto optimalizačního procesu.

Declaration

I hereby declare that I worked out the thesis individually and that I listed all the literature and software used.

In Prague on ______________ ________________________

Michal Aschermann

Acknowledgments

Firstly, I would like to express my gratitude to Ing. Jan Zavřel Ph.D. for supervision of my thesis and for the time he devoted to me. Secondly, I wish to express many thanks to Ing. Přemysl Kuchař Ph.D. for many indispensable consultations. I am also grateful to Ricardo Prague for supplying me with working place and software licenses for creation of this thesis.

Poděkování

Tímto bych rád poděkoval vedoucímu práce Ing. Janu Zavřelovi, Ph.D. za rady a čas, který mi věnoval. Dále bych rád poděkoval Ing. Přemyslu Kuchařovi Ph.D. za mnoho nepostradatelných konzultací. Také bych rád poděkoval firmě Ricardo Prague za poskytnutí pracovního místa a potřebných softwarových licencí pro tvorbu této práce.

Nomenclature

y	Valve lift	$[mm]$
h	Maximum valve lift	$[mm]$
v	Valve velocity	$[mm/deg]$
a	Valve acceleration	$[mm/deg^2]$
j	Valve jerk	$[mm/deg^3]$
ϕ_i	Local coordinate of i-th valve lift segment	$[deg]$
θ_i	Length of i-th valve lift segment	$[deg]$
C_{ij}	Coefficients of MPOL valve lift segments	$[-]$
b	Coefficient of valve lift curve	$[-]$
c	Coefficient of valve lift curve	$[-]$
d	Coefficient of valve lift curve	$[-]$
e	Coefficient of valve lift curve	$[-]$
A_f	Flank acceleration	$[mm/deg^2]$
A_n	Nose acceleration	$[mm/deg^2]$
$\mathbf{x}$	Vector of optimization parameters	$[mm/deg^2]$

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis explores the feasibility of using optimization process for design and optimization of valve lift curve and the effect on design time compared to manual process.

This document introduces the fundamentals of cam systems when applied in internal combustion engine including various design approaches. Lift profile definition described by Valkin is utilized. The limits given for the cam-follower-valve systems are listed. Various optimization methods are described and the components of optimization chain are developed and assembled using Isight. The best available method was chosen to drive the optimization component. In the end, the model was driven for several input conditions. The computation times and the results were observed and compared. Additionally, they were compared against manually designed profile. The motivation of this work was to develop an automated process for the valve lift curve design and optimization.

Chapter 2

Objectives of the Thesis

The objective of this thesis is to develop a process for design and optimization of valve lift curve for valvetrain of internal combustion engine. This is done based on given models of valvetrain mechanism in Valkin and Valdyn. Firstly, the fundamentals of valvetrain mechanisms are observed. Then, the study focuses on various optimization methods that are described. Valkin and Valdyn models are edited in order to obtain the required output to work together with Isight. Finally, the Isight optimization model is assembled.
The outcome of this thesis is the optimized valve lift curve for the given model and the Isight model that can be used for similar optimization after neccesary changes.

These partial objectives are to be met:

- Become familiar with the design of cam profile for valve lift.
- Become familiar with the design environment of Valkin, Valdyn and Isight.
- Perform the design and optimization of valve lift curve with respect to established limits.

Chapter 3

Theoretical Background

In the first part of this chapter, the cams are introduced, their usage is described and the types of cams are listed. Then, several methods of cam design are introduced. The design that is selected for the optimization task is described in more detail. In the second part, the optimization process is introduced and the single parts as Design of Experiment, stochastic and deterministic optimization methods are explained. In the end of this chapter, the software tools are introduced.

3.1 Cam

A cam is an eccentric wheel device designed to move a follower in a controlled fashion. It can transfer rotational motion to translational motion or rotational motion to rotational motion. It has a various use but common and critical application is in internal combustion engines. Other applications include early automated machine tools that were controlled by cams which were moving the blades. More recently, a dishwashers were programmed by using a series of cams to turn on and off the parts of the programme. In these applications, cams were mostly substituted by electronic control systems. Cams can be found in combustion engines where they are used for driving intake and exhaust valves of valvetrain mechanism or for driving pistons in fuel injection pump. Usually, a spring is used to maintain contact between the cam and the follower. In some special applications, form-closed cam is used where the follower is guided by a track or a groove and there is no need for a spring.

3.2 Valvetrain Mechanisms

Valvetrain mechanisms control the air intake and the exhaust of post combustion gases in combustion engines. There is usually a camshaft providing the opening and closing of the valves. In the four-stroke engine the camshaft is always running half the speed of the crankshaft. There are several types of valvetrain mechanisms that differ by topology. Valvetrain designs can fundamentally be divided into three main groups. Overhead valve (OHV), overhead cam (OHC) and direct attack (DA). These types can be seen in figure 3.1 and the main parts can be identified.

- DA - the driving torque must be transferred from crankshaft to camshaft - usually by a chain or toothed belt. The follower mechanism consists only of the valve, so the advantage is low mass and low inertia. On the other hand, there is sliding contact between the roller and the follower.

- OHC - the camshaft is driven in the same fashion as DA. Here, the motion is transferred by the rocker arm to the valve. The rocker arm can be designed with a roller which prevents the wear between the cam and the rocker arm and at the same time prevents loss caused by friction. Slightly different configurations of OHC topology than the one in figure 3.1 are used in current engines.

- OHV - the camshaft is driven directly from crankshaft by gear. The pushrod transfers the motion to the rocker arm which drives the Valve. This topology features more moved mass than the previously mentioned topologies. Thus the OHV topology is not suitable for higher rotation speeds and the application is in heavy engines.

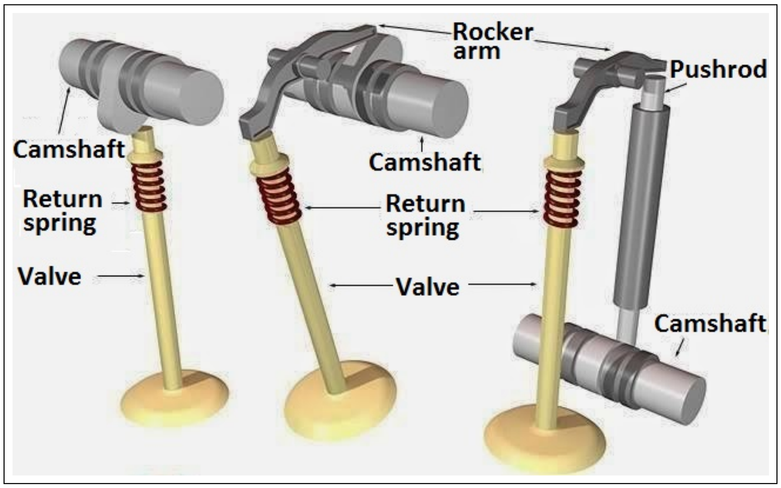


Figure 3.1: Valvetrain designs from the left: DA, OHC, OHV

Current valvetrain systems are a bit more complicated. In the second part of this text, one specific case will be described in detail. While designing the valvetrain, there are a number of key factors and constraints that must be satisfied to achieve the most efficient design. These are the limit of maximum jerk, maximum stress between the cam and the follower and so on. These will be discussed in more detail in subsequent chapters.

3.3 Cam Design

In the cam design process, the valve lift curve has to be described mathematically as a dependency of valve lift on the crankshaft angular position. Then the cam shape can be obtained easily. The easiest concept of the cam manufacturing process is that the camshaft rotates at constant angular velocity and the machine tool of a shape of the follower follows the follower's lift curve. According to [Norton, 2009] there are some basic rules the lift curve should fulfill. The follower lift function must be continuous through the first and second derivatives and the third derivative should be finite. The first derivative is velocity and it determines the follower's kinetic energy. The second derivative is acceleration and it is proportional to forces necessary for the follower actuation. The third derivative is jerk and has an influence on system vibration. While designing the lift curve, these rules should be remembered. The lift curve can be designed in various ways as a sequence of polynomials or sequence of splines. The polynomial design is described in more detail since it is used for optimization task in the second part of this text.

3.3.1 Polynomial Design

Polynomial functions can be used for designing valve lift curve. To design a lift curve for opening the valve, firstly the degree of polynomial has to be determined. This is done by writing down the boundary conditions that should be satisfied. There are 3 boundary conditions for the start of the segment as:

$$y(start) = 0, \qquad v(start) = 0, \qquad a(start) = 0 \tag{3.1}$$

This conditions indicates zero initial lift, velocity and acceleration. Additionally, 3 boundary conditions are defined for the end of the segment as:

$$y(end) = 1, \qquad v(end) = 0, \qquad a(end) = 0 \tag{3.2}$$

The lift at the end of the segment was set to unity, acceleration and velocity was set to zero in order to allow the backward motion in the next segment. These are 6 boundary conditions that can be satisfied by a 5th degree polynomial for valve lift y. The derivatives describing the velocity v, acceleration a and jerk j follow as:

$$\begin{aligned}
y(\phi) &= C_1 + C_2\left(\frac{\phi}{\theta}\right) + C_3\left(\frac{\phi}{\theta}\right)^2 + C_4\left(\frac{\phi}{\theta}\right)^3 + C_5\left(\frac{\phi}{\theta}\right)^4 + C_6\left(\frac{\phi}{\theta}\right)^5 && (3.3)\\
v(\phi) &= \frac{1}{\theta}\left(C_2 + C_3\cdot 2\left(\frac{\phi}{\theta}\right) + C_4\cdot 3\left(\frac{\phi}{\theta}\right)^2 + C_5\cdot 4\left(\frac{\phi}{\theta}\right)^3 + C_6\cdot 5\left(\frac{\phi}{\theta}\right)^4\right) && (3.4)\\
a(\phi) &= \frac{1}{\theta^2}\left(C_3\cdot 2 + C_4\cdot 6\left(\frac{\phi}{\theta}\right) + C_5\cdot 12\left(\frac{\phi}{\theta}\right)^2 + C_6\cdot 20\left(\frac{\phi}{\theta}\right)^3\right) && (3.5)\\
j(\phi) &= \frac{1}{\theta^3}\left(C_4\cdot 6 + C_5\cdot 24\left(\frac{\phi}{\theta}\right) + C_6\cdot 60\left(\frac{\phi}{\theta}\right)^2\right) && (3.6)
\end{aligned}$$

Here, ϕ goes from 0 to $\phi = \theta$ so the relative coordinate $\frac{\phi}{\theta}$ goes from 0 to 1. Note that the lift unit here is 1, the velocity is of unit $1/deg$, the acceleration is of unit $1/deg^2$ and jerk is of unit $1/deg^3$. The actual values of the lift can be obtained by rescaling the plot (by multiplying the y-axis values by maximal lift h and the x-axis values by the segment length θ). The same can be done for velocity, acceleration and jerk. If the actual values of velocity, acceleration and jerk in time domain are of interest, the values have to be rescaled by actual angular velocity having units deg/s as follows:

$$v_{[mm/s]} = v_{[mm/deg]}\cdot\omega_{[deg/s]},\;\; a_{[mm/s^2]} = a_{[mm/deg^2]}\cdot\omega^2_{[deg/s]},\;\; j_{[mm/s^3]} = j_{[mm/deg^3]}\cdot\omega^3_{[deg/s]} \quad (3.7)$$

If the six boundary conditions from equations 3.1 and 3.2 are enforced and the linear system of equations 3.3 - 3.6 is solved, the constants $C_1 \dots C_6$ are obtained according to [Norton, 2009] as:

$$C_1 = 0, \qquad C_2 = 0, \qquad C_3 = 10, \qquad C_4 = -15, \qquad C_5 = 6, \qquad C_6 = 0 \quad (3.8)$$

The resulting curves of lift, velocity, acceleration and jerk for follower rise can be observed in figure 3.2.

If a single dwell profile is designed, only one polynomial can be used with defined lift in the middle of the only segment. Due to extra points with prescribed position, the order of the polynomial must be increased. Higher order polynomials are not preferred in cam design since they can behave uncertainly (unwanted oscillations) according to [Norton, 2009]. This is why a sequence of lower order polynomials is used for valve lift curve description in Valkin program. This profile is described in the end of this chapter.

3.4 Optimization Methods and Approach

Optimization is defined as "Finding an alternative with the highest achievable performance under the given constraints, by maximizing desired factors and minimizing undesired ones. Prac-

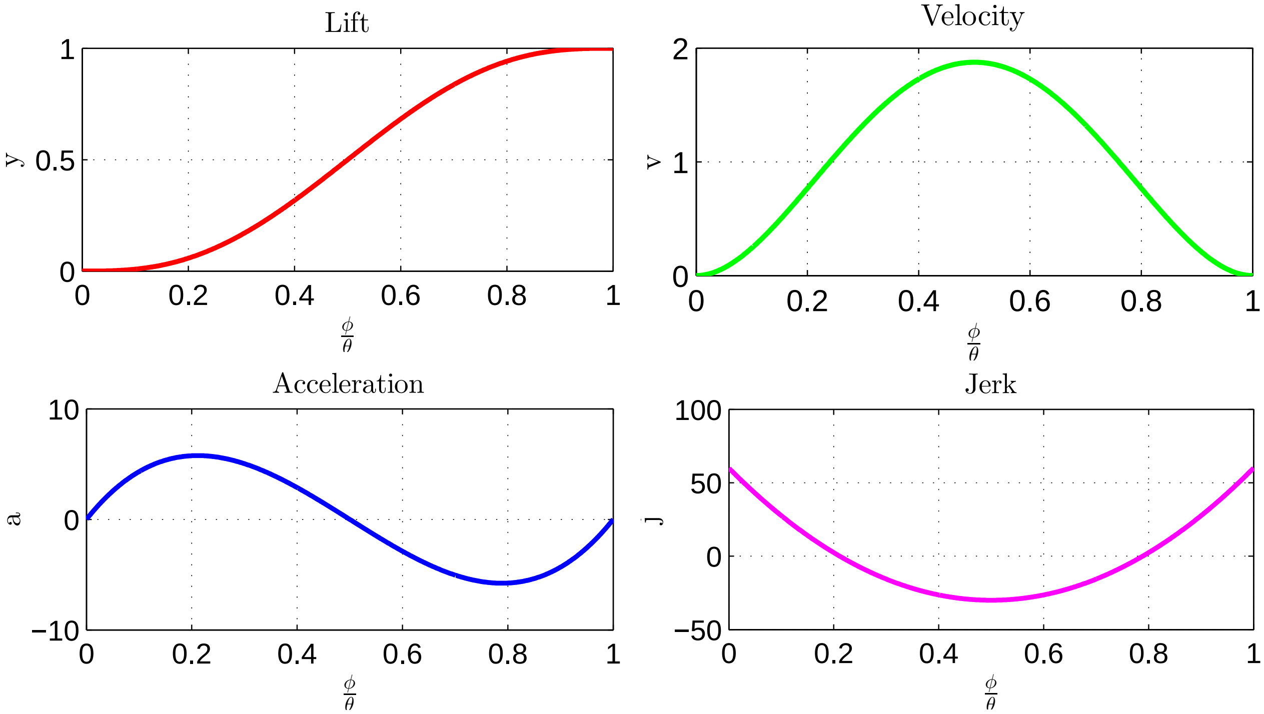


Figure 3.2: Valve lift, velocity, acceleration and jerk during the follower rise.

tice of optimization is restricted by the lack of full information, and the lack of time to evaluate what information is available.[1]*"*

The optimization process includes some of the following steps (some of them can be omitted under specific conditions). The first feasible points can be found by *Design of Experiments (DOE)* and they can be used by *Response Surface Modeling* in order to interpolate the points observed in *DOE*. If the response surface is available, the extreme can be approximated without the model or experiment evaluation [Cavazzuti, 2013]. With this identified, deterministic optimization methods are applied in order to find the local minimum or maximum. Methods for stochastic optimization can be used without the preceding steps as they are designed to explore the whole design space. The response surface modeling is not used throughout this work so it is not described here.

3.4.1 Design of Experiments

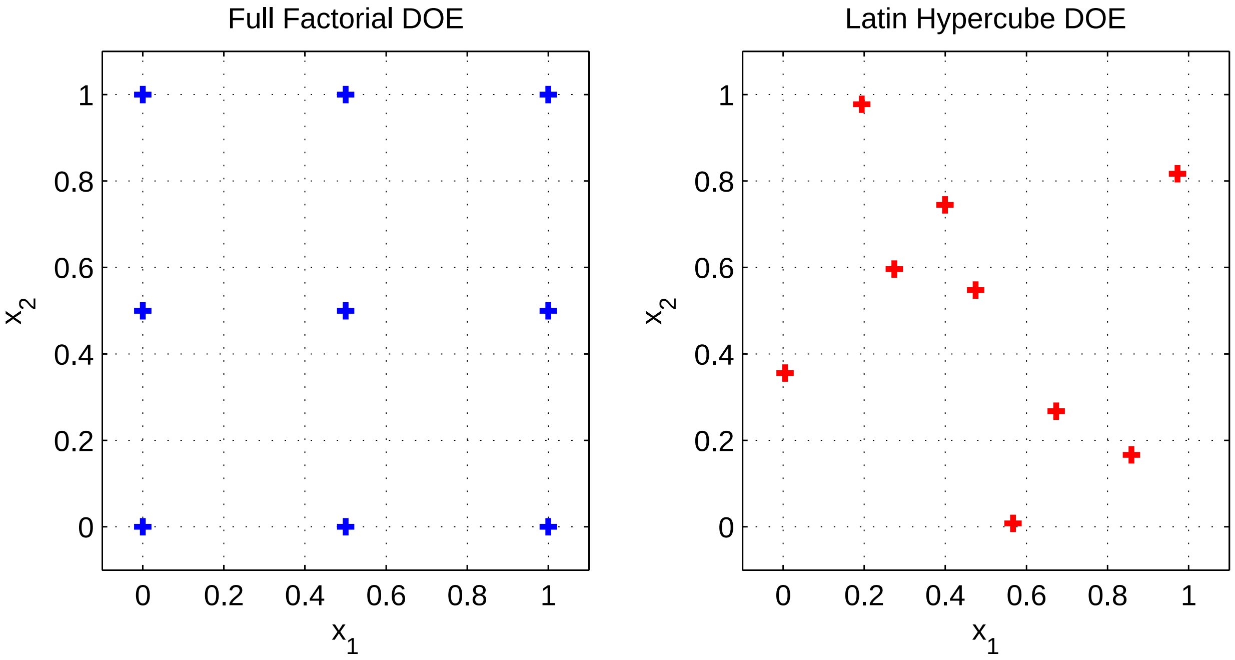
DOE can be used as the first step in an optimization process. The DOE method produces input data for number of experiments that are then executed. Then, the data can be used to model a response surface in order to analyze the influences of input factors to output variables or to select a starting point for another optimizing method. There are several techniques to generate

[1] Shortened optimization definition from `http://www.businessdictionary.com/definition/optimization.html` [accessed 04-05-2017]

the DOE matrix. Here, only two of them are briefly described since they are used in the following chapters. More detailed explanation can be found in chapter 2 of [Cavazzuti, 2013].

DOE Method 1: Full Factorial

The full factorial DOE matrix produces all the possible combinations of selected variables and levels. Considering k input factors and l levels for each factor, the DOE matrix would be of size l^k. Therefore, if two factors with three levels would be chosen, it results in $3^2 = 9$ design points. It follows that the number of experiments grows with both the number of factors and the number of levels. On the other hand, the number of levels determinates the coarseness of the obtained design, so some effects might not be observed. In order to create this matrix, the Matlab function `fullfact` can be used. In figure 3.3 a, the `fullfact` function was used to create 9 design points in square 1-by-1. It can be seen that the points are arranged in some sort of regular grid.



a) b)

Figure 3.3: Comparision of DOE methods a) Full Factorial, b) Latin Hypercube.

DOE Method 2: Latin Hypercube

The latin hypercube DOE matrix attempts to gain more out of the same amount of experiments. Therefore, some randomness is included and some iterations are done before the final DOE matrix is produced. For generating latin hypercube matrix, Matlab function `lhsdesign` can be used. It has several properties: one can choose the number of iterations or the criterion to be minimized. The criterion can be the minimum distance between the points so it should lead to the design that is well spread out in the design space, or it can be a minimum correlation. In figure 3.3 b, the Latin hypercube design can be seen for the same number of points as full factorial design. The Latin Hypercube has also one interesting property: if one of the factors is omitted, the number of nonoverlapping points remains the same (this could be visualized by projecting the points either to axis x_1 or x_2). More about latin hypercube design can be found in [Viana, 2015].

3.4.2 Deterministic Optimization

Deterministic optimization methods are based on mathematical properties of functions. Often, gradient or hessian methods are used in order to determine the subsequent evaluation point. This is why deterministic methods only work for single objective functions. There is no random component in these optimization methods. Due to this fact, the optimization process is replicable and for similar initial conditions, one should obtain the same solution if ran again. On the other hand, these methods tend to converge to local minimum so combination with *DOE* or stochastic optimization can be beneficial.

Deterministic Method 1: Downhill Simplex

Simplex methods are using $n+1$-gon in n-dimensional space. It can be illustrated in two-dimensional space with a triangle. One step can be seen in figure 3.4. There are three basic rules of this method:

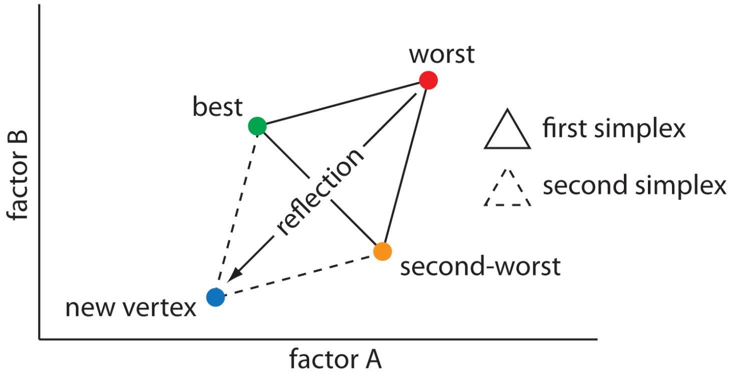


Figure 3.4: Simplex method illustration.[3]

- The value of objective function is computed in all vertices of the n+1-gon. The vertex with the worst value of objective function is omitted and a new point is generated reflecting

[3] http://sherrytowers.com/wp-content/uploads/2014/07/simplex.jpg [accessed 23-04-2017]

the worst point over the remaining n-point centroid.

- The same vertex can not be reflected again in the next round so if the same vertex has still the worst value, the second worst is reflected.
- If a certain point is old enough, the simplex is shrinked to smaller dimensions. This moment can be determined differently method to method. It should be avoided for simplex to make more than 6 iterations with one vertex fixed in two-dimensional space because then the method could only go to the points that were already evaluated. In higher dimensional spaces, the limiting oldness has to be determined.

More detailed explanation of this method can be found in [Lederer and Brousil, 1989]. The simplex method has several modifications which allow for example non-uniform edges. This should fasten the process of getting closer to the local minimum. More about this topic can be found in chapter 4 of [Cavazzuti, 2013].

Deterministic Method 2: Hook Jeeves

This method searches through the design space along search directions in two basic steps. Exploratory moves observing the properties of objective function in the search directions are combined with pattern moves that are accelerating the optimization process. The search directions are usually chosen to be the coordinates.

