Automatic Generation of Combustion Engine Models using MatLab & Idle Drive Train Model in MatLab / Simulink

L.H.J. Römers

Tutors:
Dipl. Ing. B. Pennec LuK GmbH & Co.
Dr. S. Jones LuK GmbH & Co.
Dr. P.A. Veenhuizen Technical University of Eindhoven

December 2005 – March 2006
Preface

I would like to thank all employees of the dynamic simulation department at LuK Bühl, for giving me the chance to do an interesting and varied traineeship. Besides gaining a lot of technical knowledge, I have experienced what it is like to be part of a professional development team. Special thanks go out to Bertrand Pennec and Dr. Stephen Jones for their great support during the last months. I would also like to thank Dr. Ad Kooy, as my contact at LuK, for giving me the chance to gain these experiences.

Luc Römers
Table of contents

1. Introduction 4
   1.1 History of the company LuK GmbH & Co. 4
   1.2 Goal of the traineeship 6

Automatic generation of combustion engine models using MatLab 7

2. Generation of pressure-based engine models 8
   2.1 Geometry and kinematics of the crank mechanism 10
   2.2 Dynamics of the crank mechanism 11
   2.3 Mass torque 12
   2.4 Compression / expansion torque 13
   2.5 Combustion torque 17
      2.5.1 Isermann method 18
      2.5.2 Isermann/Pennec method 20
   2.6 Total torque resulting from gas forces 22
   2.7 Parameter maps 23
   2.8 Reducing the parameter maps 31
   2.9 Peak-to-peak comparison 33
   2.10 Output for DyFaSim software 40

3. Generation of standard engine models 42
   3.1 Combustion engine data sheet 42
   3.2 Replacing missing data by estimates 43
      3.2.1 Geometry data 43
      3.2.2 Net torque curve at full load 43
      3.2.3 Drag torque curve 44
      3.2.4 Intake pressure curve at full load 45

4. Graphical User Interface in MatLab 46
   4.1 Graphical User Interface, pressure-based engine model 46
   4.2 Graphical User Interface, standard engine model 54

5. Comparison between pressure-based and standard models 59

Idle drive train model in Matlab / Simulink 61

6. Idle drive train model in Matlab / Simulink 61
   6.1 Virtual test bench for Simulink representation of LuK arc springs 61
   6.2 Drive train model for idle simulation 64
   6.3 Comparison with DyFaSim counterpart 67
7. Conclusions

List of symbols

References
1. Introduction

1.1 History of the company LuK GmbH & Co.

In Bühl, at the edge of the Black Forest, the headquarters of the international company LuK is situated. At the company LuK, 9000 employees working at 20 production locations in Germany, Brazil, Great-Britain, India, Mexico, South-Africa, Hungarian, France, Korea and the United States yearly produce more than 14 million clutches for passenger cars and tractors, over 3 million lock-up clutches and over 5 million dual-mass flywheels. One in every four cars produced worldwide contains a LuK clutch. The LuK dual-mass flywheel (DMF) increases comfort and decreases fuel consumption in millions of cars worldwide.