- Exploratory search - firstly, the point $\mathbf{x}_0$ is evaluated. Then, a move is done in the direction of first coordinate. If it leads to a better objective function, this point is kept. If it leads to a worse value of objective function, it tries the step in the opposite direction of the same coordinate. If it leads to a better objective function, the point is kept. If it leads to a worse objective function, the original point is kept. This is done in all the search directions and the point $\mathbf{x}_1$ is approached.
- Pattern search - is done as a move in the same direction as the previous exploratory search. The coordinates are shifted by $(\mathbf{x}_1 - \mathbf{x}_0)$. It is checked if this leads to a better objective function. If it does, the exploratory search is performed around the new point. If it does not, the exploratory search is performed around the point $\mathbf{x}_1$.

These steps are visualized in figure 3.5. This method does not require continuous design space or objective function. It does not evaluate the gradient of the function. More details about this method can be found in [Lederer and Brousil, 1989] or [Kirgat and Surde, 2014].

[5] https://www.researchgate.net/publication/220308495_Application_of_Derivative-Free_Methodologies_to_Generally_Constrained_Oil_Production_Optimization_Problems [accessed 08-05-2017]

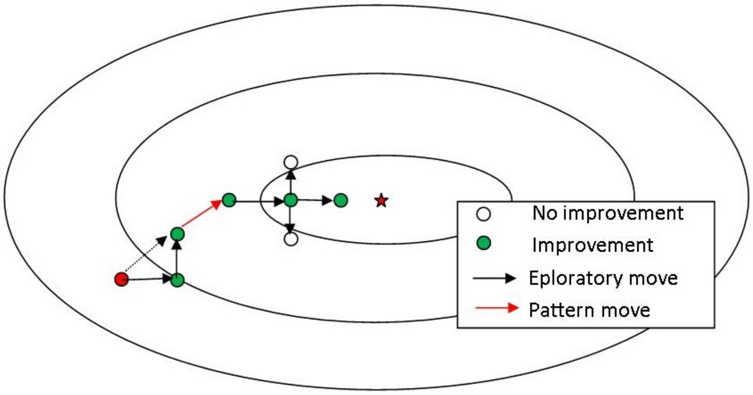


Figure 3.5: Hook Jeeves method illustration.[5]

Deterministic Method 3: Sequential Quadratic Programming

This method approximates the objective function in the current location by quadratic function and based on that, it determines the next iteration. Moreover, it allows the solution of problems in constrained design space. This method requires the objective function to be convex. The minimum is the point of interest. Firstly, the function is approximated as Taylor polynomial of second degree as:

$$f(\mathbf{x}) \approx \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{x}^T\mathbf{g} + \mathbf{c} \tag{3.9}$$

If equation 3.9 is differentiated with respect to $\mathbf{x}^T$, the following equation is obtained as follows.

$$\Delta f(\mathbf{x}) = \mathbf{H}\mathbf{x} + \mathbf{g} \tag{3.10}$$

the minimum of convex function is found in point where $\Delta f(\mathbf{x}) = 0$. So the solution is found as follows.

$$\mathbf{x}^* = -\mathbf{H}^{-1}\mathbf{g} \tag{3.11}$$

Then, $\mathbf{x}^*$ is the new iteration and the process starts over with this new iterate. Note that this method requires hessian (matrix of combined derivatives $H_{ij} = \dfrac{\partial f}{\partial x_i \partial x_j}$) and gradient $\mathbf{g} = \dfrac{\partial f}{\partial \mathbf{x}}$ (vector of first derivative coefficients). These can be computed analytically if the function f is known. If the function f is unknown, both of them can be approximated. Because approximating hessian by numerical differentiations requires a lot of function evaluations which is often computationally expensive, some algorithms as BFGS for approximating hessian iteratively have been developed.

This was a solution to a simplified problem without any constraints. More information about sequential quadratic programming can be found in chapter 4 of [Cavazzuti, 2013] or in more detail in chapter 3 of [Mikuláš, 2013], where the solution of constrained problem is carried out.

3.4.3 Stochastic Optimization

Stochastic optimization methods include randomness in their procedures. They are usually based on some phenomena that were observed in nature. Population evolution was the inspiration for Evolutionary algorithms, the birds searching for food were the inspiration for particle swarm algorithm. There are many other stochastic algorithms such as Simulated Annealing that will not be discussed in this thesis since they are not used in the next part of this text.

Stochastic Method 1: Evolutionary Algorithm

Evolutionary algorithms imitate the process of natural selection through generations. "Better" individuals have a bigger chance of becoming the parent of a new offspring. The process consists of 4 steps:

- Initialization - the initial population is selected.
- Evaluation - the fitness is evaluated for all the individuals of the generation.
- Selection - the parents are selected based on fitness and randomness.
- Recombination - the new population is generated by combining the properties of parents. Additionally, mutation is implemented here in order to supply some new randomness to the system.

These steps, except for the first one, are executed until the stopping criteria are achieved or the maximum number of fitness function evaluations is reached. Due to the effects of randomness, convergence must be evaluated over several iterations to achieve a solution. Moreover consideration must be given to the extent of evaluation cycles to provide an optimal solution. In the case of the finite element analysis, it could require hours of computation time. These algorithms do lead themselves to parallel evaluation where computational resources are available. More information about evolutionary optimization, implementation and modifications can be found in chapter 5 of [Cavazzuti, 2013].

Stochastic Method 2: Particle Swarm

Particle swarm algorithm was founded by Kennedy and Ebehart in 1995, this method is inspired by a shoal of fish or flock of birds searching for food. All the particles representing points in design space are given a velocity vector. They have the ability to change their behavior depending on the previous fitness and other particles' fitness. The process operates as follows:

- Initialize particles - particles are generated randomly in design space or around the given starting point and assigned the initial velocity.

- Calculate fitness values for each particle.
- Each particle updates its personal best location and local best location if the actual value of fitness is better than the stored one. The local best location is the best value of fitness of the particle and its neighbors. The neighbors are defined based on chosen topology.
- Update the velocity for each particle as:

  $$\mathbf{v_i}^{(n)} = W\mathbf{v_i}^{(n-1)} + K_1 r_1(\bar{\mathbf{x}}_i - \mathbf{x_i}^{(n-1)}) + K_2 r_2(\tilde{\mathbf{x}}_i - \mathbf{x_i}^{(n-1)}) \tag{3.12}$$

  where $\mathbf{v_i}$ is velocity of the particle, $\bar{\mathbf{x}}_i$ is the personal best location, $\tilde{\mathbf{x}}_i$ is the local best location, the K_1 is cognitive learning factor, K_2 is social learning factor, W is the inertia factor of the individual and r_1, r_2 are random numbers taken from interval $(0, 1)$. The superscript is identifying whether the values are new $^{(n)}$ or old $^{(n-1)}$.
- Move the particles by the new velocity vector.

These steps, except for the first one, are executed until the stopping criteria are achieved or the maximum number of steps is done. Various topologies can be chosen as shown in figure 3.6. These topologies are determining the neighbors for the third step of this algorithm. More information about the particle swarm method can be found in [McCullock, 1999] or in chapter 5 of [Cavazzuti, 2013].

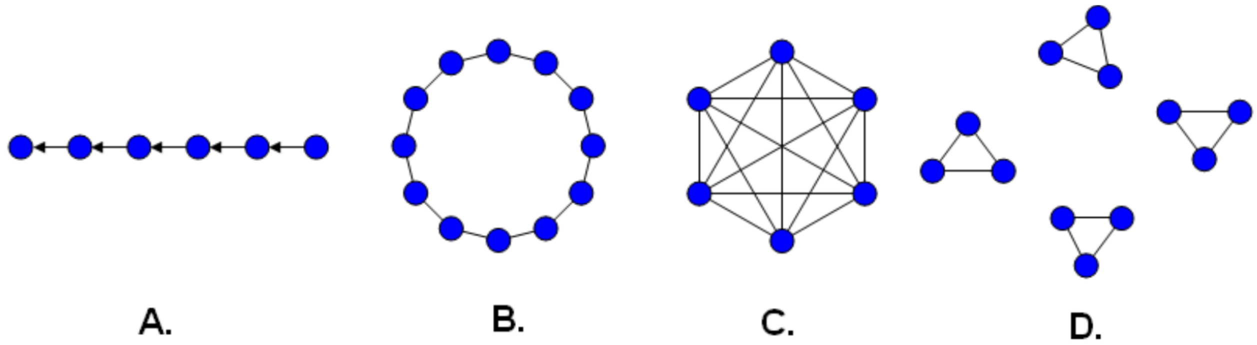


Figure 3.6: (A) Single-sighted, where individuals only compare themselves with the next best. (B) Ring topology, where each individual compares himself only with the adjacent ones. (C) Fully connected topology, where everyone is compared together. (D) Isolated, where individuals only compare themselves with those within specified groups.[7]

[7] `http://mnemstudio.org/particle-swarm-introduction.htm` [accessed 26-05-2017]

3.5 Software Tools for Optimization and Analysis

In this section, the software that is used for optimization task in the next chapter is shortly introduced.

3.5.1 Matlab

Matlab is a product of MathWorks company and is a toolset for developing analytical framework in a very wide range of industries and applications. In this software, one can develop programs using simple programming language. There is a number of predefined functions and new ones can be defined as well. The work with matrices is convenient. Plots of functions or data can be displayed and saved easily.

3.5.2 Isight

Isight is a commercial software product of DS Simulia developed specifically for optimization studies. It allows the user to configure evaluation loops controlled by user defined process components. Inside the loops, application components are used. The following process components were used in the work discussed in this thesis:

- Task - defines input for a single loop execution.
- DOE - design of experiments defines a set of input values for a number of loop executions. Additionally, it stores the execution results. Some techniques can be selected or the DOE matrix can be generated externally and input by referring to the file containing DOE matrix.
- Optimization - this block evaluates the loop with the given starting point and then it chooses the next iterate input parameters based on the selected optimization algorithm. Several optimization methods including deterministic and stochastic methods or a combination of both is available. Moreover, the limits of the design space, stopping criterion, objective function and output constraints have to be set.
- Exploration - component similar to optimization component, but slightly different set of optimization techniques is available. A set of methods can be planned to work one after another.

The following application components were used in the work discussed in this thesis:

- Simcode - a general component allowing to automatically edit text input, executing the solver and extracting the output data from output text file.

- Pause - a component that can wait for a specified time or until some file is present in a specified folder. This feature is also implemented directly in Simcode block.
- Excel - a component that inputs the content of specified cells, then evaluates all the relations and reads the content from output mapped cells.
- System command - a component that can be used for deleting specified file or killing some job. The windows bash suits well for this purpose.

The input and output parameters for single blocks are to be configured and mapped to their preceding and following blocks to ensure required data flow.

3.5.3 Valkin

Valkin is a product of Ricardo Software specifically designed for the solving of combustion engine valvetrain kinematics. Each part of the modeled valvetrain is represented by software block characterized by their input parameters and connected into a complex model using drag and drop approach. Output properties are available for storing - this must be configured for the outputs that are of interest. Additionally, interactions (connections) of the used blocks must be defined. Valkin also solves several dynamic output parameters in a quasi-static way, so it can predict for instance contact stresses. The solution can be obtained for several engine rotation speeds and then evaluated.

The whole model is saved as a text editable file so it can be edited by an external program. Additionally, this submission is possible from windows command line. This way the process of editing and executing the model is possible without running the graphical user interface and can be automated conveniently. The results can be read from a text file. Valkin also produces some files that are necessary for running Valdyn and in the Valdyn model, these files must be referred to.

Polynomial Valve Lift Curve Definition

A cam in a combustion engine is of type Rise - Fall - Dwell. The rise and fall parts obey a similar structure. There is a ramp to transfer dwell to rise. This ramp starts lifting the valve before the opening sequence. Then, the opening sequence is active, followed by closing sequence. Finally, the closing ramp slows the valve down and makes sure it is seating gently on the seat. Then, the cam base circle is active until the opening ramp comes again. The ramps are usually asymmetric in automobile engines application. The opening ramp acts as a boundary condition for position and velocity at the start of the opening sequence and the closing ramp acts as a boundary condition at the end of the closing sequence. Due to this, the opening and closing sequences cannot be symmetric.

As previously discussed, the focus of this study is the derivatives of angle lift dependency. The first derivative is velocity, the second derivative is acceleration (representing force) and the third derivative is jerk. The typical course of these functions can be seen in figure 3.7 The asymmetry is obvious. Note that the valve lift is displayed against the crankshaft angular position. It can be seen that approximately 200 *deg* is the duration of the whole lift including ramps. Since the camshaft rotates half the speed of crankshaft, the lift is executed by 100 *deg* of the cam. The rest is a circular shape executing the low dwell. The ramps are visible as constant parts in velocity profile. The transition from the low dwell to the ramp is realized by an acceleration step. This results in the infinite jerk. It can be seen that these are the only points where the jerk continuity is violated.

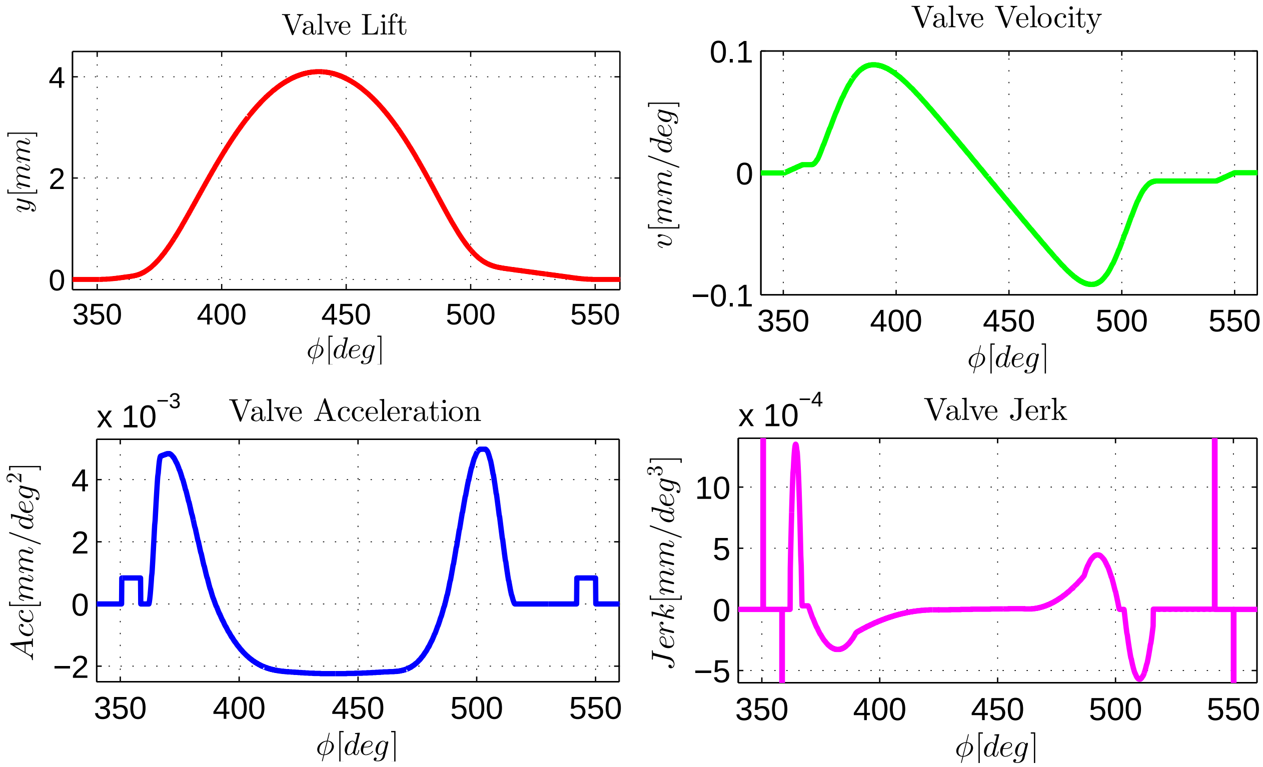


Figure 3.7: Valve lift, velocity, acceleration and jerk during the rise-fall depending on crankshaft angular position.

Valkin allows the user to choose one of the following valve lift curve definitions:

- MPOL - is a multi-polynomial definition of valve lift curve which consists of opening and closing sequences. Each sequence consists of a number of polynomials.

- SPLINE - the valve lift is defined by a set of B-spline curves. The user can define the lift curve in position, velocity or acceleration coordinates. The rest is computed by integration and boundary conditions or differentiation. The curve is defined by a set of control points where the curve segments come together.

- User defined - the user can deliver a file describing the valve lift curve in angle-lift coordinates. This can be used for externally generated or measured valve lift.

In practice, MPOL is the most often profile definition used in Valkin.

MPOL

Since MPOL is the most frequently used valve lift curve definition, it will be described here in more detail. The opening sequence consists of six segments of length θ_i. The valve lift in every segment is described by a polynomial of 5th degree as follows:

$$y(\phi_i) = C_{i1} + C_{i2}\left(\frac{\phi_i}{\theta_i}\right) + C_{i3}\left(\frac{\phi_i}{\theta_i}\right)^2 + C_{i4}\left(\frac{\phi_i}{\theta_i}\right)^3 + C_{i5}\left(\frac{\phi_i}{\theta_i}\right)^4 + C_{i6}\left(\frac{\phi_i}{\theta_i}\right)^5 \tag{3.13}$$

the first derivative describing the velocity through the segment follows as:

$$v(\phi_i) = \frac{1}{\theta_i}\left(C_{i2} + C_{i3}\cdot 2\left(\frac{\phi_i}{\theta_i}\right) + C_{i4}\cdot 3\left(\frac{\phi}{\theta_i}\right)^2 + C_{i5}\cdot 4\left(\frac{\phi_i}{\theta_i}\right)^3 + C_{i6}\cdot 5\left(\frac{\phi_i}{\theta_i}\right)^4\right) \tag{3.14}$$

the second derivative describing the acceleration through the segment follows as:

$$a(\phi_i) = \frac{1}{\theta_i^2}\left(C_{i3}\cdot 2 + C_{i4}\cdot 6\left(\frac{\phi_i}{\theta_i}\right) + C_{i5}\cdot 12\left(\frac{\phi_i}{\theta_i}\right)^2 + C_{i6}\cdot 20\left(\frac{\phi_i}{\theta_i}\right)^3\right) \tag{3.15}$$

and finally the third derivative describing the jerk through the segment follows as:

$$j(\phi_i) = \frac{1}{\theta_i^3}\left(C_{i4}\cdot 6 + C_{i5}\cdot 24\left(\frac{\phi_i}{\theta_i}\right) + C_{i6}\cdot 60\left(\frac{\phi_i}{\theta_i}\right)^2\right) \tag{3.16}$$

Here the ϕ_i goes from 0 to $\phi_i = \theta_i$. So the ratio of $\dfrac{\phi_i}{\theta_i}$ goes from 0 to 1.

For connecting the single segments into opening sequence, conditions for start and end points of every segment are stated in table 3.1. Additionally, the same conditions can be seen in figures 3.8-3.10. These curves are the solution of the system of linear equations (there are 36 unknown coefficients C_{ij}). Before solving these equations, the parameters b, c, d, e and θ_1 - θ_5 of profile must be chosen. The angle θ_6 is computed as: $\theta_6 = \dfrac{Open\,period}{2} - \sum_{i=1}^{5}\theta_i$. Where the *Open period* is the duration in *deg* allowed for valve lift. This is determined by the engine configuration but normally, it is about half of the crankshaft revolution.

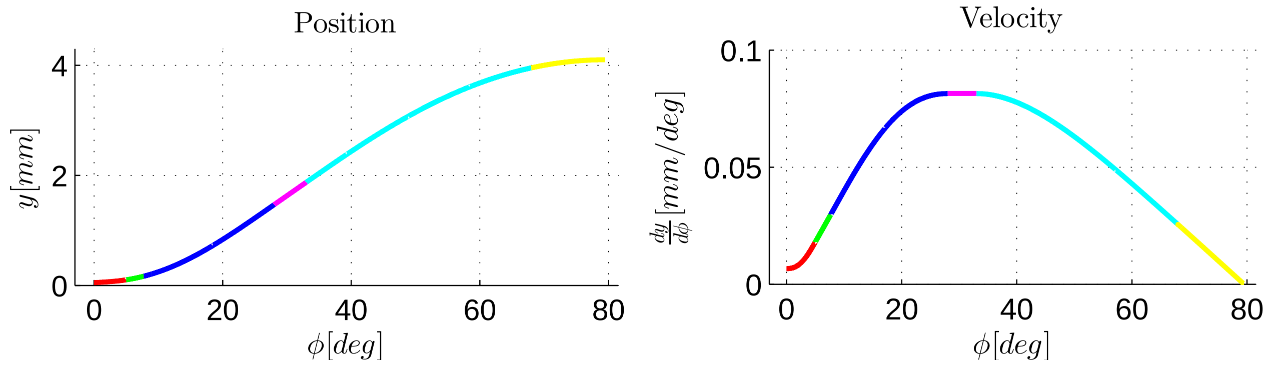
For opening and closing sequences the same model of six polynomials of fifth degree connected one to another is used. The continuity is guaranteed until the second differentiation,

Table 3.1: Values of position, velocity, acceleration and jerk on the start and end of segments. The symbol =↑ means equality condition with the end value of the preceding segment. The symbol =↓ signs equality condition with the start value of the following segment.

segment point	Lift	Velocity	Acceleration	Jerk
1 start	y_r	v_r	0	$\frac{A_f \cdot e}{\theta_1}$
1 end	=↓	=↓	A_f	$\frac{(d-1)A_f}{\theta_2}$
2 start	=↑	=↑	=↑	=↑
2 end	=↓	=↓	$A_f \cdot d$	$\frac{(d-1)A_f}{\theta_2}$
3 start	=↑	=↑	=↑	=↑
3 end	=↓	=↓	0	$-c\frac{A_f \cdot d}{\theta_3}$
4 start	=↑	=↑	=↑	0
4 end	=↓	=↓	0	0
5 start	=↑	=↑	=↑	$-c\frac{A_f \cdot d}{\theta_3}$
5 end	=↓	=↓	$A - n \cdot \cos(b \cdot \theta_6)$	$b\frac{\pi}{180}A_n \sin(b \cdot \theta_6)$
6 start	=↑	=↑	=↑	=↑
6 end	$Lmax$	0	A_n	0

meaning the acceleration is continuous.[8] There are 9 variables that determine the shape of the opening sequence of the valve lift curve. Four out of these nine coefficients determine the shape and the remaining five define segment lengths. Note that the opening range angle is given by the engine settings, so the sixth segment's length can be easily calculated. As stated before, the closing part has the same design as the opening part. In the easiest case, the profile is symmetric. Then, the continuity in the place of connection of opening and closing parts is guaranteed. Otherwise the parameters of opening and closing sequences must be adjusted in order to achieve acceleration continuity.

[8] If the segment 4 (constant velocity) is omitted by setting $\theta_4 = 0$ the jerk continuity is achieved.



a) b)

Figure 3.8: Visualization of a) lift - angle, b) velocity - angle for valve opening sequence (they consist of six polynomial parts).

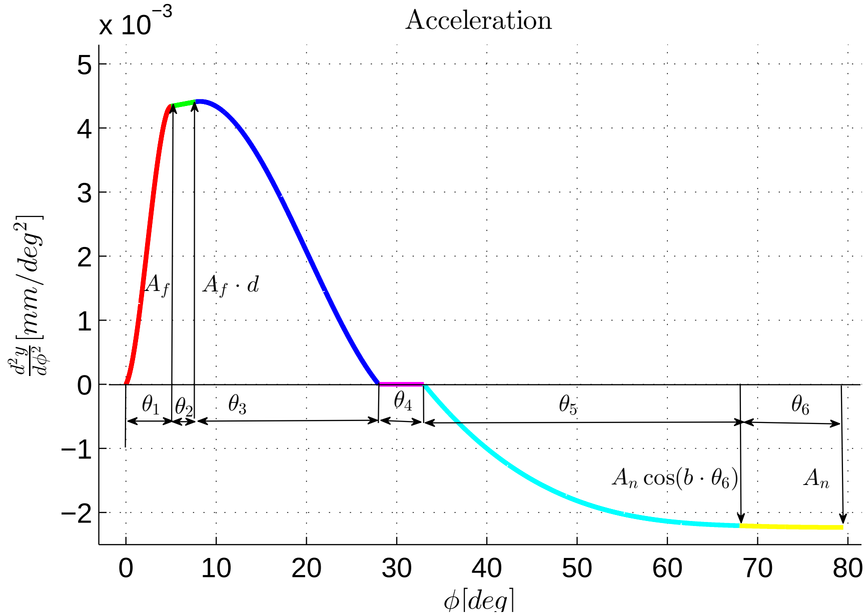


Figure 3.9: Visualization of acceleration - angle dependency for valve opening sequence (consists of six polynomial parts).

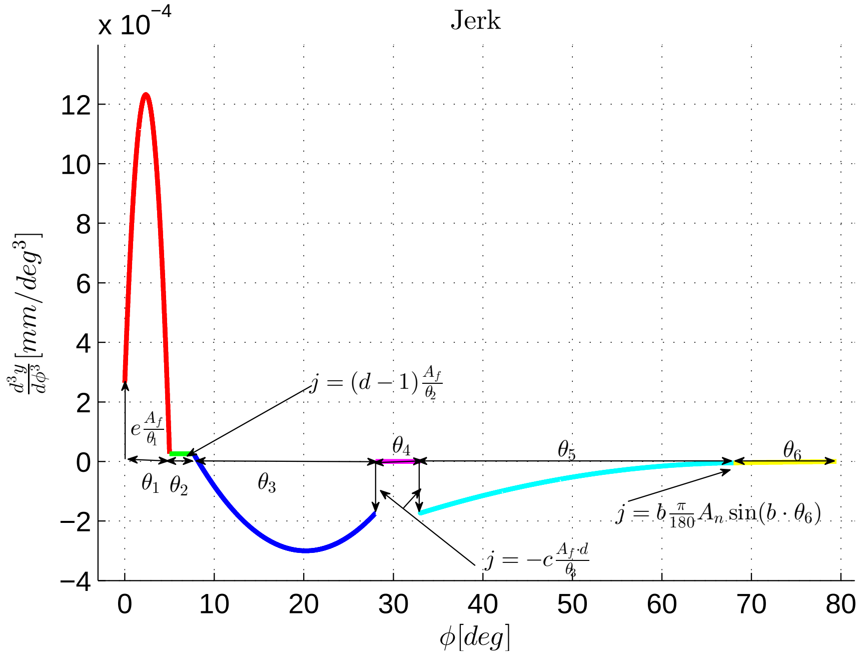


Figure 3.10: Visualization of jerk angle dependency for the valve opening sequence (consists of six polynomial parts).

3.5.4 Valdyn

Valdyn is a product of Ricardo Software specifically designed for the solving of combustion engine valvetrain dynamics. Each part of the modeled valvetrain is represented by software block characterized by their input parameters and connected into a complex model using drag and drop approach. Interactions (connections) of the blocks must be defined. Output properties are available for storing - this must be configured for the outputs that are of interest. Additionally, reduced finite element models can be included. The number of loadcases for evaluation can be specified. Loadcases can differ from each other typically by engine rotational speed. One can also define parameters to be modified through loadcases. These parameters can be used in blocks characterizing the model. There are always at least two revolutions of engine crankshaft to be observed if the four stroke engine is modeled. The solution is driven by integration method. In this text, only the Fourth-order Runge-Kutta-Nystrom explicit integration method was used. After that, the loadcases are solved and the results can be viewed as a caseplot or sumplot. Caseplot shows the dependency of observed quantity on crankshaft angle for chosen loadcase. Sumplot shows the dependency of selected statistic (maximum, minimum, mean, . . .) on loadcase (engine rotational speed). Submitting the computation and obtaining the results works exactly the same for Valdyn as for Valkin.

Chapter 4

Asymmetric Cam Design

In this chapter, the design and optimization process of valve lift curve is described. Firstly the optimization parameters are depicted. Then, the single components occurring in the optimization model are introduced. The last part of this chapter shows the Isight optimization model and discusses its sections in more detail with its results.

4.1 Optimization parameters

In a Valkin model there is MPOL block used for valve lift creation. This block is featuring 18 parameters that define the valve lift curve. In practice, only 16 of them are actually used. The same simplification was done in this thesis. Dropping two optimization parameters allows the exploration of the remaining 16 parameters to be more precise. Table 4.1 shows the parameters and limits. Their meaning was described in section 3.5.3. The limiting values were selected with respect to [VALDYN, 2016]. All the other parameters of the models are assumed to be constant. They are determined based on the engine specification and they are included in the given model.

Table 4.1: Optimization parameters and limits.

parameter	BO	CO	DO	EO	$TH1O$	$TH2O$	$TH3O$	$TH5O$
	BC	CC	DC	EC	$TH1C$	$TH2C$	$TH3C$	$TH5C$
	$[-]$	$[-]$	$[-]$	$[-]$	$[deg]$	$[deg]$	$[deg]$	$[deg]$
low limit	0.4	0.3	0.95	0.2	8	8	10	15
upper limit	1	1	1.3	2.5	18	18	24	35

4.2 The Model

In this section, the components used in the optimization process are described. Firstly, the Matlab polynomial design tool `CamPol` is introduced, then, Valkin and Valdyn are covered including their inputs and outputs. Lastly, the excel sheet for summarizing Valdyn results is described.

4.2.1 Matlab Cam Polynomial Design

When one designs multipolynomial profile using Valkin software, the aim is to find a combination of the optimization parameters in a fashion that all the limits of Valkin and Valdyn output parameters are satisfied and the valve area integral is maximized. Additionally, he must be aware of acceleration discontinuity in the place where the opening sequence joins the closing sequence. To achieve the continuity of acceleration, some sort of manual iterative process must have been applied.

In order to avoid time consuming iterative solutions, Matlab script `CamPol` was developed for solving the problem of acceleration discontinuity analytically. This script computes the coefficient CC for the closing sequence based on the remaining 17 parameters. Additionally, this script allows the angle θ_4 to be zero. Therefore the number of optimization parameters is reduced by one. After computing the coefficient CC, it must be checked if it lies within the specified limits. The inputs and outputs of CamPol script can be seen in figure 4.1. This script has several parts and functions. The overview of the Matlab CamPol script operation can be seen in figure 4.2. The limits for the output parameters are summarized in table 4.2 and their meaning follows.

Table 4.2: Matlab outputs and their allowed values.

Constrained output	CC	Max jerk	Af/An	JerkOK
Allowed interval	$\langle 0.3; 1\rangle$ [-]	$\langle 0; 0.0012\rangle$ $[mm/deg^3]$	$\langle 0; 3\rangle$ [-]	OK

- CC is the coefficient c of closing sequence computed in a way that the acceleration is continuous.

- Af/An is the absolute value of ratio of flank acceleration (the first positive peak) over the nose acceleration (the only negative peak).

- JerkOK checks if there are any unwanted oscillations in the acceleration profile. This is done by limiting the jerk to be positive or negative in specified sections.

- Valve integral is the area under the valve lift curve. This is going to be the objective of optimization and it is to be maximized.

In order to allow a good communication with Isight, the text input and output were carried out and the `CamPol` script was packed by matlab deploytool. This means that it can be executed as an executable file with extension `.exe`. Matlab runtime must be installed.

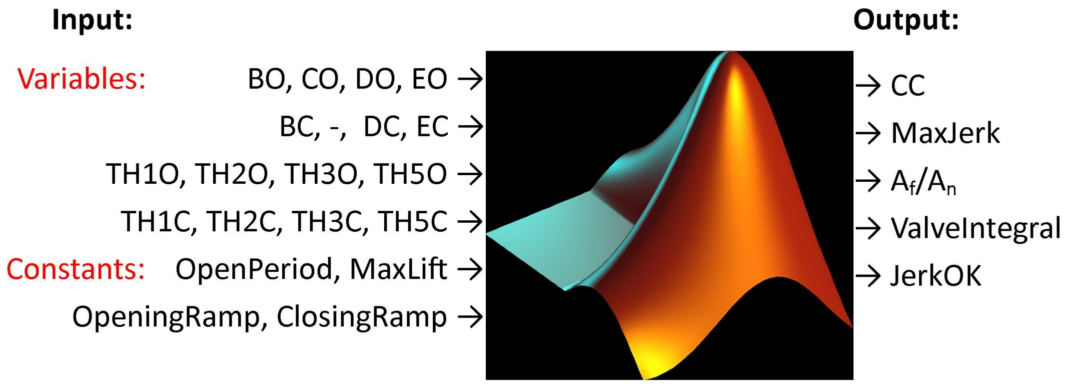


Figure 4.1: Inputs and outputs of Matlab `CamPol` script .

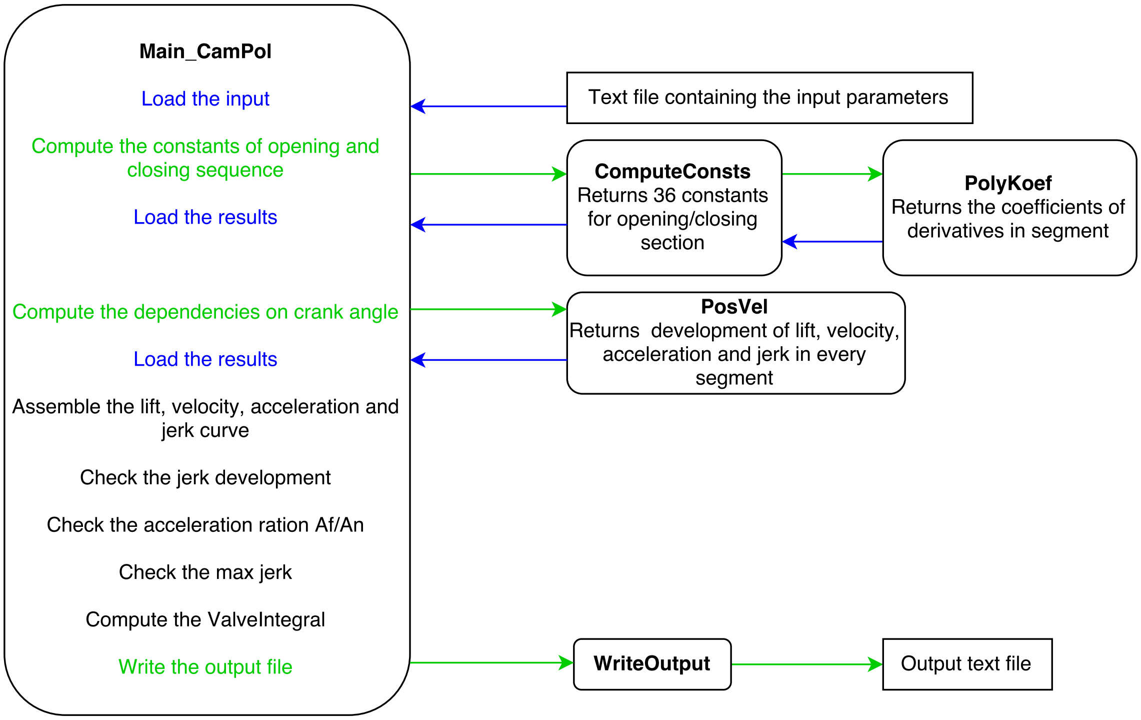
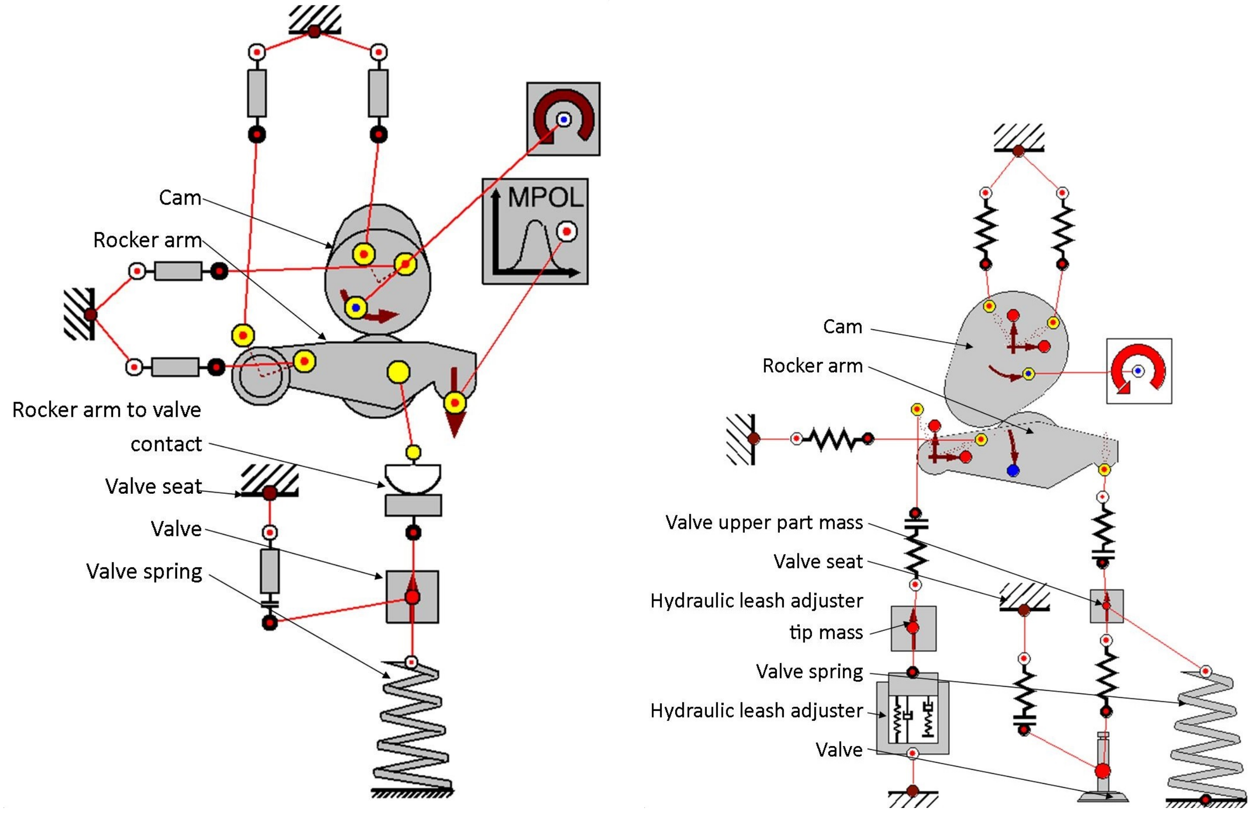

Main_CamPol
Load the input
Compute the constants of opening and closing sequence
Load the results
Compute the dependencies on crank angle
Load the results
Assemble the lift, velocity, acceleration and jerk curve
Check the jerk development
Check the acceleration ration Af/An
Check the max jerk
Compute the ValveIntegral
Write the output file
Text file containing the input parameters
ComputeConsts
Returns 36 constants for opening/closing section
PolyKoef
Returns the coefficients of derivatives in segment
PosVel
Returns development of lift, velocity, acceleration and jerk in every segment
WriteOutput
Output text file

Figure 4.2: Matlab script function overview.

4.2.2 Kinematic Model in Valkin

Kinematic model of valvetrain in Valkin was given for a template engine. The model can be seen in figure 4.3 a. Examples of real parts that are modeled by the blocks in Valkin can be seen in the figure 4.4.



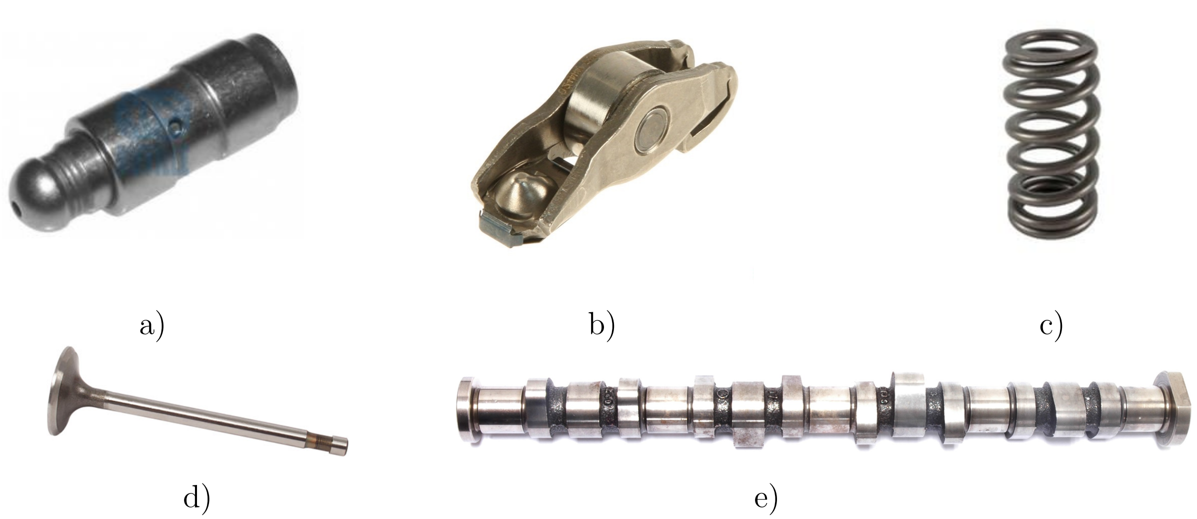
a) b)

Figure 4.3: Model of valvetrain in a) Valkin, b) Valdyn.

The constraints for Valkin output were depicted and their limits were set. The limits and constrained outputs can be seen in table 4.3. The constraints were chosen in the same fashion as it is done when manual valve lift curve design is applied. The limits were determined based on the engine specification. If a valve lift curve for different engine is to be optimized by this method, the limits have to be reconsidered assuming the new engine. They can be influenced by the valvetrain topology, used material or maximum operation speed of the engine.

Table 4.3: Valkin outputs and its allowed values.

Constrained output	Cam concavity	Contact stress	Spring cover factor
Allowed interval	$(-\infty; -60\rangle$ and $\langle 0; \infty)$ $[mm]$	$\langle 0; 1250\rangle$ $[MPa]$	$\langle 1.35; \infty)$ [-]



a) b) c)

d) e)

Figure 4.4: Real parts: a) Hydraulic lash adjuster, b) Rocker arm, c) Spring, d) Valve, e) Camshaft.[1]

The meaning of the outputs is described as follows:

- Cam concavity is the maximum negative radius occurring on the cam surface. This determines the parameters of machining tool that can be used for manufacturing the cam. If smaller grinding wheel is necessary, the manufacturing cost can increase. Cam concavity can also return a positive value. In that case, the cam can be manufactured by a grinding wheel of any size.

- Contact stress is the stress between the roller on the rocker arm and the cam. The contact stress is computed as Hertz stress. It uses reduced Young's modulus and forces from quasi static solution. Limiting of this contact stress is done to avoid pitting (fatigue of surface) of the cam or the roller.

- Spring cover factor is the ratio of spring force and inertia forces. Fulfilling this limit allows the proper function of the whole valvetrain mechanism without separation.

The blocks that return these outputs are marked in figure 4.6. The model was set to compute all the parameters only for one rotational speed. Shorter running times were obtained by this. The continuous overspeed (engine rotation speed when the output power is zero) was selected

[1] a) `https://www.motor-doctor.co.uk/products/2384081-rocker-tappet` [accessed 12-04-2017], b) `https://www.europaparts.com/roller-rocker-arm-06e109417s.html` [accessed 12-04-2017], c) `https://www.stevemorrisengines.com/steve-morris-engines-store/engine-parts-and-short-blocks/pac-1219x-drop-in-ls-valve-springs.html` [accessed 12-04-2017], d) `http://www.cb750supply.com/products/4/engine/69/valves-valve-seals` [accessed 12-04-2017], e) `http://www.lohen.co.uk/shop/gen-1-mini/engine-ecu/internals/newman-camshaft-detail` [accessed 12-04-2017]

because it is the most critical speed for contact stress and spring cover factor. The Valkin inputs and outputs can be seen in the figure 4.5. This is the way how the block is used in the Isight model.

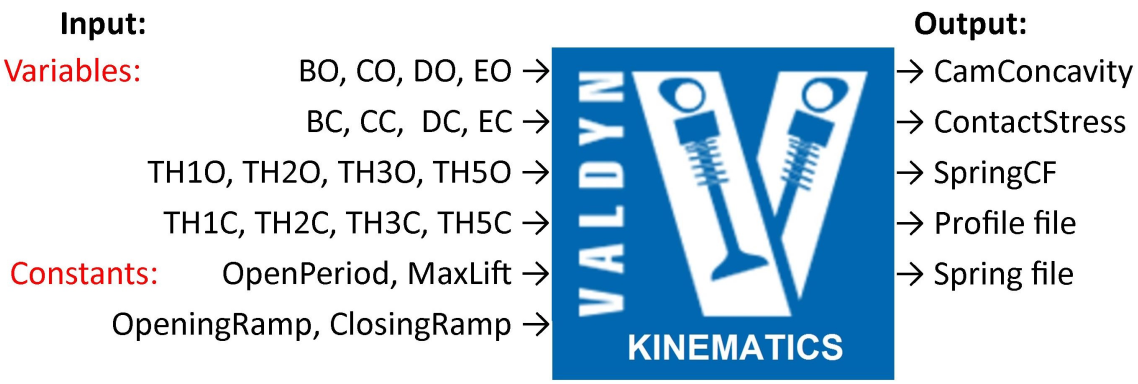


Figure 4.5: Valkin inputs and outputs.

4.2.3 Dynamic Model in Valdyn

Dynamic model of valvetrain in Valdyn was given for a template engine. The model can be seen in figure 4.3 b. The constraints for Valdyn output were depicted and their limits were set. The constraints with their limits can be seen in table 4.4. The constraints were chosen in the same fashion as it is done for manual valve lift curve design. The limits were determined based on the engine specification. If a valve lift curve for different engine is to be optimized by this tool, the limits have to be reconsidered assuming the new engine.

Table 4.4: Valdyn outputs and their allowed values.