**History of LuK GmbH & Co.**

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1965</td>
<td>Foundation of LuK in Bühl Germany</td>
</tr>
<tr>
<td>1974</td>
<td>Foundation of LuK Brazil and LuK Mexico</td>
</tr>
</tbody>
</table>
| 1976 | Foundation of AS (Autoteile-Service GmbH)  
Development of torsion damped clutch plates |
| 1977 | Foundation of LuK Inc. USA |
| 1982 | New LuK location in Unna |
| 1985 | LuK dual-mass flywheel (DMF): new standard for noise isolation in the drive train |
| 1986 | Takeover of AFT in Werdohl |
| 1987 | Takeover of Repco S.A. (today LuK South-Africa)  
Takeover Laycock in England (today LuK UK) |
| 1989 | Opening LuK (UK) Ldt. |
| 1990 | Takeover of TCM Ldt. in Hereford/England (today Hereford Ltd.) |
| 1991 | Joint-venture between LuK and Barmag (future BarLuK)  
Takeover of automotive-hydraulic division Vickers (today LuK Fahrzeug-Hydraulik)  
First electronic clutch management for BMW Alpina |
<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1992</td>
<td>Foundation of LuK Getriebe-Systeme in Bühl</td>
</tr>
<tr>
<td>1993</td>
<td>Complete takeover BarLuK (today LuK Automobiltechnik)</td>
</tr>
<tr>
<td>1994</td>
<td>In corporation with a car manufacturer LuK develops parts for a CVT transmission with a maximum torque of about 300 Nm</td>
</tr>
<tr>
<td>1995</td>
<td>Foundation LuK Korea</td>
</tr>
<tr>
<td>1996</td>
<td>Foundation Beijing (LuK) Liaison Office in China</td>
</tr>
<tr>
<td></td>
<td>Joint-venture between Rane und LuK (Rane LuK Clutch Ltd.) in India</td>
</tr>
<tr>
<td></td>
<td>Foundation LuK Savaria in Hungarian (start of production, late 1997)</td>
</tr>
<tr>
<td></td>
<td>SAC-clutch (Self Adjusting Clutch)</td>
</tr>
<tr>
<td>1997</td>
<td>The LuK-EKM is produced in series for the Mercedes A-class</td>
</tr>
<tr>
<td></td>
<td>The production location in Wooster/Ohio starts to produce the first torque-converter for light trucks in series</td>
</tr>
<tr>
<td>1998</td>
<td>Takeover of Kongsberg TechMatic in England and Norway</td>
</tr>
<tr>
<td></td>
<td>Foundation of AS divisions in Spain, UK, USA, Brazil, Mexico, South-Africa, Poland, Russia und Czechia</td>
</tr>
<tr>
<td>1999</td>
<td>Takeover of Holzhäuser, Germany</td>
</tr>
<tr>
<td></td>
<td>Foundation AS Argentina</td>
</tr>
<tr>
<td></td>
<td>Opening LuK Leamington, UK</td>
</tr>
<tr>
<td></td>
<td>New foundry LuK Brazil</td>
</tr>
<tr>
<td></td>
<td>LuK presents CVT parts for high torques. Audi Multitronic®: the first efficient CVT transmission for more than 300 Nm is produced in series</td>
</tr>
<tr>
<td></td>
<td>Valeo sells LuK shares to INA Holding</td>
</tr>
<tr>
<td>2000</td>
<td>Opening of LuK production location in Bußmatten, Germany</td>
</tr>
<tr>
<td></td>
<td>LuK takes over the production of clutch release systems from INA</td>
</tr>
<tr>
<td></td>
<td>Opening of LuK tool shop in Kappelrodeck, Germany</td>
</tr>
<tr>
<td></td>
<td>Automated manual transmission, Easytronic®, is produced in series for the Opel Corsa</td>
</tr>
</tbody>
</table>
2001 | LuK sells LuK Norway
    | LuK fully takes over Rane LuK (future LuK India Private Ltd.)

2002 | The 20 millionth dual-mass flywheel is produced in Bühl

2003 | The LuK integrated crank-shaft damper is produced in series for the VW Touareg

2004 | In Bühl the production of torque-converters is started

2005 | Takeover of APTEC (future LuK Friction)
    | Takeover of AP France (future LuK France)
    | Opening LuK Korea

Research and development are of great importance at LuK. Therefore about one of every six employees works in this field. The inventiveness of the product developer creates innovations for the automotive world of tomorrow.

1.2 Goal of the traineeship

The main assignment is creating an automated and simple way of generating combustion engine models, for use in the LuK-DyFaSim simulation software. For this, the mathematical tool MatLab will be used.