Constrained output	Spring safety factor	Valve seating velocity	Spring surge	Cam to roller lash
Allowed interval	$\langle 1.1; \infty \rangle$ [-]	$\langle 0; 0.3 \rangle$ $[m/s]$	$\langle 0; 1 \rangle$ $[mm]$	$\langle 0; 0.05 \rangle$[mm]

The meaning of the outputs is described as follows:

- Spring safety factor is fatigue safety factor of the spring.
- Valve seating velocity is the velocity of the valve when the first contact of the valve with the seat occurs. Limiting this value should guarantee that the sealing capability will be preserved.
- Spring surge is the amplitude of oscillation of the spring. It is logged after the valve is closed to account for the increased probability of residual ”opening” due to transient oscillations in the spring.

- Cam to roller lash is the gap size if the contact between the roller and the follower is lost. The loss of the contact should be avoided, however, in the overspeed it can happen in the allowed limit.

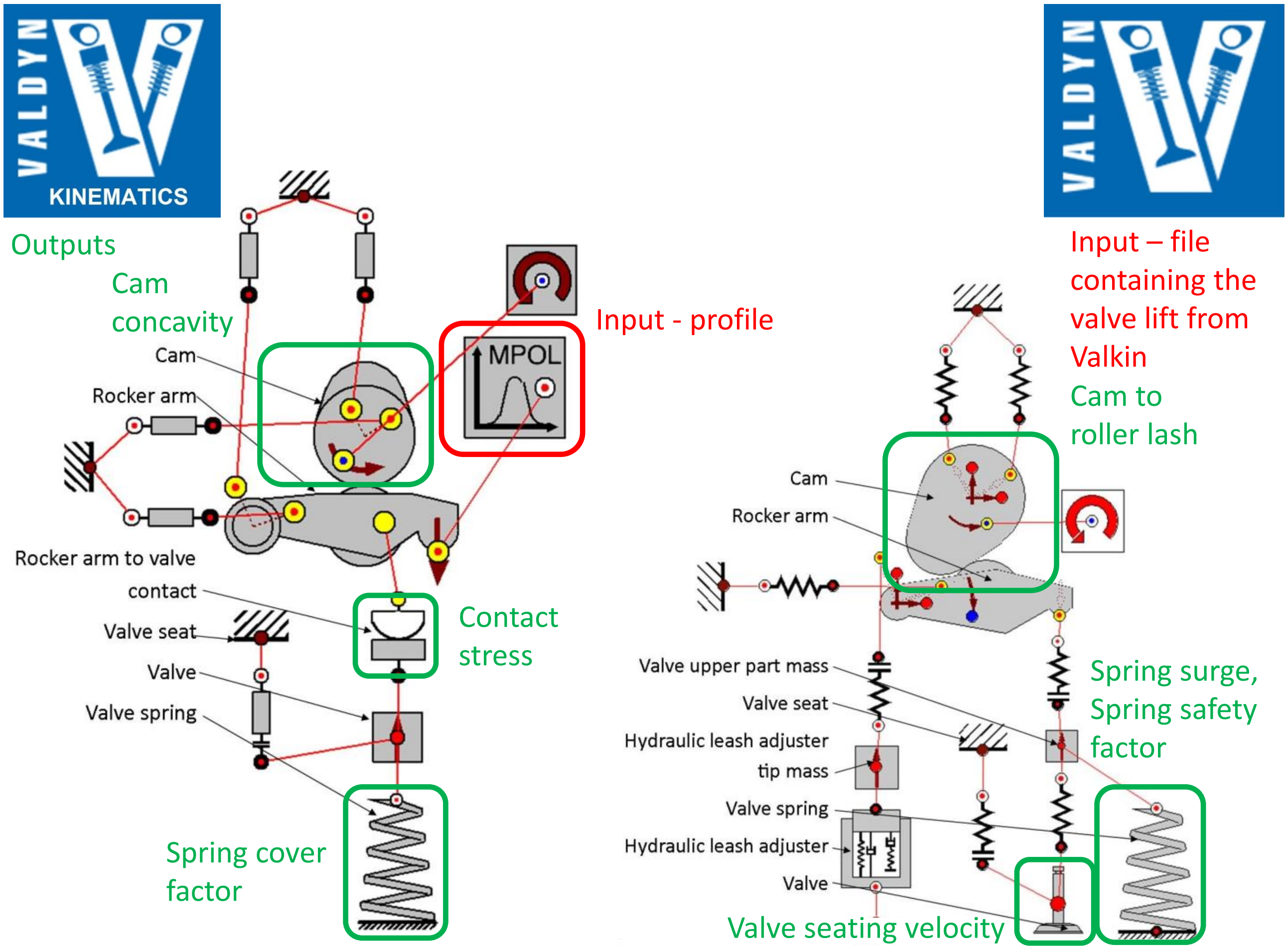


Figure 4.6: Valkin (left) and Valdyn (right) model input and output blocks.

The Valdyn inputs and outputs can be seen in figure 4.7. This is how the block is used in Isight model.

Note that all the outputs from Valdyn are vectors containing the critical values for every simulated rotational speed (loadcase). In this model, 22 loadcases were set. In order to obtain the critical values over all working conditions of the engine, an excel file was used. If this optimization tool is used for different model, the sumplot output and Isight data exchange would have to be adjusted.

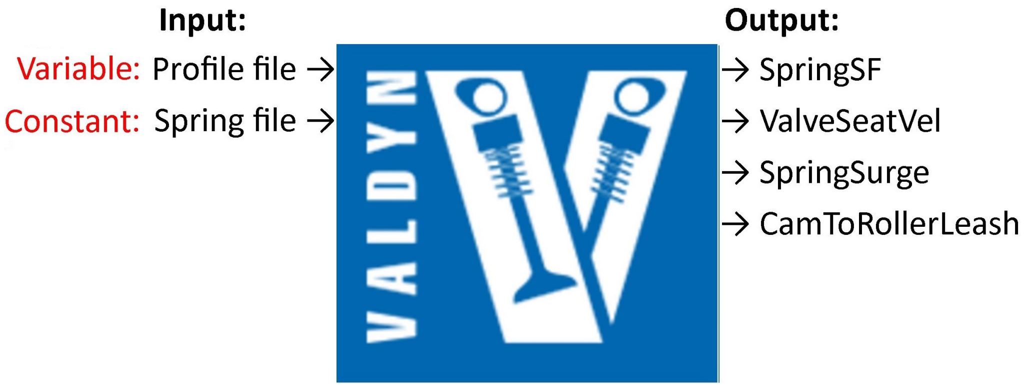


Figure 4.7: Valdyn inputs and outputs.

4.2.4 Excel Data Evaluation

The purpose of this simple excel file is to pick up the most critical values from Valdyn results. As the input, there are vectors of spring safety factor, valve seating velocity, spring surge and cam to roller lash. As output, it returns the minimum spring safety factor, the maximum valve seating velocity, the maximum spring surge and the maximum cam to roller lash.

If this optimization tool was used for a different model with different number of loadcases, this excel sheet and its inputs would have to be redesigned.

4.2.5 Objectives and Constrains Summarization

As the objective function, the valve integral was chosen to be maximized. Maximizing this value allows good aspiration of the engine. The optimization parameters including their limits are listed in table 4.1. The constraints are listed in the following table 4.5. The limiting values for every constrain can be found in the previous section.

Table 4.5: Summarization of optimization constraints.

Constrain	Block output
Maximum valve jerk	Matlab
Af/An	Matlab
Jerk Ok	Matlab
CC	Matlab
Cam concavity	Valkin
Contact stress	Valkin
Spring cover factor	Valkin
Spring safety factor	Valdyn
Valve seating velocity	Valdyn
Spring surge	Valdyn
Cam to roller lash	Valdyn

4.3 Optimization Process

In this section the optimization process will be described. The first stage is done outside Isight in Matlab. The optimization layout in Isight can be seen in figure 4.8. The reference components were used because the same configuration of some blocks was reused. This should make editing of the whole model easier. The process is as follows:

- Kinematic DOE is executed, after the file containing the DOE matrix is available.
- Dynamic DOE is executed with several feasible results of kinematic DOE.
- Several best feasible designs should be improved by optimization.
- The optimization results are evaluated against the design constraints in the last DOE.

These parts are to be explained in more detail further bellow.

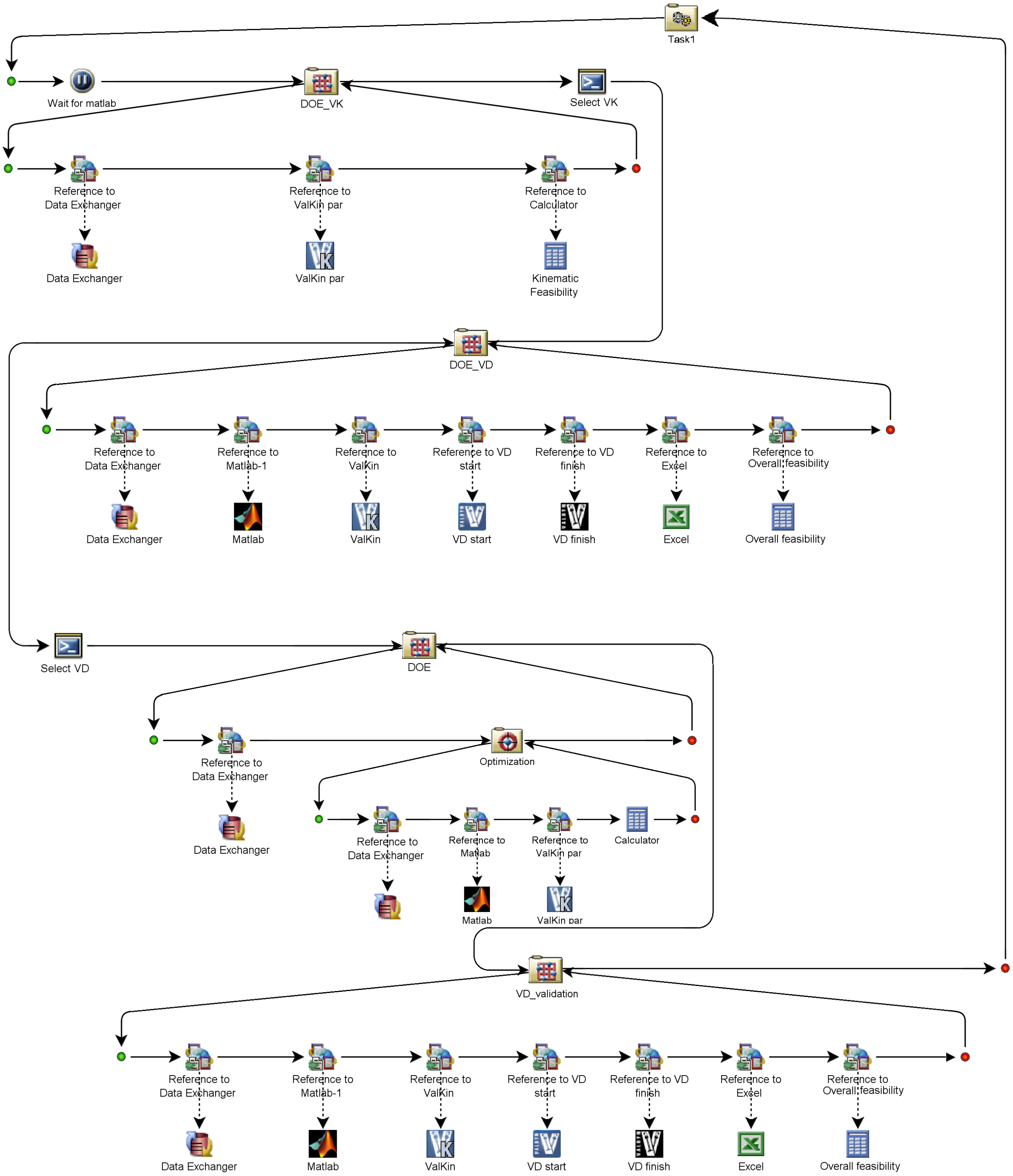


Figure 4.8: Isight optimization model.

4.3.1 Design of Experiment

The design of experiment was chosen as the first stage of optimization based on the computational cost of single blocks. Evaluating Matlab script takes fractions of seconds. Computing

the Valkin model takes just a few seconds and performing the Valdyn model computation takes slightly over one minute.
DOE was used here to generate a big number of input sets, downselect from these, and then perform the Matlab solution. After considering the limits, some input sets can be moved to the next round of solving kinematics in Valkin. After considering the limits again, just a fraction of original generated input sets will undergo the Valdyn solution. After this process is ready, some feasible starting conditions can be given as an input of optimization itself.

Matlab DOE

Because the number of experiments Isight can generate in DOE component is limited to be 100 000 at maximum, the first stage of DOE was done outside Isight in Matlab. Two methods for assembling DOE matrix were used. Full factorial DOE matrix was generated by the Matlab function `fullfact` and Latin hypercube matrix was generated by the Matlab function `lhsdesign`. Both the sets of resulting matrices were adjusted to fit the limits of the design variables. The product of this phase is a DOE matrix where every row of the matrix represents single experiment and the design variables for this experiment are arranged in columns.
In the DOE matrix, the values of variables cover the design space. Combinations giving negative θ_6 [2] either for opening or closing sequence are meaningless and therefore can be eliminated immediately. Parameters b, d, e, θ_1, θ_2, θ_3, θ_5 of opening sequence are compared against parameters of closing sequence. If big difference is found the sample is omitted. This is done because significant asymmetry would only lead to unacceptable value of closing coefficient CC. By this procedure a DOE matrix that is ready to proceed to Matlab `CamPol` script is constructed. Then, the Matlab `CamPol` script from subsection 4.2.1 is executed for all the experiments and the limits for output parameters are checked. The feasible designs are stored in a text file that can be used as Isight input.

Initially, the Latin hypercube and full factorial DOE matrices were compared. In table 4.6, the performance can be seen. The computations were performed using desktop computer equipped with 18 Gb RAM and CPU W3550 3.07 GHz. The full factorial matrix was generated having three levels per design variable. This determines the number of designs.[3] The number of designs for Latin hypercube was set to be equal to the number of designs of full factorial DOE in order to allow the comparison. It can be seen that the Latin hypercube matrix was more successful according to the number of experiments that went through the process. The Latin hypercube DOE matrix is more successful in single steps. The difference in computation time was predetermined by the initial selection. In the computational environment available for

[2] See subsection 3.5.3.

[3] The number of designs for full factorial matrix is $(levels)^{NumberOfDesignVariables}$ in this case $3^{15} = 14\,348\,907$

this study, memory limitations forced only three levels thus, large increments in the variables are not acceptable. This is why the Latin hypercube matrix was chosen to be used for this optimization task. The process of creating this DOE matrix by matlab can be done by script `CamPolDOE`.

In order to shorten the computation time, the kinematic solution should only be computed for the experiments that are close to the expected value of valve integral. This should be set in the `CanPolDOE` m-file. For this case, it was set to be 328.[4] If one was wrong about this value, it could be changed easily and the last part of the script can be evaluated again within several seconds.

Table 4.6: Comparison of DOE matrices. The number in bracket says the percentage that was preserved from the previous step.

DOE type	DOE initial size	After initial selection	After Matlab limits check	Matlab computation time
Full factorial	14 348 907	509 751 (3.5%)	8 366 (1.6%)	4.8 hours
Latin hypercube	14 348 907	1 652 749 (11.5%)	66 543 (4.0%)	15.8 hours

Isight DOE Valkin

In the Isight model, firstly, the kinematic DOE loop is performed. The wait block was included before the kinematic DOE loop in order to allow start of the loop directly after the DOE matrix is available from Matlab. This loop can be seen in more detail in figure 4.9. The purpose of this loop is to compute the kinematic solution of Valkin for the designs coming from Matlab as feasible ones. Firstly, the limits are loaded from a text file by data exchanger. This is done in order to allow easy change of the limiting values. Secondly, the kinematic solution is carried out. Lastly, the limits are considered and the designs are marked as feasible if they meet the demands. After this is done for all the experiments in the DOE matrix, the file containing the output values is stored. This file is then edited by the component Select_VK. The feasible designs are arranged according to the valve area integral and the best 30 experiments are stored in the text file. This file can be accessed by the next block of dynamic DOE. Note that the DOE component here is only used as a cyclic block which executes the predefined designs and stores the results. The option of computing in parallel was used here because it significantly reduces the computational time.

[4] This number can be estimated as $ValveOpenPeriod \cdot MaximumLift/2$. This is the area under a triangle profile. One would like to be well above this number.

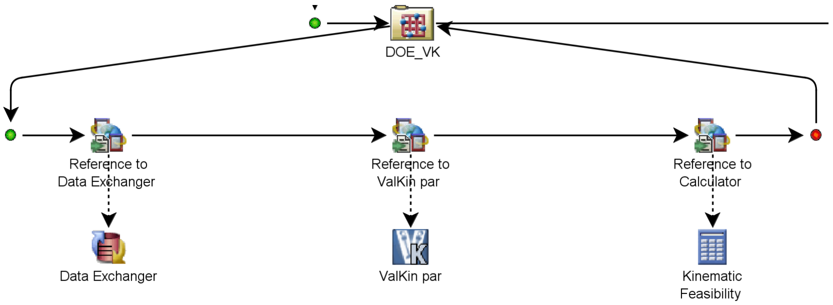


Figure 4.9: DOE kinematic loop.

Isight DOE Valdyn

The next loop in Isight model is dynamic loop. It can be seen in more detail in figure 4.10. It follows that the Valdyn block is executed in two sub-blocks. This was done in order to ensure that the output files of Valdyn are produced by combining the partial results from distributed runs. The execution directory was chosen to be the same for all the components. Then, Valdyn can easily access the files produced by Valkin. All the necessary parameters are linked from the DOE component to the blocks, the blocks are linked together and the output limits are checked in the calculator. The output of the calculator is linked to the DOE component again to store the results. The same configuration is used for validation of optimized designs in the last loop. Finally, the starting points for final optimization are chosen by selecting the best 10 points coming from the dynamic DOE using the Select_VD block. If dynamic optimization is chosen, then only one starting point should be used if reasonable computational times are required.

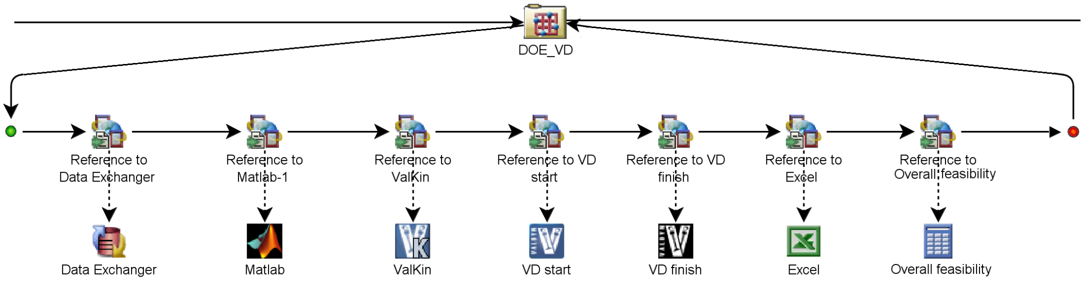


Figure 4.10: DOE dynamic loop.

4.3.2 Optimization

Optimization is the final step of this process after the feasible points from DOE are available. Two approaches to the optimization loop were developed:

- full dynamic optimization - the whole solution (Matlab, Valkin, Valdyn) is computed for every iteration of optimization. This ensures that all the monitored quantities are evaluated in every optimization iteration. This approach can be seen in figure 4.11.

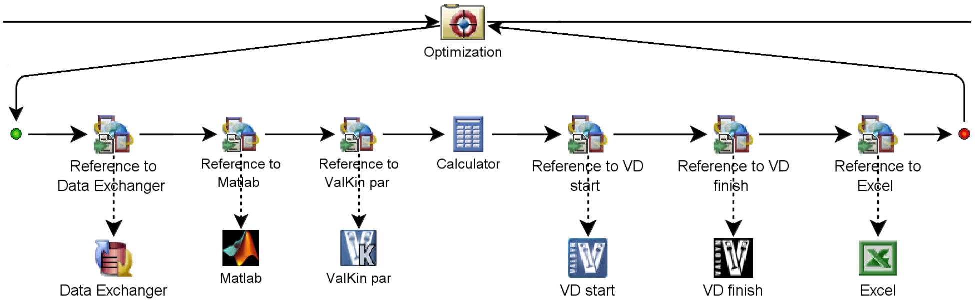


Figure 4.11: Dynamic optimization loop.

- kinematic optimization with dynamic validation - only Matlab `CamPol` and Valkin are built under the optimization component. This results in significantly shorter computation time. The trade off of this reduction in compute time is the need to run Valdyn solution as validation. Starting points should be chosen because some of them can appear to be unfeasible (from dynamic point of view) after optimization. This approach can be seen in figure 4.12.

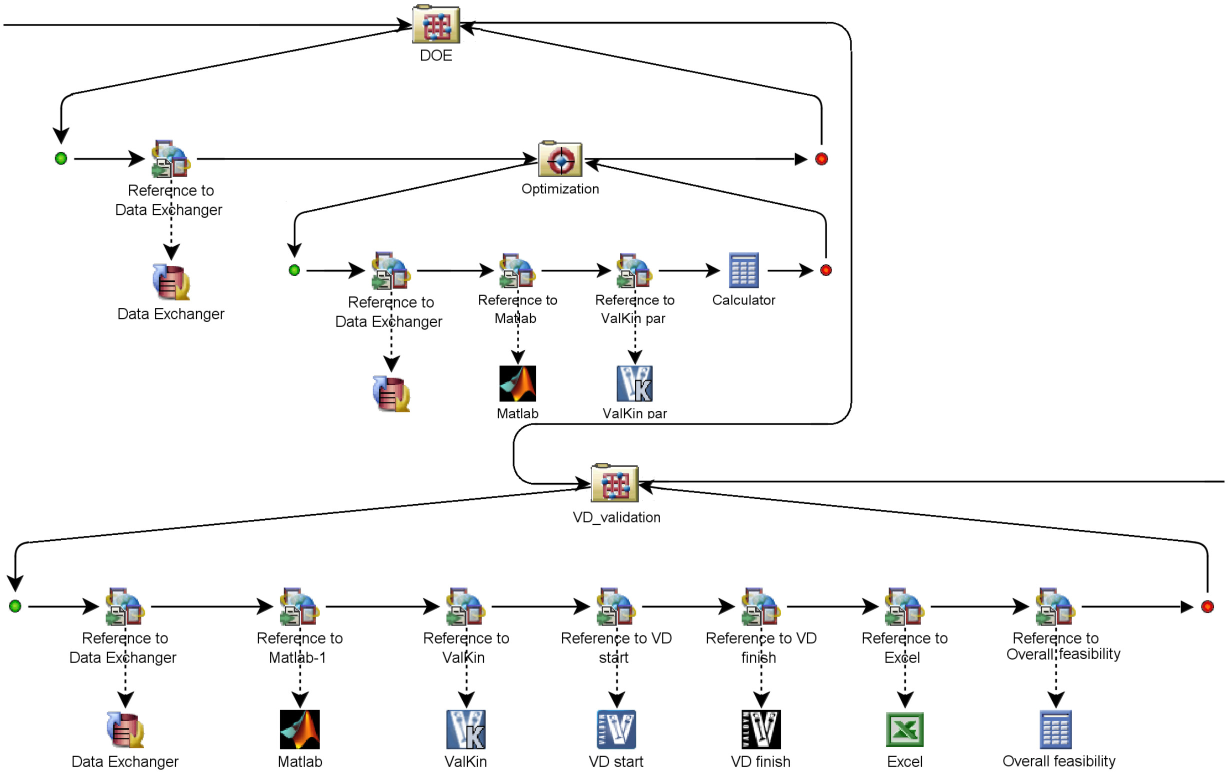


Figure 4.12: Kinematic optimization loop with dynamic validation.

If the dynamic outputs exceeds their limits and the kinematic limits are still within the bounds, then the first option of full dynamic optimization should be used. This is dependent on the valvetrain design being evaluated. Otherwise the second option of kinematic optimization with dynamic validation offers the benefit of less computational effort and possibly better results. The computation time of 300 iterations using dynamic model was around 6 hours. Doing the same optimization only with Matlab and Valkin using parallel computation and 10 starting points took about one and half hour. Both variants were tested and the second one was chosen since it yields better results. The number of 300 iterations was kept for this method due to possible exceed of dynamic limits if more iterations used. Some modification of this process for validating the best solution after certain number of iterations could be beneficial.

Optimization Techniques

Several optimization methods were compared in order to choose the best technique for this problem. The comparison can be seen in figure 4.13. The methods from section 3.4 perform as follows:

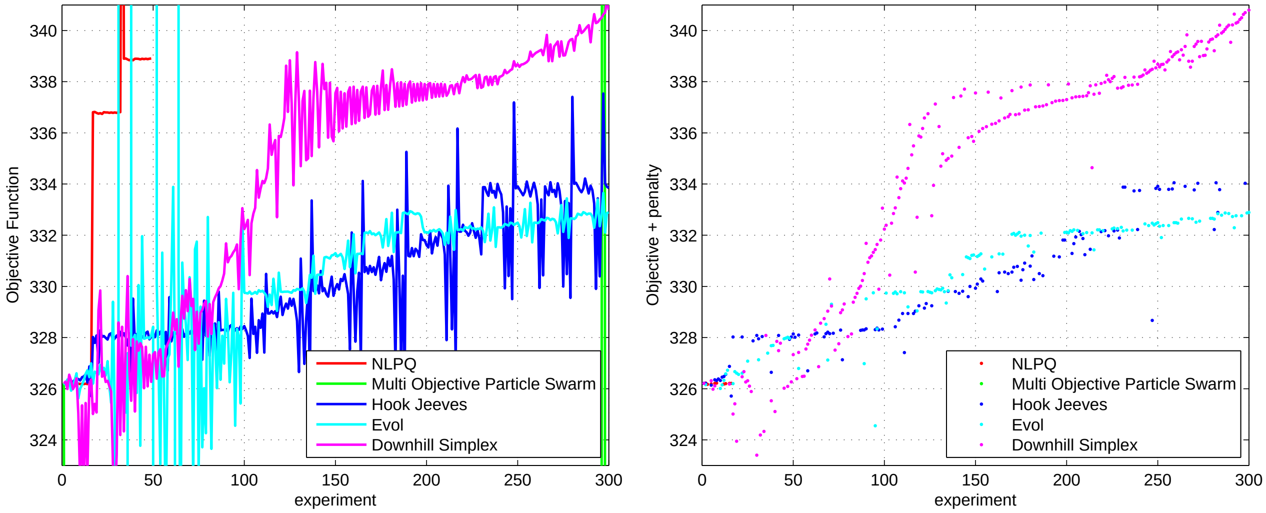
- Nonlinear quadratic programming reaches large values of objective function in a small amount of iterations, but for reasons that require further study it appears to fail when constraint boundary conditions are present. It is possible that the proper setting of this method was not found or the prerequisites (function properties) where not satisfied. This method was assumed to be unsuitable for this problem.

- Multi objective particle swarm method performed poorly due to its focus on global optimization across the design space thus requiring significantly more iterations to converge on an optimized solution.

- Hook Jeeves method was performing acceptably as it is deterministic approach derives a solution with acceptable constraints if some feasible starting point is available (as it was in this case).

- Evolutionary algorithm performed well despite being a global optimization method. This was likely due to the selection of the starting point. It can be seen that in the first half it was performing better than Hook Jeeves.

- Downhill simplex method was performing the best and thus was selected as the driver for the kinematic optimization in Isight model.



a) b)

Figure 4.13: Performance of optimization methods for cam profile optimization a) objective function, b) objective and penalty

4.4 Summary

The model from section 4.1 was computed with several input matrices from Matlab DOE. Then, the courses of optimization were compared.

The size of original matrices and the computation times can be seen in table 4.7. The computations were performed using desktop computer equipped with 18 Gb RAM and CPU W3550 3.07 GHz. It can be seen that both the computation time and the number of experiments that go through the process are proportional to the initial DOE size.

Table 4.7: Matlab DOE matrices.

DOE matrix	DOE initial size	After initial selection	After Matlab limits check	Computation time [hours]
lhs small	$7 \cdot 10^6$	806 691	28 872	8.5
lhs medium	$14 \cdot 10^6$	1 652 749	59 221	15.8
lhs large	$40 \cdot 10^6$	4 603 902	184 532	37

Then, the Isight model was executed for a set of best outputs from Matlab DOE and the key variables evaluated. Figures 4.14-4.18 show the development of objective function and key constrained output parameters through the optimization. The graphs contained below are derived from the second computational cycle that was accomplished using the medium DOE matrix (see Appendix for high resolution versions of these plots). Additionally, the figures showing the course of other runs with small DOE and large DOE can be found in the appendix as well.

The blue points represent the feasible experiments and the red points represent the experiments where at least one limit was exceeded. Additionally, the lower limit was displayed as a dashed green line and the upper limit was displayed as a dashed magenta line.

It can be seen that the objective function in figure 4.14 reached its maximum under the ninth starting configuration. This example clearly demonstrates the benefits of multiple starting points of the optimization model in searching for minimum or maximum. A design evaluation based only on the starting points prior to the point 9 would not have revealed this potentially significant maximum.

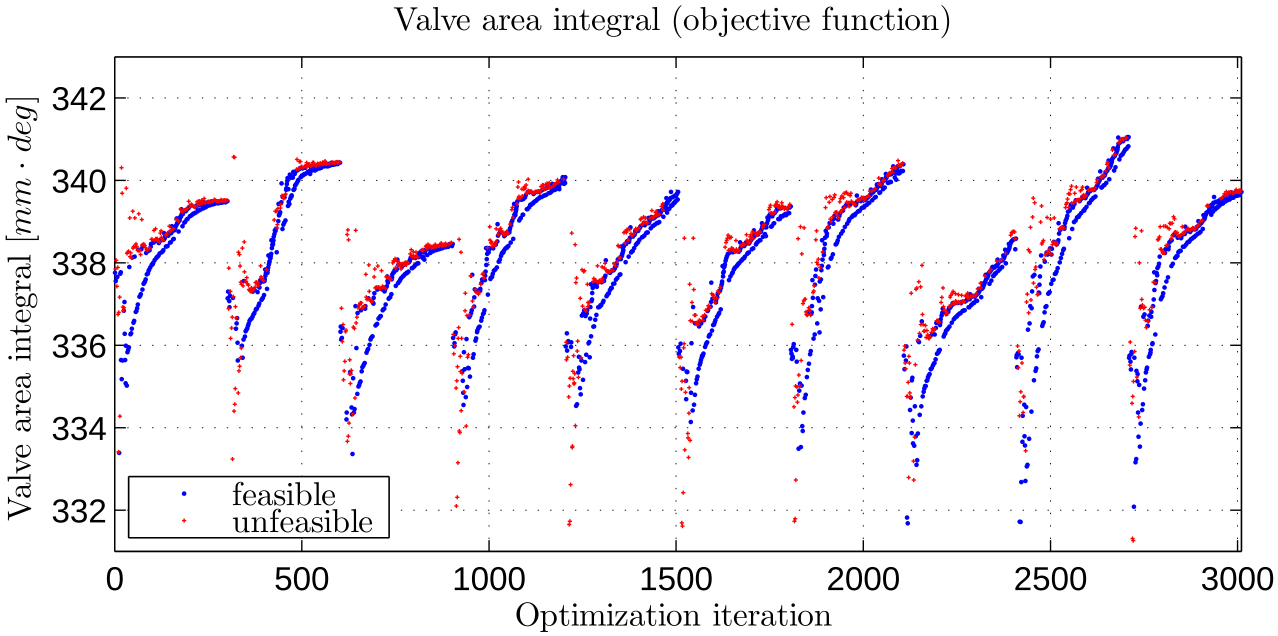


Figure 4.14: Course of objective function through the kinematic optimization process.

In figure 4.15 the course of coefficient CC can be seen. There are some points that exceeded both the upper and the lower bound. Generally, the optimal value seems to be in the lower half of the interval. The coefficient CC demonstrates effect associated with the steepness of acceleration curve in proximity to the zero line.

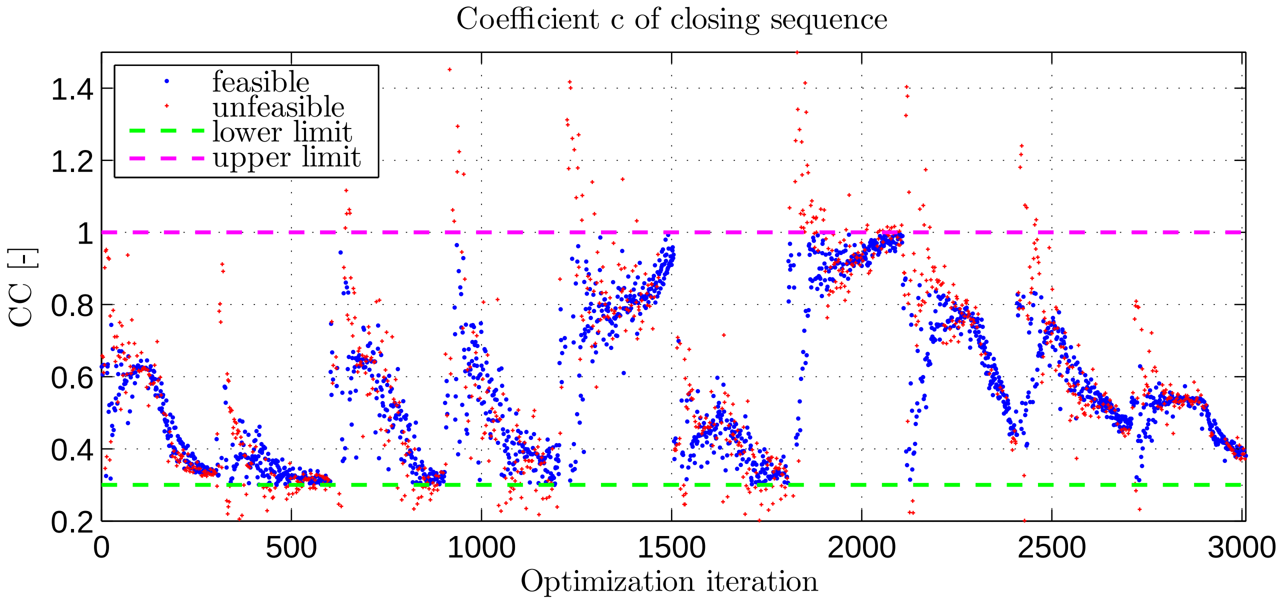


Figure 4.15: Course of coefficient c of closing sequence.

The course of maximum jerk can be seen in figure 4.16. It clearly demonstrates this was a significant limit of the objective function increase. If the course of the maximum jerk in figure 4.16 is compared to the course of the objective function in figure 4.14 it can be seen that after the maximum allowed jerk is arrived, the rise of objective function is not stopped thus there is still space for improvement.

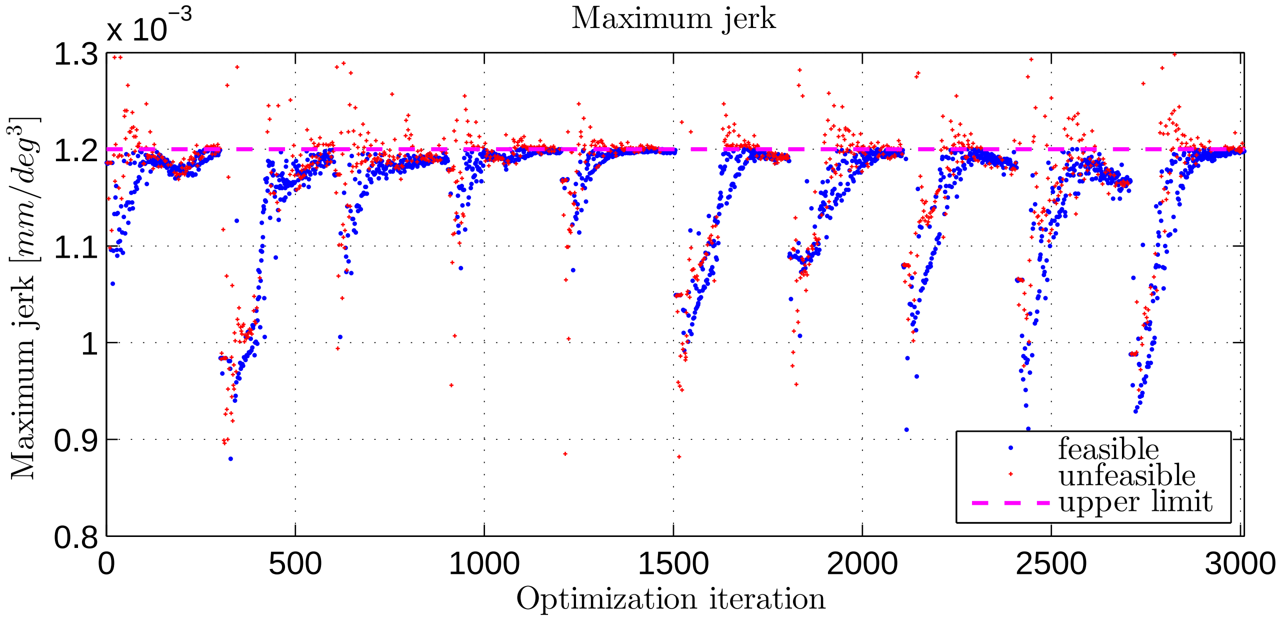


Figure 4.16: Course of maximum jerk.

The course of spring cover factor through the optimization process is shown in figure 4.17. It can be seen that only a few of the points are under the chosen limit. Majority of the experiments is fairly above the limit.

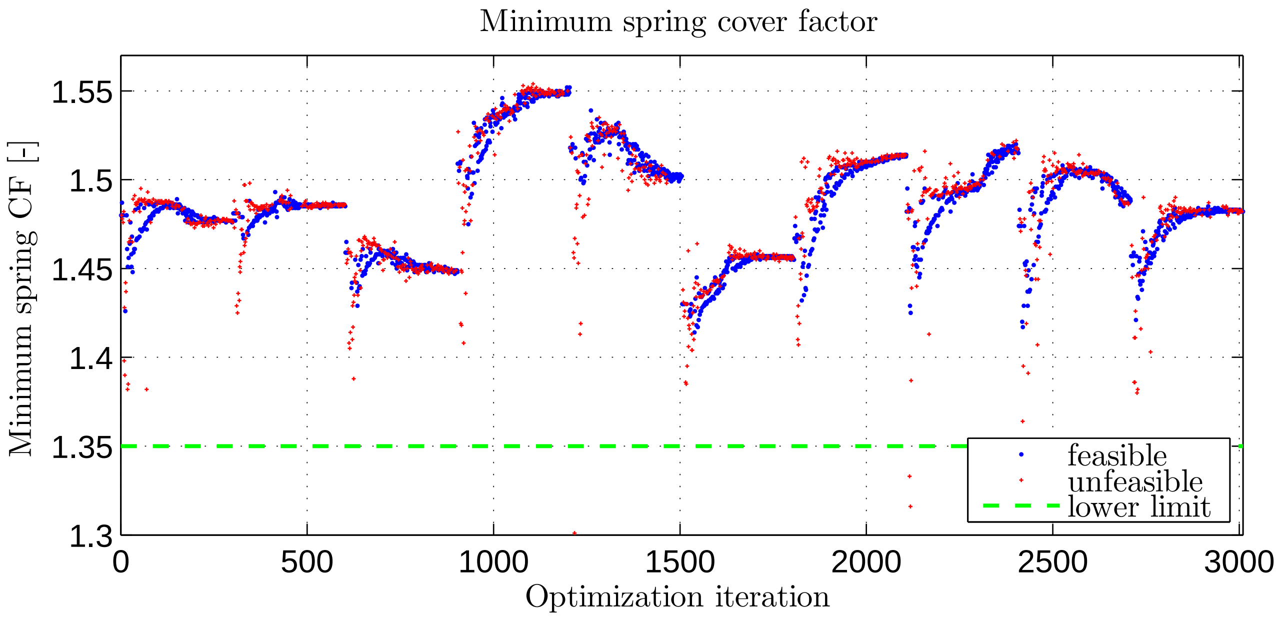


Figure 4.17: Course of minimum spring cover factor.

In figure 4.18, the cam concavity can be seen as it develops through the optimization process. It clearly demonstrates this was a significant limit of the objective function increase. If a valve profile with a bigger valve area integral is needed, the value of cam concavity limit can be adjusted. It can result in higher manufacturing cost.

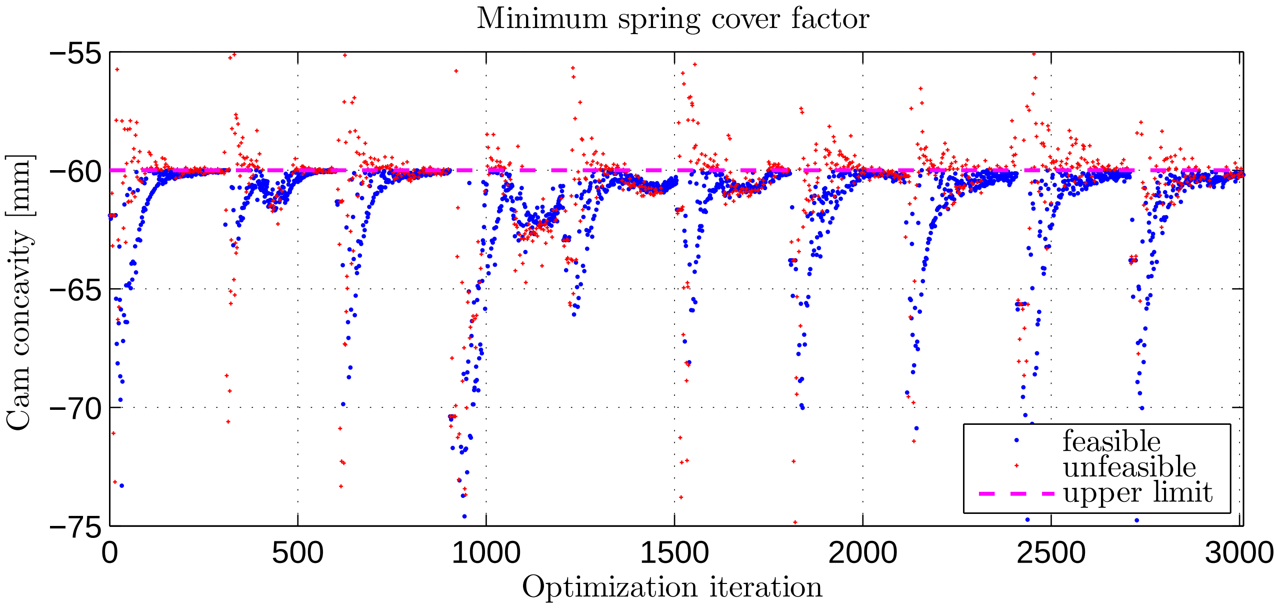


Figure 4.18: Course of cam concavity.

In table 4.8 the best runs from DOE and the best runs after optimization are summarized. It can be seen that the best initial condition from DOE was not the one with the highest objective function. Additionally, the comparison with the manually designed valve lift can be done. The manual design slightly exceeds the limit of maximum jerk. This is assumed to be acceptable since this limit is more a recommendation than a hard constrain according to [VALDYN, 2016]. Observing the first part of table 4.8, the design variable sets can be seen for nearly optimal designs. It clearly follows that the objective function features by number of local maxima.

Tables summarizing the starting configurations for optimization loop and the optimized configurations can be found in the appendix. The outputs are also included so the values can be checked and compared to the limits. It can be seen that the dynamic limits were satisfied. The valve seating velocity was closest to the limit. The remaining observed dynamic factors were fairly far from their limits.

Table 4.8: Summary of the best runs for small (S), medium (M), large (L) and optimized (-O) DOE compared to manual design (MD).

Run	BC	BO	CO	DC	DO	EC	EO	TH1C	TH1O	TH2C	TH2O	TH3C	TH3O	TH5C	TH5O
MD	0.58	0.76	0.80	1.00	1.02	0.32	0.31	12.40	5.00	2.00	2.70	15.00	20.31	26.00	35.00
S1	0.83	0.73	0.67	1.06	1.06	0.83	1.88	4.90	7.60	4.80	7.10	17.70	10.60	25.00	20.00
1O	0.86	0.73	0.67	1.04	1.06	0.96	1.86	4.86	7.43	4.57	7.15	17.55	10.55	25.41	21.07
S4	0.74	0.42	0.75	1.28	0.96	0.50	1.32	5.70	5.50	9.60	8.50	10.70	13.10	18.80	24.80
S4O	0.79	0.46	0.82	1.24	1.01	0.97	1.29	4.30	5.61	10.37	7.77	10.10	12.12	16.73	16.45
M1	0.72	0.44	0.91	0.96	1.03	2.49	1.39	9.8	5.6	6.1	4.6	10.9	16.6	25.8	21.6
M1O	0.72	0.44	0.96	0.97	1.02	2.35	1.35	9.37	5.72	5.27	4.70	10.01	16.17	26.87	23.12
M9	0.50	0.52	0.74	1.28	1.00	0.90	1.38	5.9	6.2	5.2	6.8	16.5	13.7	27.4	19.5
M9O	0.52	0.45	0.75	1.19	1.04	1.25	1.48	5.07	6.10	4.35	7.40	15.56	11.82	29.42	15.10
L1	0.58	0.44	0.40	1.27	1.06	0.44	1.31	6.1	5.8	9.5	8.2	10.6	11.1	20.2	15.2
L1O	0.57	0.43	0.40	1.20	1.05	0.87	1.29	5.97	5.44	9.17	8.44	10.04	10.90	20.07	15.61
L9	0.82	0.65	0.71	1.04	1.03	1.06	0.68	7.1	6.3	9.1	8.4	10.3	11.0	26.1	16.3
L9O	0.87	0.61	0.74	1.02	1.08	1.34	1.15	5.43	5.19	8.09	8.75	10.68	10.88	20.36	18.63

Run	CC	Max jerk	Af/An	JerkOk	Area integral	Cam concavity	Contact stress	Spring cover factor	Spring safety factor	Valve seating velocity	Spring surge	Cam to roller lash
Limits	$\langle 0.3; 1\rangle$ [-]	$\langle 0; 0.0012\rangle$ $[mm/deg^3]$	$\langle 0; 3\rangle$ [-]	Ok	max	$(-\infty; -60\rangle$ & $\langle 0; \infty)$ $[mm]$	$\langle 0; 1250\rangle$ $[MPa]$	$\langle 1.35; \infty)$ [-]	$\langle 1.1; \infty\rangle$ [-]	$\langle 0; 0.3\rangle$ $[m/s]$	$\langle 0; 1\rangle$ $[mm]$	$\langle 0; 0.05\rangle$ $[mm]$
MD	0.820	1.35E-03	2.12	-	329.96	-60.66	625.5	1.41	1.81	0.20	0.37	5.89E-07
S1	0.935	1.17E-03	2.24	1	338.28	-61.48	616.9	1.53	1.66	0.18	0.350332	-5.1E-08
1O	0.889	1.20E-03	2.28	1	339.35	-60.38	617.2	1.52	1.65	0.20	0.341614	-5.0E-08
S4	0.325	1.15E-03	2.14	1	335.68	-70.83	630.9	1.46	1.72	0.23	0.364674	-9.8E-09
S4O	0.458	1.20E-03	2.36	1	340.97	-60.08	631.3	1.50	1.61	0.26	0.380591	2.2E-05
M1	0.628	1.19E-03	2.24	1	337.76	-61.90	618.7	1.48	1.81	0.23	0.34	1.2E-02
M1O	0.332	1.20E-03	2.32	1	339.5	-60.05	625.8	1.48	1.73	0.25	0.32	7.60E-06
M9	0.804	1.07E-03	2.20	1	335.71	-65.65	623.5	1.47	1.69	0.20	0.36	1.1E-02
M9O	0.483	1.17E-03	2.35	1	341.05	-60.07	627.4	1.49	1.64	0.22	0.35	-3.09E-08
L1	0.687	1.12E-03	2.29	1	338.47	-60.94	634.5	1.47	1.67	0.32	0.39	7.2E-04
L1O	0.430	1.20E-03	2.33	1	340.03	-60.16	633.4	1.48	1.63	0.25	0.35	-3.01E-08
L9	0.590	1.03E-03	2.27	1	336.92	-61.15	624.8	1.51	1.57	0.26	0.41	1.5E-02
L9O	0.363	1.20E-03	2.28	1	341.93	-60.09	626.2	1.52	1.57	0.25	0.40	-4.20E-08

It can be seen that the limits were reached and there is not much more space of increasing the objective function with conventional cam. One way how to omit some of the limits would be to use an electric actuator for driving the valves. This would give more space for optimizing the profile. Additionally, specific profile for various engine working conditions can be used.