Because the schedule allowed it, there was the possibility to do a second assignment. This assignment consists of the build-up of an idle drive train model in MatLab / Simulink. This model will be used to investigate the unwanted phenomenon of sub-harmonic vibrations at idle operation.
Automatic generation of combustion engine models using MatLab

The goal of this assignment is to create a method, which allows automatic generation of combustion engine models for the LuK DyFaSim simulation software. DyFaSim is a software package developed by LuK and is used to simulate the dynamics of the vehicle drive train. The program is based on a modular structure, allowing single elements like the combustion engine, dual-mass flywheel, clutch and transmission to be joined to form a complete drive train. The combustion engine module of DyFaSim uses an empirical method to simulate the dynamic engine torque in all operating points. To create a DyFaSim engine model, a large amount of data from the car manufacturer needs to be processed. This has been a semi-manual process until now, carried out by only a few engineers due to its complex nature. Automating the complete modelling process would reduce the manual effort, thereby increasing the capacity of the involved engineers. For this cause, the mathematical software package MatLab will be used to create an automatic modelling process, based on the semi-manual process used until now.

The automatic engine model generation will handle two different situations:
- creating an accurate combustion engine model based on cylinder pressure data
- creating a standard combustion engine model, in case very little data is available

These two situations will be referred to as pressure-based model generation and standard model generation. A full description of the pressure-based model will be given in chapter 2, the standard model will be treated in chapter 3. In chapter 4 an explanation of the graphical user interface in MatLab will be presented.

![Figure 1.1: overview of the MatLab program](image)
2. Generation of pressure-based engine models

The purpose of this model is to simulate the dynamic torque output in all operating points. The model is based on cylinder geometry data and cylinder pressure data, these are delivered by the car manufacturer. First, the torque resulting from one single cylinder will be modelled, which will finally be cascaded into a whole engine. The basic operating method of a four-stroke cycle is shown in the next figure.

![Four-stroke cycle](image)

**Figure 2.1: four-stroke cycle [ENCA98]**

Compression and combustion cause a pressure rise inside the cylinder. An example is given in the next diagram.

![Cylinder pressure over full cycle](image)

**Figure 2.2: cylinder pressure over full cycle**

The pressure rise results in a force on the piston, which in turn results in a crankshaft torque. This torque varies strongly with crank angle, therefore it is called the dynamic cylinder torque.
If we look at the dynamic torque resulting from one cylinder, it is possible to subdivide it into three sources:

- torque due to acceleration forces working on the oscillating piston mass
- torque due to pressure build-up resulting from compression/expansion
- torque due to pressure build-up resulting from the combustion process

The last two components form the torque due to gas forces. Examples of all three components are given in the figure below.

These three components will be described using cylinder geometry data, cylinder pressure data and an empirical part. Once all components are simulated, they can be summed to form the total dynamic torque resulting from one cylinder. It is important to realise that this is the gross engine torque. The net torque can be found by subtracting the engine friction. However, in the LuK DyFaSim software, friction is taken into account outside the engine module.

Before starting the modelling process of these torque components, a description of the geometry, kinematics and dynamics of the crank mechanism will be given.
2.1 Geometry and kinematics of the crank mechanism

The following schematic drawing shows the crank mechanism. It consists of the piston, connecting rod and the crankshaft.

![Crank mechanism diagram]

**Figure 2.4: crank mechanism**

Using these geometry parameters, the displacement of the piston (relative to TDC) can be described. From this expression the piston speed and acceleration can be derived [MOOG00]:

\[
x(\varphi) = r \left[ (1 - \cos \varphi) + \frac{1}{\lambda} (1 - \sqrt{1 - \lambda^2 \sin^2 \varphi}) \right]
\]

\[
\dot{x}(\varphi) = r \cdot \omega \left[ \sin \varphi + \frac{\lambda \cdot \sin(2\varphi)}{2 \cdot \sqrt{1 - \lambda^2 \sin^2 \varphi}} \right]
\]

\[
\ddot{x}(\varphi) = r \cdot \omega^2 \left[ \cos \varphi + \frac{\lambda \cdot \cos(2\varphi)}{\sqrt{1 - \lambda^2 \sin^2 \varphi}} + \frac{\lambda^3 \cdot \sin^2(2\varphi)}{4 \cdot \left(\sqrt{1 - \lambda^2 \sin^2 \varphi}\right)^3} \right]
\]