Chapter 5

Results

The resulting profiles, generated by each methodology are presented and discussed in this chapter. The best profiles from the three runs with various DOE matrices are compared with a manually designed valve lift curve. The valve lift curve can be seen in figure 5.1. What is readily apparent is the close fit between all the available methods. Thus a difference plot is used as shown in figure 5.2. Here, the manually designed valve lift profile was chosen as a baseline since it has the smallest value of valve area integral. It can be seen that the objective function copies the computational cost of the initial DOE matrix. The best objective function was achieved when large DOE matrix was used. The corresponding configurations can be found in table 4.8 in rows MD, S4O, M9O, L9O. The resulting valve area integral is also reported as nondimensional area integral that characterises the ratio of the valve area integral over the circumscribed rectangle area. The nondimensional area integral for the compared valve lift configurations can be seen in table 5.1.

Table 5.1: Comparison of nondimensional area integral for the optimized profiles and manual design.

Design	MD	S4O	M9O	L9O
nondimensional area integral	53.9%	55.7 %	55.7	55.9%

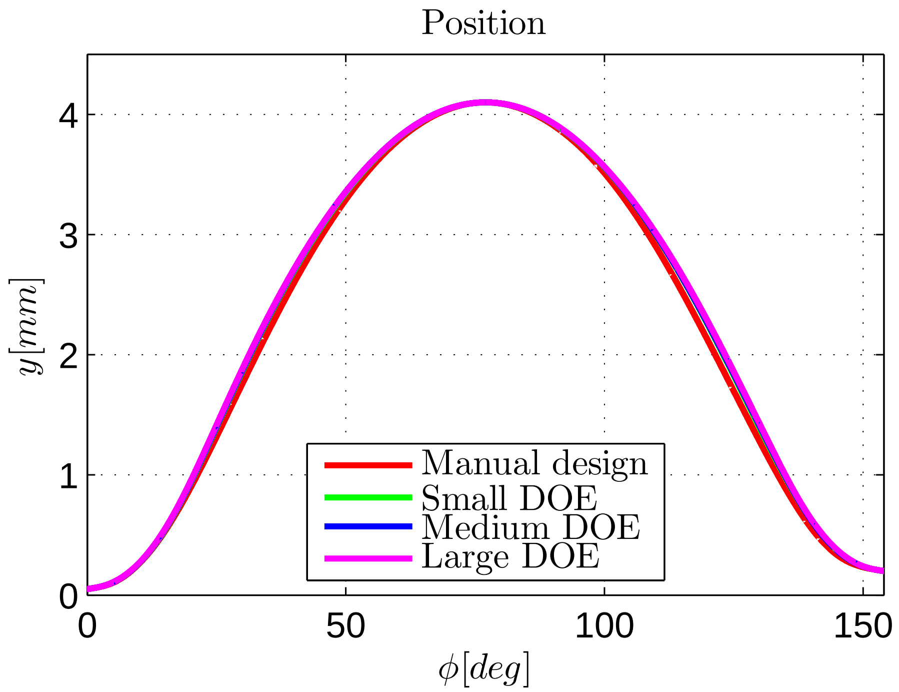


Figure 5.1: Valve lift course.

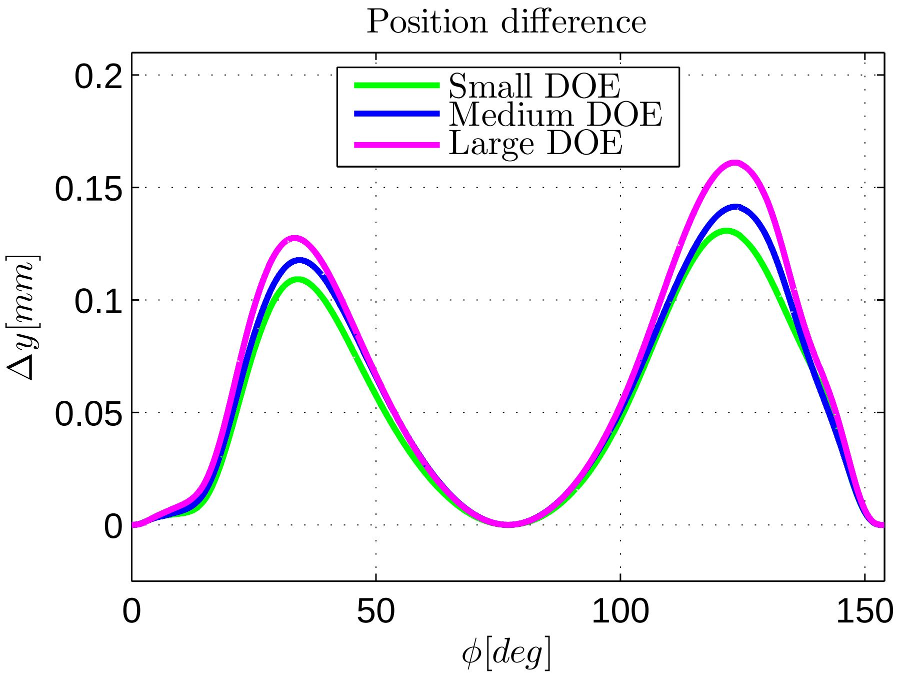


Figure 5.2: Valve lift difference. The manual design was chosen as a baseline.

The velocity profile can be seen in figure 5.3. Here, it can be clearly seen that the velocity profile for optimized valve lift curve achieves higher maximum values. Additionally, the acceleration in figure 5.4 is steeper for the optimized profiles. In the plot of acceleration, the differences are more obvious than in the plots of lower derivatives.

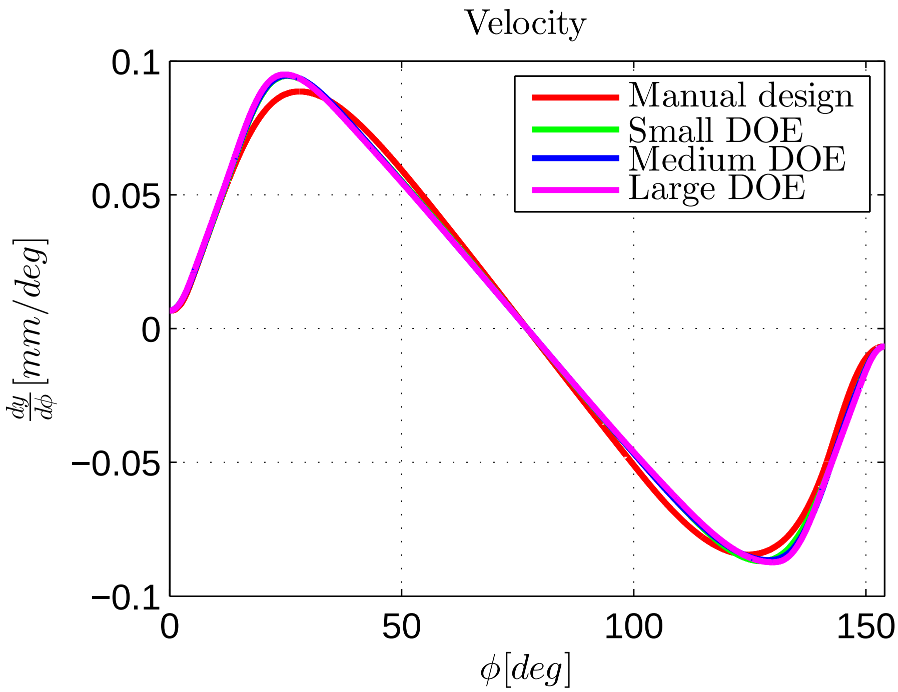


Figure 5.3: Valve velocity course.

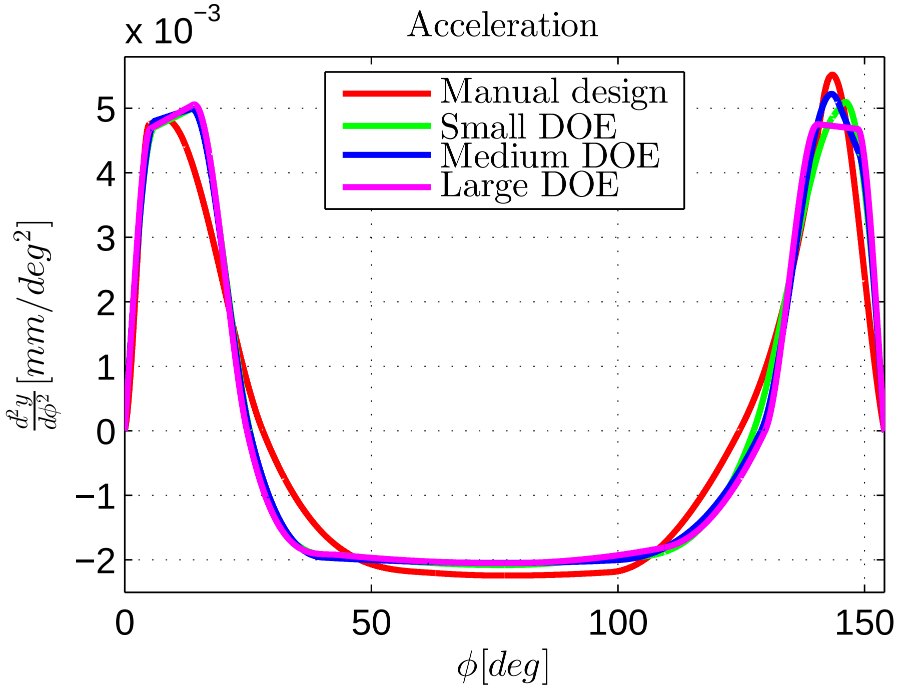


Figure 5.4: Valve acceleration course.

Chapter 6

Conclusion

Throughout this work, the fundamentals of valvetrain mechanisms, optimization techniques and software tools are introduced. The kinematic and dynamic models of valvetrain in Valkin and Valdyn are described including their input and output parameters. Then, the optimization process is developed in Isight.

The results show that the valve integral can achieve larger values under given limits when using optimization tools. One can obtain several feasible solutions by running an overnight computation that offers a reduction in design time over traditional manual methods. Three initial DOE matrices differing by their size were used and the expected results were observed. If the large matrix was used, the biggest objective function was obtained. According to this setting, trade-offs can be done with respect to the time available for the computation by sizing the initial DOE matrix and the allowed number of iterations for the optimization component.

However, optimal setting might differ engine to engine. Hence, some adjustments to the model must be done when used for optimizing different valvetrain system. Moreover, this tool uses parallel computation so it uses the whole CPU. This means that other computationally expensive jobs are not recommended to be executed on the same machine while optimizing the valve lift.
By this, the objectives set out in the second chapter were met.

Additionally, this research highlighted a number of areas that may require further investigation:

The failure of nonlinear quadratic programming method for this problem discussed in section 4.3.2 should be investigated further in order to verify whether the root cause is in the model or whether it is associated with the modeling tool or method implementation.

It is also suggested there is merit in exploring additional modeling methods such as spline that offer the prospect of reducing optimization parameters, hence offering the potential for additional decreases in computational time.

Bibliography

[Cavazzuti, 2013] Cavazzuti, M. (2013). *Optimization Methods: From Theory to Design.* Springer.

[Kirgat and Surde, 2014] Kirgat, G. and Surde, A. (2014). Review of Hooke and Jeeves Direct Search Solution Method Analysis Applicable To Mechanical Design Engineering. *Novateur publications international journal of innovations in engineering research and technology*, 1: p. 1 – 14.

[Lederer and Brousil, 1989] Lederer, P. and Brousil, J. (1989). *Teorie a optimalizace mechanických systémů I: Určeno pro stud. fak. strojní.* ČVUT.

[McCullock, 1999] McCullock, J. (1999). Particle swarm optimization. `http://mnemstudio.org/particle-swarm-introduction.htm`. Online; accessed 8 April 2017.

[Mikuláš, 2013] Mikuláš, O. (2013). *Quadratic Programming Algorithms for Fast Model-Based Predictive Control.* PhD thesis, Czech Technical University in Prague.

[Norton, 2009] Norton, R. L. (2009). *Cam Design and Manufacturing Handbook.* Industrial Press.

[VALDYN, 2016] VALDYN (May 2016). Valdyn documentation 2016.1.

[Viana, 2015] Viana, F. A. C. (2015). A Tutorial on Latin Hypercube Design of Experiments. *Quality and Reliability Engineering International*, 32.

Appendix A

Tables and Plots of Optimization Results

Table A.1: Small DOE results to be optimized.

Run	BC	BO	CO	DC	DO	EC	EO	TH1C	TH1O	TH2C	TH2O	TH3C	TH3O	TH5C	TH5O	CC	Max jerk	Af/An	JerkOk	Area integral	Cam concavity	Contact stress	Spring cover factor	Spring safety factor	Valve seating velocity	Spring surge	Cam to roller lash
Limits																$\langle 0.3; 1\rangle$ [-]	$\langle 0; 0.0012\rangle$ $[mm/deg^3]$	$\langle 0; 3\rangle$ [-]	Ok	max	$(-\infty; -60\rangle$ & $\langle 0; \infty)$ $[mm]$	$\langle 0; 1250\rangle$ $[Mpa]$	$\langle 1.35; \infty)$ [-]	$\langle 1.1; \infty\rangle$ [-]	$\langle 0; 0.3\rangle$ $[m/s]$	$\langle 0; 1\rangle$ $[mm]$	$\langle 0; 0.05\rangle$ $[mm]$
1	0.83	0.73	0.67	1.06	1.06	0.83	1.88	4.90	7.60	4.80	7.10	17.70	10.60	25.00	20.00	0.935	1.17E-03	2.24	1	338.28	-61.48	616.9	1.53	1.66	0.18	0.350332	-5.1E-08
2	0.85	0.56	0.45	1.02	1.10	1.09	2.29	6.40	9.00	2.80	6.20	17.00	10.40	28.70	31.90	0.490	1.17E-03	2.14	1	336.69	-61.44	619.9	1.47	1.64	0.24	0.315758	1.9E-06
3	0.87	0.60	0.72	1.15	0.98	1.27	1.53	5.40	6.90	3.20	7.60	18.60	12.30	26.80	25.20	0.827	1.03E-03	2.16	1	336.23	-67.16	618.5	1.47	1.78	0.20	0.337647	8.6E-06
4	0.74	0.42	0.75	1.28	0.96	0.50	1.32	5.70	5.50	9.60	8.50	10.70	13.10	18.80	24.80	0.325	1.15E-03	2.14	1	335.68	-70.83	630.9	1.46	1.72	0.23	0.364674	-9.8E-09
5	0.93	0.54	0.64	1.14	1.14	2.02	2.40	9.00	9.40	6.00	6.00	10.50	10.90	29.60	22.20	0.417	1.13E-03	2.07	1	335.40	-60.80	632.0	1.45	1.64	0.24	0.403535	-2.7E-08
6	0.49	0.52	0.39	1.21	0.96	1.46	1.44	4.40	5.80	8.80	9.00	14.40	10.60	32.80	33.50	0.585	1.18E-03	2.16	1	335.33	-65.29	616.2	1.48	1.78	0.26	0.306027	1.5E-06
7	0.76	0.46	0.82	1.13	1.00	0.69	1.49	5.50	6.90	2.80	5.70	17.80	15.20	30.50	21.40	0.437	1.01E-03	2.10	1	335.19	-63.52	622.3	1.44	1.78	0.24	0.30215	1.6E-05
8	0.87	0.42	0.73	1.21	1.21	1.52	1.47	8.20	5.20	6.80	5.80	10.60	16.10	28.90	24.10	0.376	1.13E-03	1.84	1	334.60	-61.06	633.6	1.45	1.80	0.27	0.359434	8.5E-07
9	0.56	0.46	0.39	1.09	1.06	2.35	1.12	10.00	5.90	4.80	7.20	13.50	12.70	19.80	23.00	0.818	1.04E-03	2.15	1	334.52	-63.55	620.7	1.46	1.78	0.18	0.356025	3.0E-07
10	1.00	0.45	0.87	1.25	1.21	1.03	1.13	4.40	4.10	3.00	7.40	19.10	16.70	24.90	23.10	0.481	1.18E-03	1.69	1	334.30	-67.22	620.6	1.43	1.77	0.21	0.298103	-3.1E-08

Table A.2: Small DOE after optimization.

Run	BC	BO	CO	DC	DO	EC	EO	TH1C	TH1O	TH2C	TH2O	TH3C	TH3O	TH5C	TH5O	CC	Max jerk	Af/An	JerkOk	Area integral	Cam concavity	Contact stress	Spring cover factor	Spring safety factor	Valve seating velocity	Spring surge	Cam to roller lash
Limits																$\langle 0.3; 1\rangle$ [-]	$\langle 0; 0.0012\rangle$ $[mm/deg^3]$	$\langle 0; 3\rangle$ [-]	Ok	max	$(-\infty; -60\rangle$ & $\langle 0; \infty)$ $[mm]$	$\langle 0; 1250\rangle$ $[Mpa]$	$\langle 1.35; \infty)$ [-]	$\langle 1.1; \infty\rangle$ [-]	$\langle 0; 0.3\rangle$ $[m/s]$	$\langle 0; 1\rangle$ $[mm]$	$\langle 0; 0.05\rangle$ $[mm]$
1	0.86	0.73	0.67	1.04	1.06	0.96	1.86	4.86	7.43	4.57	7.15	17.55	10.55	25.41	21.07	0.889	1.20E-03	2.28	1	339.35	-60.38	617.2	1.52	1.65	0.20	0.341614	-5.0E-08
2	0.88	0.57	0.45	1.02	1.10	1.37	2.28	5.89	8.97	2.62	6.16	16.37	10.33	29.34	31.98	0.302	1.20E-03	2.17	1	338.06	-60.03	622.5	1.47	1.66	0.25	0.28645	3.2E-06
3	0.95	0.65	0.72	1.12	0.98	1.40	1.55	5.17	6.30	2.21	7.19	18.01	12.34	28.21	24.66	0.639	1.20E-03	2.31	1	340.07	-60.21	624.0	1.50	1.63	0.20	0.337384	-4.3E-08
4	0.79	0.46	0.82	1.24	1.01	0.97	1.29	4.30	5.61	10.37	7.77	10.10	12.12	16.73	16.45	0.458	1.20E-03	2.36	1	340.97	-60.08	631.3	1.50	1.61	0.26	0.380591	2.2E-05
5	1.00	0.56	0.66	1.14	1.12	2.09	2.34	8.57	8.84	4.95	6.22	10.58	10.93	28.75	22.29	0.334	1.19E-03	2.13	1	337.58	-60.19	635.3	1.46	1.58	0.21	0.374086	-3.6E-08
6	0.49	0.53	0.48	1.21	0.97	1.44	1.41	4.43	5.95	8.51	8.69	14.11	10.52	31.19	32.59	0.658	1.20E-03	2.31	1	338.17	-60.04	619.5	1.50	1.70	0.26	0.328939	-5.9E-08
7	0.77	0.48	0.86	1.05	1.03	0.85	1.66	5.15	6.66	2.78	5.85	18.05	13.43	31.80	20.82	0.676	1.20E-03	2.33	1	340.38	-60.20	620.8	1.48	1.74	0.23	0.312558	1.1E-05
8	0.97	0.46	0.96	1.07	1.20	1.63	1.43	6.25	4.82	6.70	5.92	11.55	16.11	29.75	23.74	0.380	1.20E-03	1.92	1	338.64	-60.33	626.9	1.46	1.72	0.23	0.321922	1.1E-05
9	0.73	0.40	0.54	1.09	1.03	2.07	1.58	8.86	6.46	4.00	6.94	13.40	12.13	20.11	15.50	0.840	1.18E-03	2.34	1	339.32	-60.12	627.8	1.47	1.70	0.19	0.362215	1.4E-05
10	0.97	0.43	0.90	1.27	1.23	1.22	1.11	4.06	4.22	3.53	6.41	17.84	16.22	28.02	16.81	0.649	1.20E-03	1.87	1	338.78	-60.13	624.5	1.45	1.73	0.21	0.319381	1.4E-05

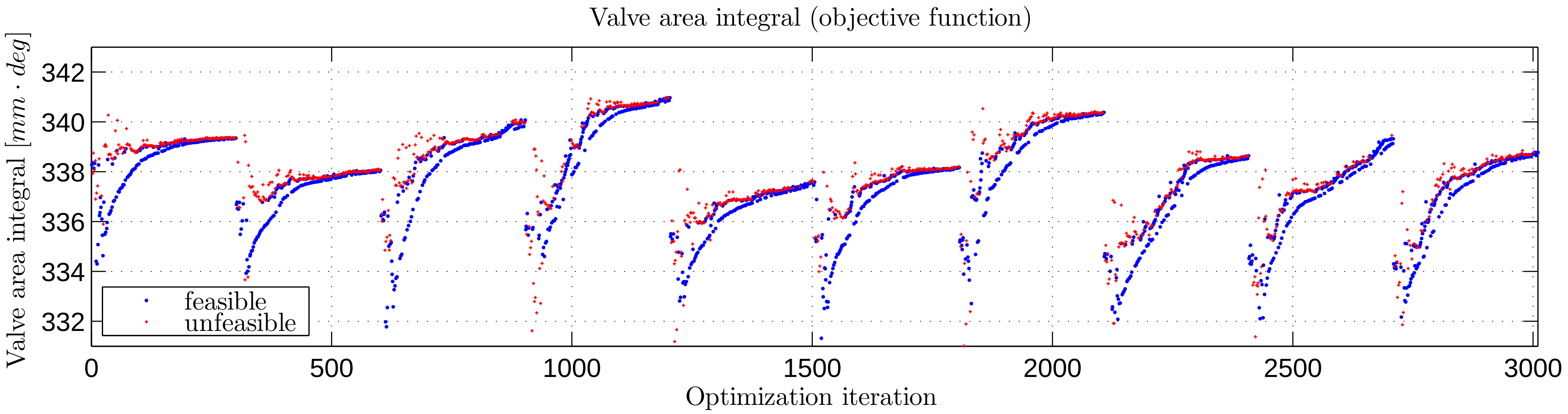


Figure A.1: Course of objective function through the optimization process. Small DOE matrix.

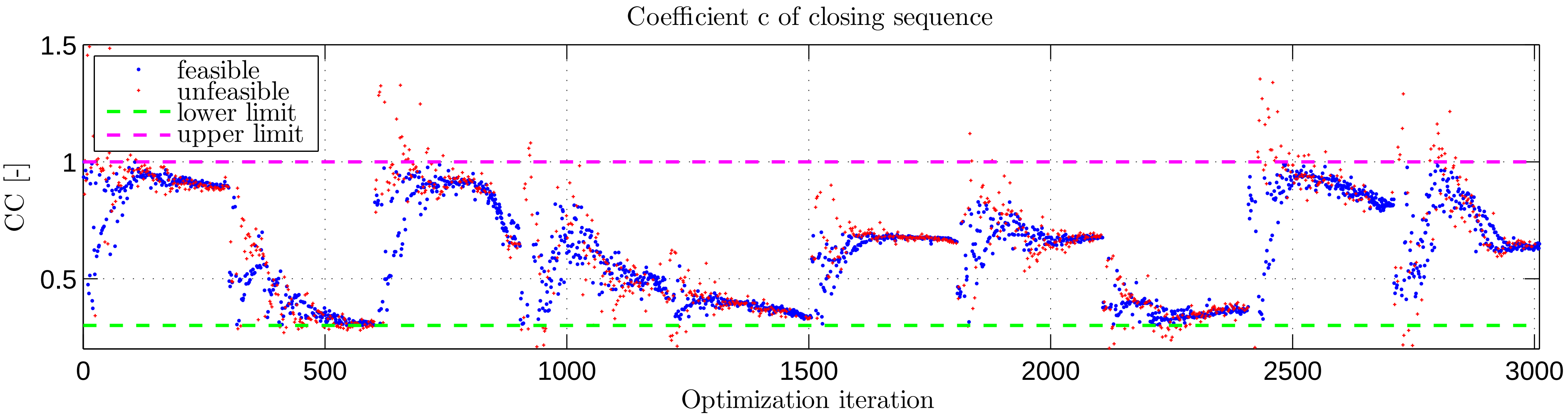


Figure A.2: Course of coefficient c of the closing sequence through the optimization. Small DOE matrix.

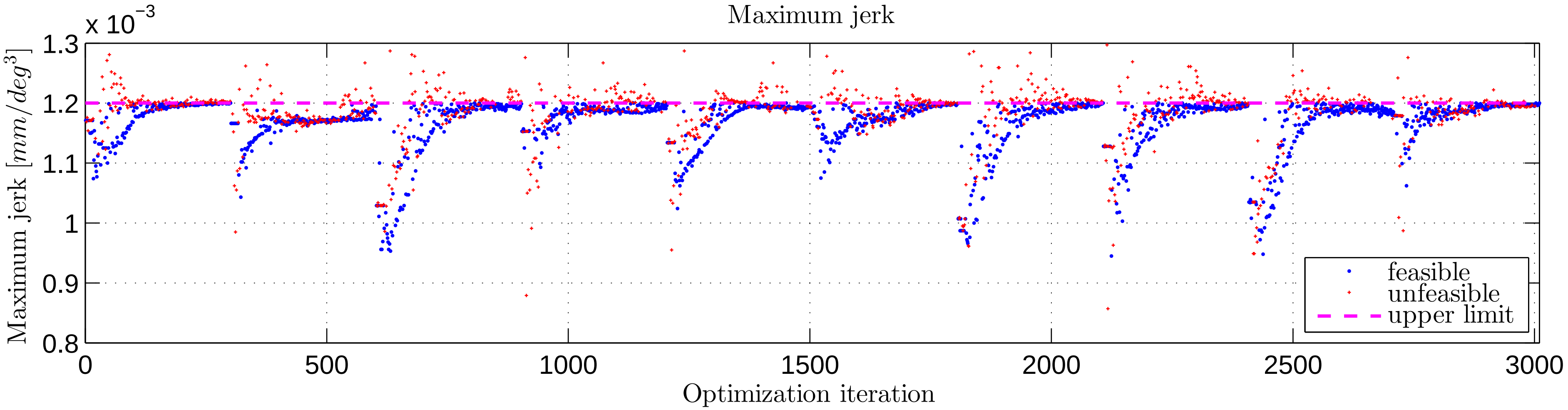


Figure A.3: Course of maximum jerk through through optimization. Small DOE matrix.

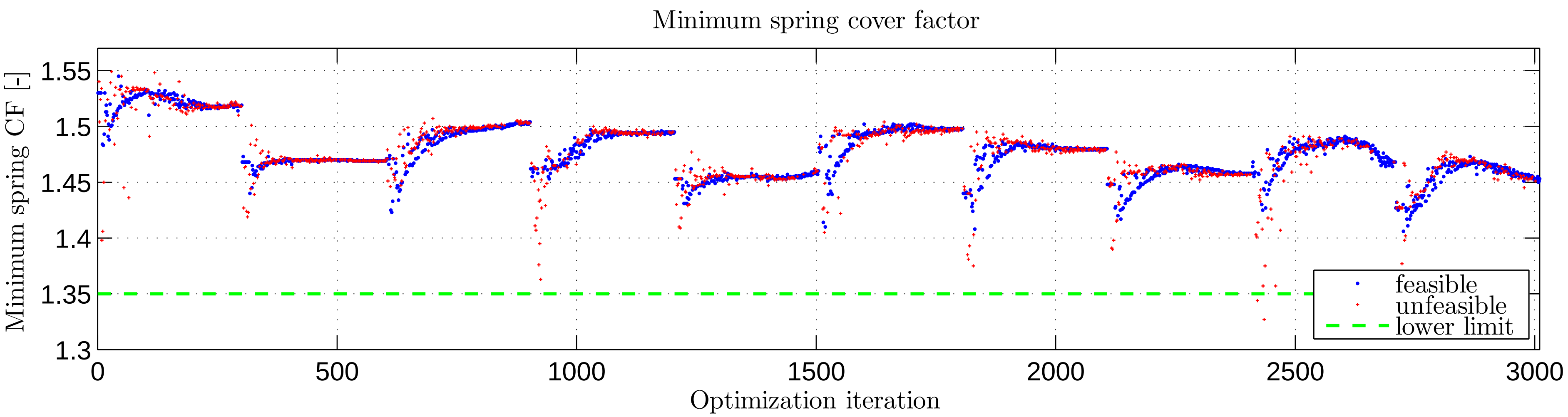


Figure A.4: Course of minimum spring cover factor through the optimization. Small DOE matrix.

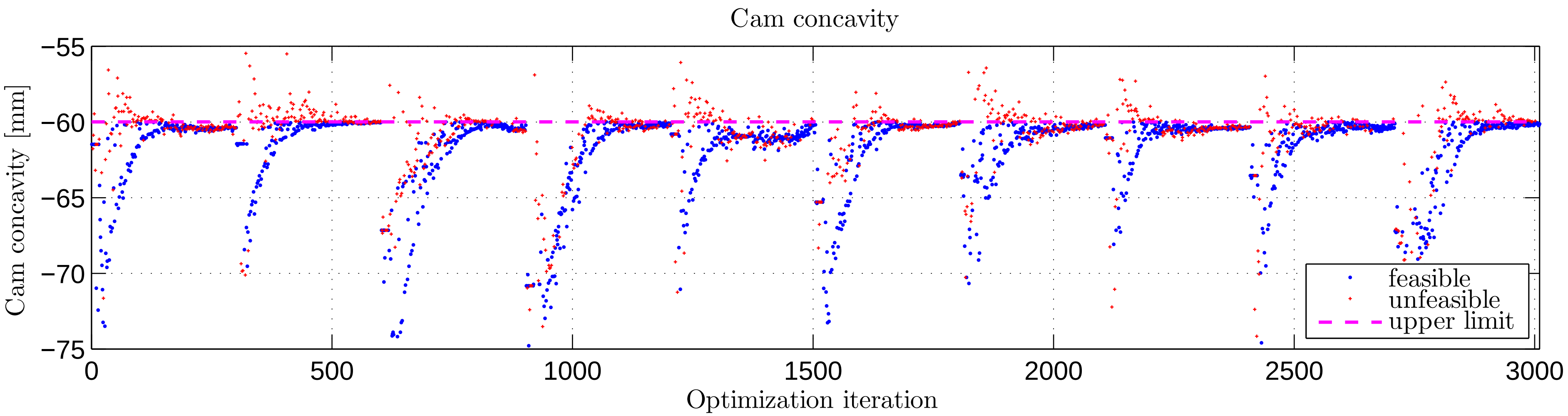


Figure A.5: Course of cam concavity through the optimization. Small DOE matrix.

Table A.3: Medium DOE results to be optimized.

Run	BC	BO	CO	DC	DO	EC	EO	TH1C	TH1O	TH2C	TH2O	TH3C	TH3O	TH5C	TH5O	CC	Max jerk	Af/An	JerkOk	Area integral	Cam concavity	Contact stress	Spring cover factor	Spring safety factor	Valve seating velocity	Spring surge	Cam to roller lash
Limits																$\langle 0.3; 1\rangle$ [-]	$\langle 0; 0.0012\rangle$ [mm/deg^3]	$\langle 0; 3\rangle$ [-]	Ok	max	$(-\infty; -60\rangle$ & $\langle 0; \infty)$ [mm]	$\langle 0; 1250\rangle$ [Mpa]	$\langle 1.35; \infty)$ [-]	$\langle 1.1; \infty\rangle$ [-]	$\langle 0; 0.3\rangle$ [m/s]	$\langle 0; 1\rangle$ [mm]	$\langle 0; 0.05\rangle$ [mm]
1	0.72	0.44	0.91	0.96	1.03	2.49	1.39	9.8	5.6	6.1	4.6	10.9	16.6	25.8	21.6	0.628	1.19E-03	2.24	1	337.76	-61.90	618.7	1.48	1.81	0.23	0.34	1.2E-02
2	0.86	0.49	0.59	1.19	0.99	0.52	1.56	6.2	7.7	8.3	6.9	10.5	11.3	19.9	23.5	0.341	9.84E-04	2.30	1	337.14	-60.00	633.0	1.48	1.64	0.25	0.37	7.4E-04
3	0.90	0.45	0.67	1.08	1.10	1.56	2.16	7.0	8.3	3.4	5.3	16.2	12.9	20.6	23.3	0.613	1.17E-03	2.11	1	336.33	-61.32	621.6	1.46	1.70	0.20	0.34	7.3E-04
4	0.69	0.77	0.98	0.96	1.10	0.76	0.71	6.5	4.9	6.3	9.1	15.0	12.4	18.7	16.7	0.527	1.18E-03	2.02	1	336.18	-70.38	614.6	1.51	1.77	0.21	0.36	1.1E-02
5	0.74	0.69	0.86	1.22	1.04	0.72	0.88	7.8	5.3	5.9	6.1	13.1	15.1	15.2	20.8	0.638	1.17E-03	2.19	1	335.97	-62.95	632.3	1.52	1.67	0.21	0.39	1.9E-02
6	0.99	0.42	0.98	1.10	1.04	0.93	2.34	7.9	10.6	6.5	2.7	10.5	13.9	24.0	17.8	0.421	1.05E-03	2.23	1	335.95	-61.66	633.6	1.43	1.70	0.27	0.40	7.6E-03
7	0.96	0.55	0.43	1.04	1.07	0.92	0.91	5.2	6.2	4.4	7.8	18.2	11.8	22.5	23.9	0.920	1.09E-03	2.12	1	335.89	-63.80	613.2	1.47	1.78	0.21	0.32	-5.5E-05
8	0.92	0.62	0.70	0.98	0.97	2.46	2.00	12.2	9.0	2.7	5.2	12.4	12.4	21.4	17.0	0.855	1.08E-03	2.27	1	335.75	-60.25	621.1	1.48	1.67	0.27	0.38	1.6E-02
9	0.50	0.52	0.74	1.28	1.00	0.90	1.38	5.9	6.2	5.2	6.8	16.5	13.7	27.4	19.5	0.804	1.07E-03	2.20	1	335.71	-65.65	623.5	1.47	1.69	0.20	0.36	1.1E-02
10	0.40	0.43	0.93	0.98	1.04	1.34	1.23	7.3	6.3	9.7	5.2	10.2	15.6	27.8	22.6	0.530	9.88E-04	2.19	1	335.70	-63.80	618.3	1.46	1.78	0.23	0.36	8.9E-03

Table A.4: Medium DOE after optimization.

Run	BC	BO	CO	DC	DO	EC	EO	TH1C	TH1O	TH2C	TH2O	TH3C	TH3O	TH5C	TH5O	CC	Max jerk	Af/An	JerkOk	Area integral	Cam concavity	Contact stress	Spring cover factor	Spring safety factor	Valve seating velocity	Spring surge	Cam to roller lash
Limits																$\langle 0.3; 1\rangle$ [-]	$\langle 0; 0.0012\rangle$ [mm/deg^3]	$\langle 0; 3\rangle$ [-]	Ok	max	$(-\infty; -60\rangle$ & $\langle 0; \infty)$ [mm]	$\langle 0; 1250\rangle$ [Mpa]	$\langle 1.35; \infty)$ [-]	$\langle 1.1; \infty\rangle$ [-]	$\langle 0; 0.3\rangle$ [m/s]	$\langle 0; 1\rangle$ [mm]	$\langle 0; 0.05\rangle$ [mm]
1	0.72	0.44	0.96	0.97	1.02	2.35	1.35	9.37	5.72	5.27	4.70	10.01	16.17	26.87	23.12	0.332	1.20E-03	2.32	1	339.5	-60.05	625.8	1.48	1.73	0.25	0.32	7.60E-06
2	0.99	0.40	0.57	1.13	1.06	0.95	1.85	5.12	7.44	8.21	6.73	10.09	11.11	22.14	23.13	0.310	1.19E-03	2.30	1	340.43	-60.00	632.9	1.49	1.58	0.28	0.36	-4.19E-08
3	0.94	0.41	0.75	1.09	1.10	1.47	2.15	5.90	8.26	4.01	5.35	15.12	12.76	21.12	22.94	0.335	1.19E-03	2.17	1	338.46	-60.00	625.2	1.45	1.67	0.24	0.32	8.77E-06
4	0.80	0.72	0.91	1.02	1.11	0.65	1.20	6.08	5.02	5.59	8.45	14.45	11.68	15.15	15.63	0.410	1.20E-03	2.18	1	340.08	-60.66	623.1	1.55	1.60	0.24	0.36	-1.87E-08
5	0.84	0.56	1.00	1.09	1.04	1.34	1.27	7.70	5.37	5.40	5.86	12.80	14.82	16.50	19.26	0.939	1.20E-03	2.29	1	339.72	-60.42	629.0	1.50	1.74	0.25	0.37	1.01E-05
6	1.00	0.46	0.93	1.14	1.03	1.37	2.31	5.70	9.45	6.57	3.52	11.71	13.47	20.24	19.29	0.305	1.19E-03	2.32	1	339.37	-60.14	630.6	1.46	1.68	0.20	0.34	8.36E-06
7	0.97	0.54	0.45	1.06	1.05	1.16	1.37	4.89	5.59	3.68	7.83	17.88	11.32	22.87	22.66	0.990	1.20E-03	2.29	1	340.39	-60.30	617.8	1.51	1.67	0.21	0.37	-4.59E-08
8	0.94	0.64	0.76	0.98	1.01	2.07	2.12	11.66	8.72	2.24	5.98	10.95	10.93	21.15	16.65	0.439	1.18E-03	2.31	1	338.59	-60.14	630.3	1.52	1.65	0.28	0.37	2.00E-06
9	0.52	0.45	0.75	1.19	1.04	1.25	1.48	5.07	6.10	4.35	7.40	15.56	11.82	29.42	15.10	0.483	1.17E-03	2.35	1	341.05	-60.07	627.4	1.49	1.64	0.22	0.35	-3.09E-08
10	0.44	0.46	0.95	0.96	1.03	1.55	1.35	5.92	5.56	9.10	5.26	10.43	15.47	27.75	22.21	0.381	1.20E-03	2.32	1	339.72	-60.19	619.3	1.48	1.76	0.27	0.32	9.53E-06

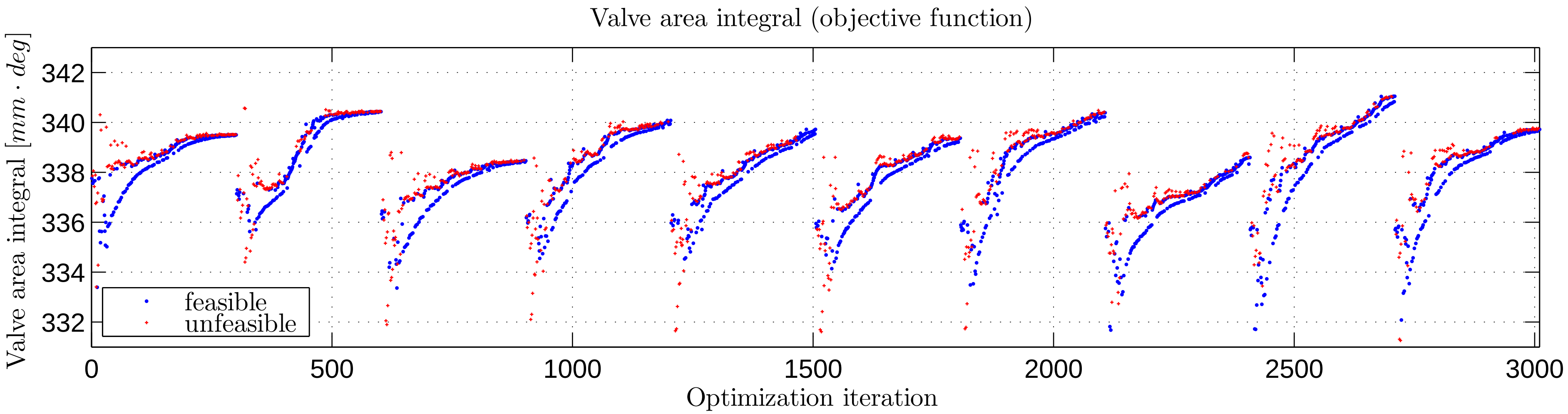


Figure A.6: Course of objective function through the optimization process. Medium DOE matrix.

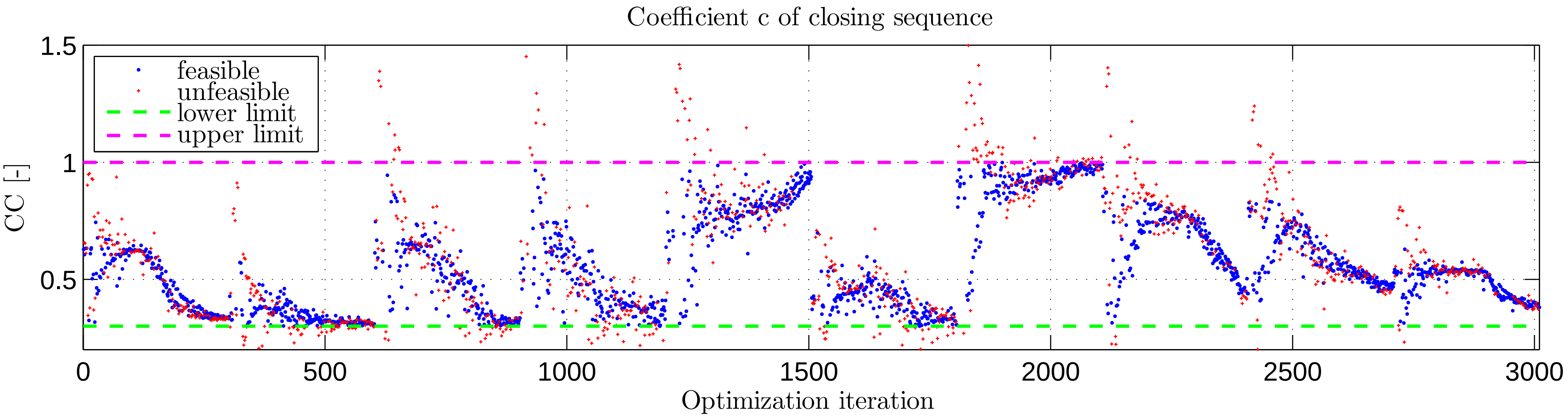


Figure A.7: Course of coefficient c of the closing sequence through the optimization. Medium DOE matrix.

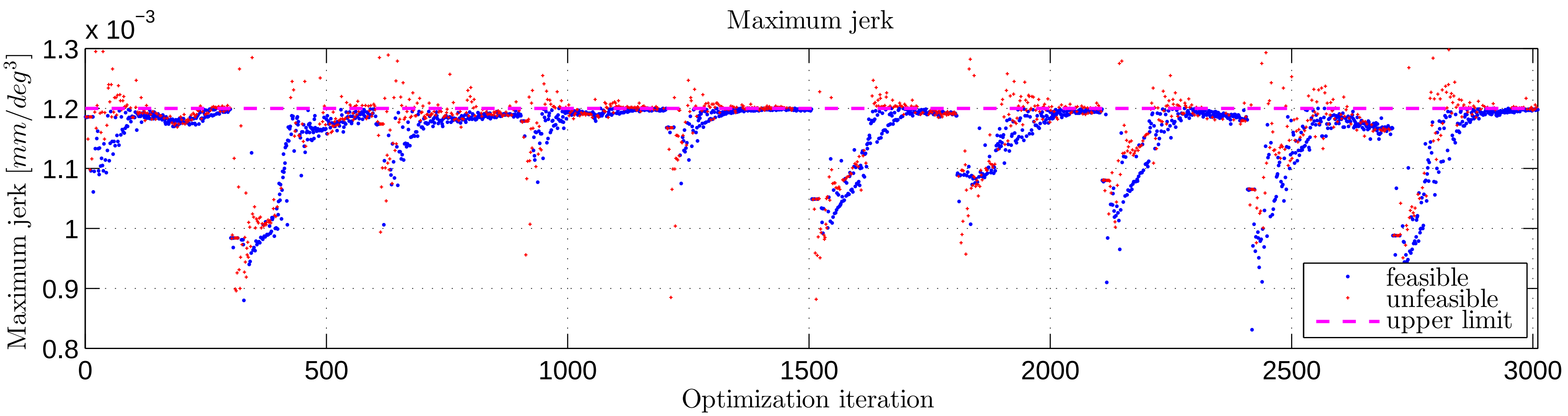


Figure A.8: Course of maximum jerk through through optimization. Medium DOE matrix.