Here \(\omega\) represents the rotational speed of the crankshaft, also known as the engine speed.
2.2 Dynamics of the crank mechanism

Knowing the kinematics, a dynamical analysis of the piston/crank mechanism can be made. The next figure shows the forces acting on the various components, as a result of an applied force on the piston. The resulting force on the crankshaft can be divided into a radial part $F_{c,r}$ and a tangential part $F_{c,t}$. Of course only the tangential part will result in a crankshaft torque.

The relation between the applied force at the piston and the tangential force at the crankshaft can be described as follows [MOOG00]:

$$ F_{c,t} = F_{\text{applied}} \left( \sin \varphi + \frac{\lambda \cdot \sin(2\varphi)}{2 \cdot \sqrt{1 - \lambda^2 \sin^2 \varphi}} \right) $$ \hspace{1cm} (2.4)

If this expression is multiplied with the crank radius, the crankshaft torque results:

$$ T_{\text{crank}} = r \cdot F_{\text{applied}} \left( \sin \varphi + \frac{\lambda \cdot \sin(2\varphi)}{2 \cdot \sqrt{1 - \lambda^2 \sin^2 \varphi}} \right) $$ \hspace{1cm} (2.5)
2.3 Mass torque

From the kinematical analysis, the acceleration of the piston as a function of the crank angle is known. By using Newton’s second law, the forces involved can be determined:

\[ F = -m_{osc} \cdot x(\varphi) \]  

(2.6)

Where \( m_{osc} \) stands for the oscillating mass of one cylinder. This force can be converted to a torque working on the crankshaft [SEGL00]:

\[ T_{mass} = r^2 \cdot F \left( \sin \varphi + \frac{\lambda \cdot \sin(2\varphi)}{2 \cdot \sqrt{1 - \lambda^2 \sin^2 \varphi}} \right) \]  

(2.7)

\[ T_{mass} = -m_{osc} \cdot \omega^2 \cdot r^2 \left( \sin \varphi + \frac{\lambda \cdot \sin(2\varphi)}{2 \cdot \sqrt{1 - \lambda^2 \sin^2 \varphi}} \right) \times \left( \cos \varphi + \frac{\lambda \cdot \cos(2\varphi)}{\sqrt{1 - \lambda^2 \sin^2 \varphi}} + \frac{\lambda^3 \cdot \sin^2(2\varphi)}{4 \cdot \left(\sqrt{1 - \lambda^2 \sin^2 \varphi}\right)^3} \right) \]  

(2.8)

The mass torque depends quadratically on the engine speed. The next figure displays the mass torque over one complete cycle. It can be seen that the amplitude increases by a factor of four when the engine speed is doubled.

![Figure 2.6: mass torque over one full cycle](image)

The derived formula reveals that the mass torque depends only on cylinder geometry parameters and engine speed.
2.4 Compression / expansion torque

Compression of the gas inside the cylinder results in a pressure rise and consequently in a force on the piston. This force generates a torque due to compression/expansion at the crankshaft. Even in case no combustion process is taking place, the compression/expansion torque will still be present.