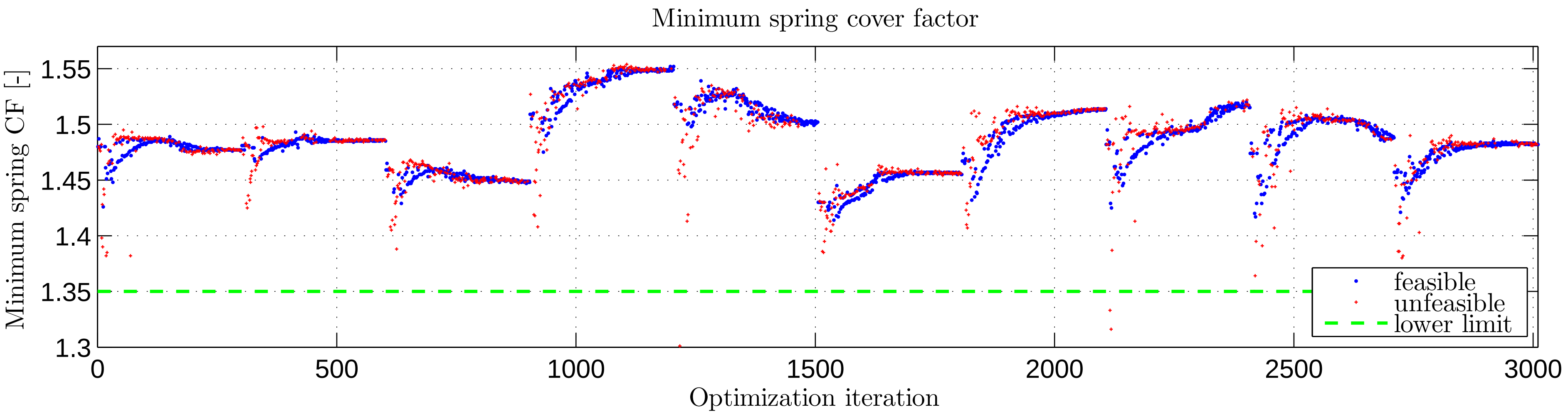


Figure A.9: Course of minimum spring cover factor through the optimization. Medium DOE matrix.

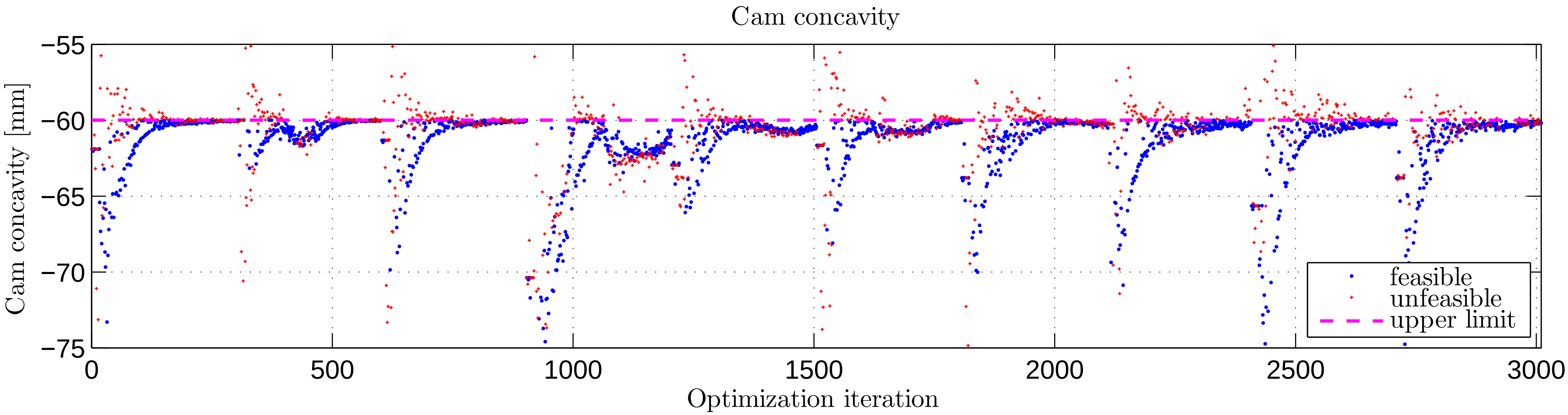


Figure A.10: Course of cam concavity through the optimization. Medium DOE matrix.

Table A.5: Large DOE to be optimized.

Run	BC	BO	CO	DC	DO	EC	EO	TH1C	TH1O	TH2C	TH2O	TH3C	TH3O	TH5C	TH5O	CC	Max jerk	Af/An	JerkOk	Area integral	Cam concavity	Contact stress	Spring cover factor	Spring safety factor	Valve seating velocity	Spring surge	Cam to roller lash
Limits																$\langle 0.3; 1\rangle$ [-]	$\langle 0; 0.0012\rangle$ $[mm/deg^3]$	$\langle 0; 3\rangle$ [-]	Ok	max	$(-\infty; -60\rangle$ & $\langle 0; \infty)$ $[mm]$	$\langle 0; 1250\rangle$ $[Mpa]$	$\langle 1.35; \infty)$ [-]	$\langle 1.1; \infty\rangle$ [-]	$\langle 0; 0.3\rangle$ $[m/s]$	$\langle 0; 1\rangle$ $[mm]$	$\langle 0; 0.05\rangle$ $[mm]$
1	0.72	0.44	0.91	0.96	1.03	2.49	1.39	9.8	5.6	6.1	4.6	10.9	16.6	25.8	21.6	0.628	1.19E-03	2.24	1	337.76	-61.90	618.7	1.48	1.81	0.23	0.34	1.2E-02
2	0.86	0.49	0.59	1.19	0.99	0.52	1.56	6.2	7.7	8.3	6.9	10.5	11.3	19.9	23.5	0.341	9.84E-04	2.30	1	337.14	-60.00	633.0	1.48	1.64	0.25	0.37	7.4E-04
3	0.90	0.45	0.67	1.08	1.10	1.56	2.16	7.0	8.3	3.4	5.3	16.2	12.9	20.6	23.3	0.613	1.17E-03	2.11	1	336.33	-61.32	621.6	1.46	1.70	0.20	0.34	7.3E-04
4	0.69	0.77	0.98	0.96	1.10	0.76	0.71	6.5	4.9	6.3	9.1	15.0	12.4	18.7	16.7	0.527	1.18E-03	2.02	1	336.18	-70.38	614.6	1.51	1.77	0.21	0.36	1.1E-02
5	0.74	0.69	0.86	1.22	1.04	0.72	0.88	7.8	5.3	5.9	6.1	13.1	15.1	15.2	20.8	0.638	1.17E-03	2.19	1	335.97	-62.95	632.3	1.52	1.67	0.21	0.39	1.9E-02
6	0.99	0.42	0.98	1.10	1.04	0.93	2.34	7.9	10.6	6.5	2.7	10.5	13.9	24.0	17.8	0.421	1.05E-03	2.23	1	335.95	-61.66	633.6	1.43	1.70	0.27	0.40	7.6E-03
7	0.96	0.55	0.43	1.04	1.07	0.92	0.91	5.2	6.2	4.4	7.8	18.2	11.8	22.5	23.9	0.920	1.09E-03	2.12	1	335.89	-63.80	613.2	1.47	1.78	0.21	0.32	-5.5E-05
8	0.92	0.62	0.70	0.98	0.97	2.46	2.00	12.2	9.0	2.7	5.2	12.4	12.4	21.4	17.0	0.855	1.08E-03	2.27	1	335.75	-60.25	621.1	1.48	1.67	0.27	0.38	1.6E-02
9	0.50	0.52	0.74	1.28	1.00	0.90	1.38	5.9	6.2	5.2	6.8	16.5	13.7	27.4	19.5	0.804	1.07E-03	2.20	1	335.71	-65.65	623.5	1.47	1.69	0.20	0.36	1.1E-02
10	0.40	0.43	0.93	0.98	1.04	1.34	1.23	7.3	6.3	9.7	5.2	10.2	15.6	27.8	22.6	0.530	9.88E-04	2.19	1	335.70	-63.80	618.3	1.46	1.78	0.23	0.36	8.9E-03

Table A.6: Large DOE after optimization.

Run	BC	BO	CO	DC	DO	EC	EO	TH1C	TH1O	TH2C	TH2O	TH3C	TH3O	TH5C	TH5O	CC	Max jerk	Af/An	JerkOk	Area integral	Cam concavity	Contact stress	Spring cover factor	Spring safety factor	Valve seating velocity	Spring surge	Cam to roller lash
Limits																$\langle 0.3; 1\rangle$ [-]	$\langle 0; 0.0012\rangle$ $[mm/deg^3]$	$\langle 0; 3\rangle$ [-]	Ok	max	$(-\infty; -60\rangle$ & $\langle 0; \infty)$ $[mm]$	$\langle 0; 1250\rangle$ $[Mpa]$	$\langle 1.35; \infty)$ [-]	$\langle 1.1; \infty\rangle$ [-]	$\langle 0; 0.3\rangle$ $[m/s]$	$\langle 0; 1\rangle$ $[mm]$	$\langle 0; 0.05\rangle$ $[mm]$
1	0.57	0.43	0.40	1.20	1.05	0.87	1.29	5.97	5.44	9.17	8.44	10.04	10.90	20.07	15.61	0.430	1.20E-03	2.33	1	340.03	-60.16	633.4	1.48	1.63	0.25	0.35	-3.01E-08
2	0.89	0.57	0.50	1.03	1.06	1.55	1.26	6.42	5.47	4.22	8.23	13.79	11.00	32.92	23.18	0.421	1.20E-03	2.29	1	341.11	-60.20	626.4	1.53	1.62	0.24	0.38	-5.84E-08
3	0.53	0.40	0.88	1.25	0.99	1.06	1.18	4.32	5.27	5.10	9.56	17.42	10.89	15.01	15.30	0.486	1.20E-03	2.29	1	340.17	-65.06	622.5	1.47	1.71	0.24	0.36	1.07E-05
4	0.99	0.56	0.78	1.03	1.09	1.53	1.11	6.49	5.02	6.69	6.57	10.32	14.44	34.85	25.51	0.303	1.19E-03	2.18	1	340.37	-60.34	630.0	1.49	1.67	0.23	0.34	1.33E-05
5	0.86	0.48	0.63	1.09	1.05	1.87	1.20	7.78	5.49	5.28	8.06	10.24	11.50	26.01	16.66	0.304	1.18E-03	2.33	1	341.53	-60.10	636.2	1.50	1.56	0.21	0.38	-4.55E-08
6	1.00	0.42	0.71	1.13	1.00	1.25	1.36	5.65	5.61	6.04	7.02	11.87	13.26	34.67	25.26	0.354	1.20E-03	2.32	1	341.06	-60.56	632.7	1.49	1.62	0.24	0.35	-5.35E-08
7	0.93	0.57	0.94	1.08	1.10	0.48	1.49	5.62	5.73	6.15	7.89	12.09	11.67	19.99	15.65	0.406	1.20E-03	2.24	1	341.59	-60.08	632.5	1.50	1.55	0.26	0.38	-2.14E-08
8	0.88	0.43	0.46	1.16	1.13	1.54	1.42	5.83	5.36	5.45	8.95	12.94	10.43	22.93	20.00	0.380	1.20E-03	2.18	1	340.73	-60.08	631.0	1.49	1.63	0.26	0.36	-1.77E-08
9	0.87	0.61	0.74	1.02	1.08	1.34	1.15	5.43	5.19	8.09	8.75	10.68	10.88	20.36	18.63	0.363	1.20E-03	2.28	1	341.93	-60.09	626.2	1.52	1.57	0.25	0.40	-4.20E-08
10	0.84	0.41	0.89	0.97	1.01	1.58	1.63	7.72	6.59	1.51	4.11	18.09	16.10	18.90	24.38	0.863	1.20E-03	2.30	1	338.77	-60.14	616.4	1.47	1.79	0.21	0.31	9.67E-06

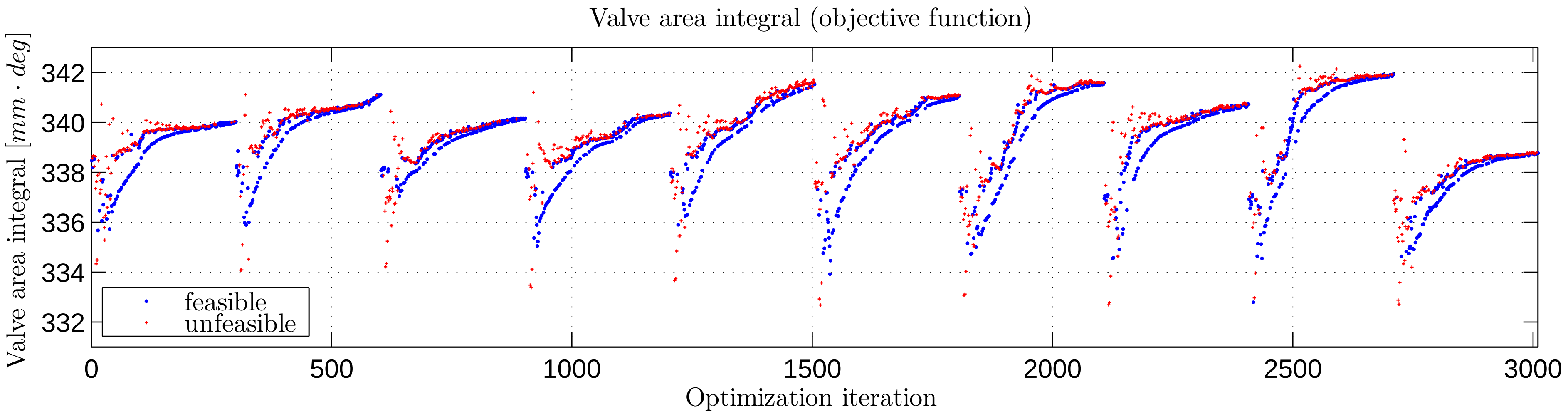


Figure A.11: Course of objective function through the optimization process. Large DOE matrix.

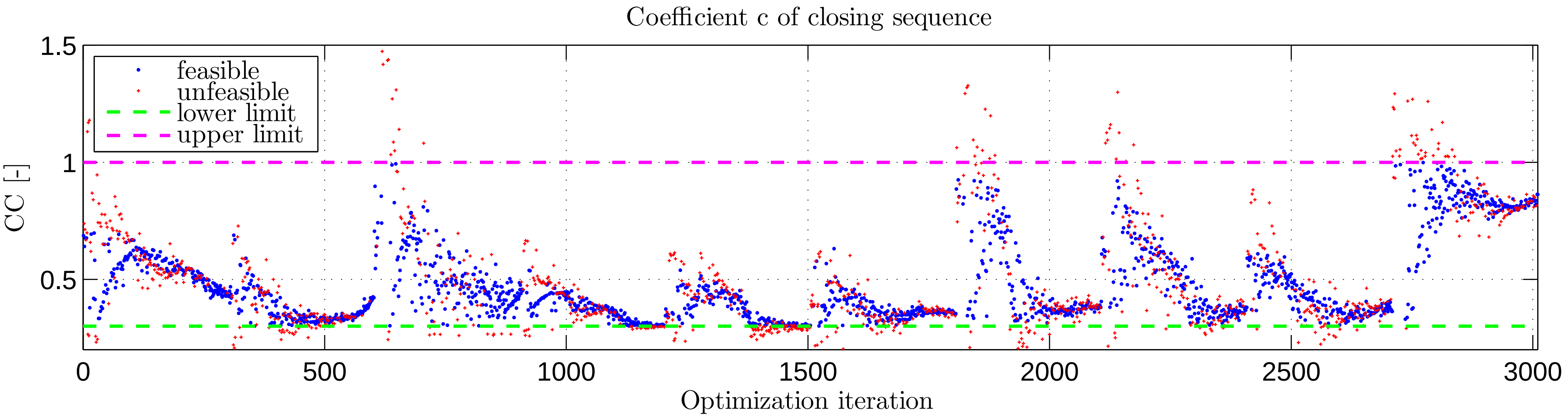


Figure A.12: Course of coefficient c of the closing sequence through the optimization. Large DOE matrix.

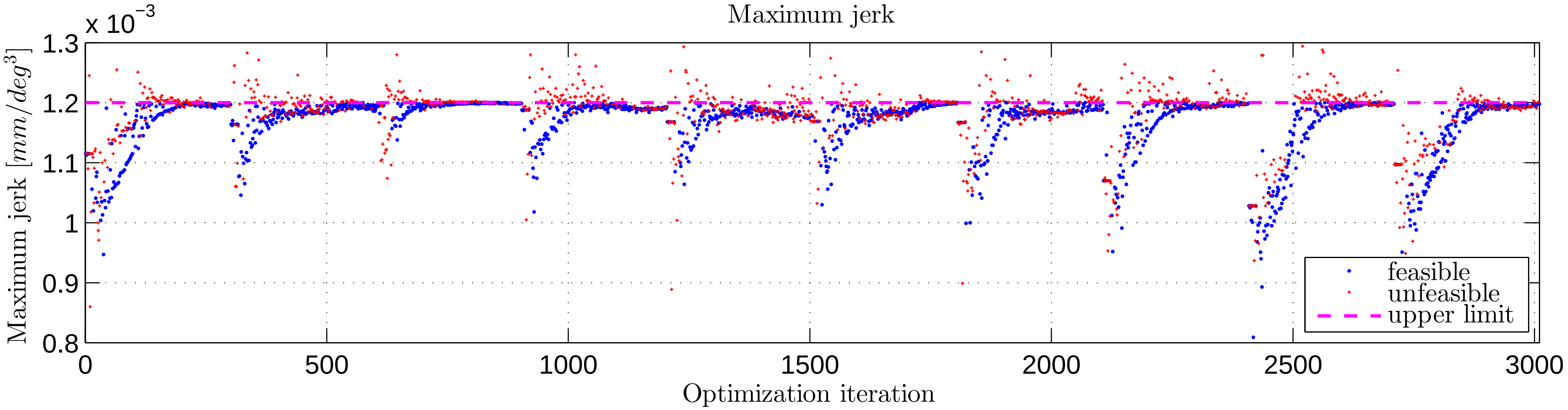


Figure A.13: Course of maximum jerk through through optimization. Large DOE matrix.

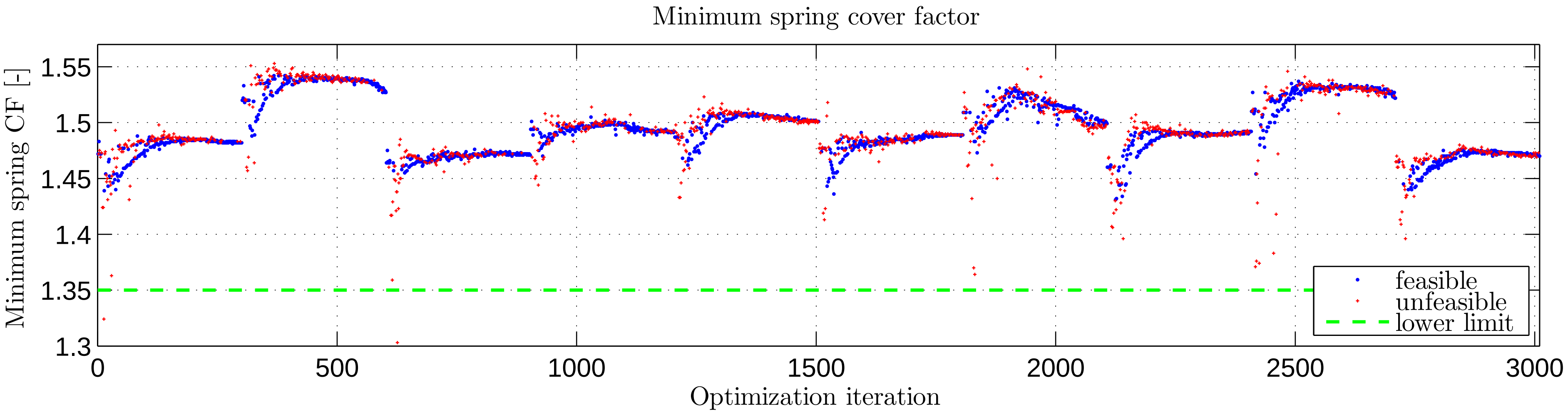


Figure A.14: Course of minimum spring cover factor through the optimization. Large DOE matrix.

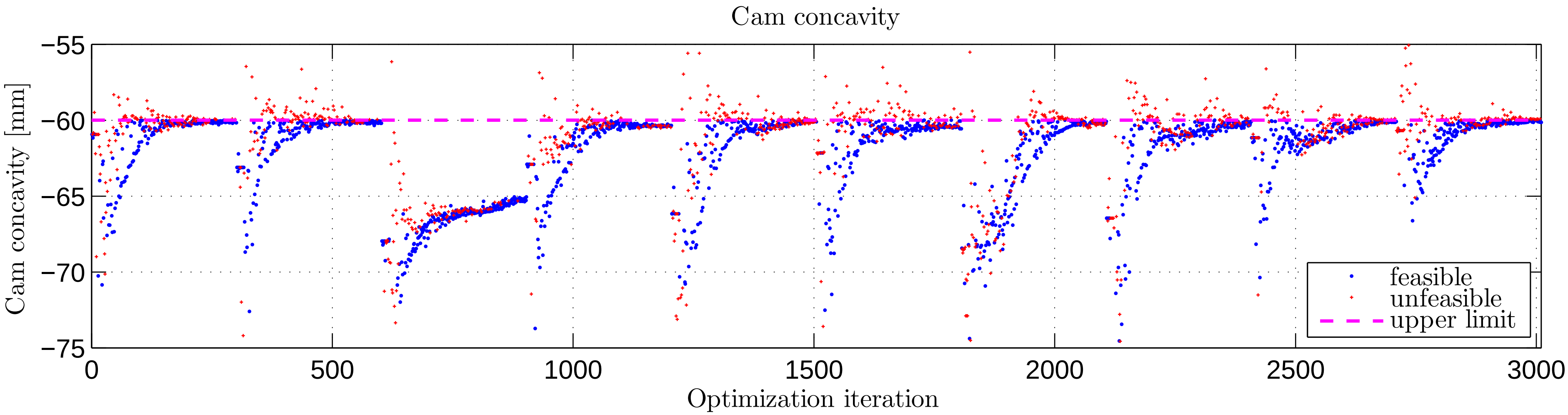


Figure A.15: Course of cam concavity through the optimization. Large DOE matrix.

Appendix B

Contents of the Attached CD

The CD contains three directories:
Thesis: Full text of the thesis in Portable Document Format.
CamOpt: Isight model and template files. The given models in Valkin and Valdyn could not be included, thus they were substituted by newly created ones. They can be only used as templates for designing the data exchange with Isight.
Matlab: Matlab codes of `main_CamPol` and `CamPolDOE` and the associated functions.