To simulate this torque component, an analysis of the cylinder pressure is needed. Cylinder pressure measurements, in various engine operating points, are delivered by the car manufacturer. For this pressure analysis, the following assumptions are made:

- the gas inside the cylinder is ideal
- the compression/expansion is adiabatic reversible, or isentropic
- the isentropic coefficient \( \kappa \) is constant, i.e. does not change with temperature

Under these conditions, one can state:

\[
P \cdot V^\kappa = \text{constant} \quad (2.9)
\]

\[
P_{\text{comp/exp}}(\phi) \cdot (V(\phi))^\kappa = P_{\text{intake}} \cdot (V_c + V_d)^\kappa \quad (2.10)
\]

\[
P_{\text{comp/exp}}(\phi) = P_{\text{intake}} \cdot \left(\frac{V_c + V_d}{V(\phi)}\right)^\kappa \quad (2.11)
\]

Where the angle dependent cylinder volume can be written as [PISC89]:

\[
V(\phi) = V_c + \frac{\pi \cdot d^2}{4} \cdot k(x(\phi)) \quad (2.12)
\]

\[
V(\phi) = \frac{\pi \cdot r \cdot d^2}{4} \cdot \left[\frac{2}{e - 1} + (1 - \cos \phi) + \frac{1}{\lambda} \cdot (1 - \sqrt{1 - \lambda^2 \sin^2 \phi})\right] \quad (2.13)
\]

This expression for the volume contains only cylinder geometry parameters, which are known.

The only remaining parameter needed to describe the compression/expansion pressure is the intake pressure. At this point the pressure measurements are used. The following graph shows an example of a cylinder pressure measurement. The intake pressure is found by taking an average between 175° and 185° crank angle, where the piston reaches bottom dead center. The isentropic equation can now be used to calculate the pressure due to compression/expansion over the next 360°. The next graph shows the total cylinder pressure and the component resulting from compression/expansion.
The compression/expansion pressure can be converted into a crankshaft torque:

\[
F_{piston}(\phi) = \frac{\pi \cdot d^2}{4} \left( P_{\text{comp/exp}}(\phi) - P_{\text{surrounding}} \right)
\]  
(2.14)

\[
T_{\text{comp/exp}}(\phi) = \frac{\pi \cdot d^2}{4} \left( P_{\text{comp/exp}}(\phi) - P_{\text{surrounding}} \right) \cdot r \cdot \left( \sin \phi + \frac{\lambda \cdot \sin(2\phi)}{2 \cdot \sqrt{1 - \lambda^2 \sin^2 \phi}} \right)
\]  
(2.15)
The same conversion is made for the measured total cylinder pressure. The resulting torque curves are displayed in the next graph.

![Figure 2.9: compression / expansion torque](image)

Before top dead center both curves are basically the same, since pressure rise due to combustion is not present in this region. There are differences however, since the simulated compression/expansion torque is based on ideal conditions. This is accounted for by introducing a correction factor, which is defined as follows:

\[
    cf = \frac{\text{Min}(T_{\text{measurement}})}{\text{Min}(T_{\text{comp/exp,sim}})}
\]

(2.16)

The simulated compression/expansion torque is multiplied with this factor, resulting in an equal minimum.

![Figure 2.10: corrected compression / expansion torque](image)
In the region between 540° and 720° there are still differences however, especially at high engine speeds. This occurs because the cylinder pressure does not completely drop to the intake pressure after the combustion process. There is a distinct difference between intake pressure and exhaust pressure.

![Graph showing torque vs. crank angle](image)

**Figure 2.11:** comp./exp. torque, without adapted exhaust pressure

To take this into account, the simulated compression/expansion pressure will be defined on three different regions:

\[
\begin{align*}
P_{\text{comp/exp}}(\phi) &= P_{\text{intake}} & \text{for } \phi < 180^\circ \\
     &= P_{\text{intake}} \left( \frac{V_c + V_d}{V(\phi)} \right)^\epsilon & \text{for } 180^\circ \leq \phi \leq 540^\circ \\
     &= P_{\text{exhaust}} & \text{for } \phi > 540^\circ 
\end{align*}
\]

(2.17)

\(P_{\text{exhaust}}\) is an average pressure taken from the measurement in the region between 625° and 635° crank angle. The simulated compression/expansion torque, based on this pressure definition, is displayed in the next graph. It can be seen that the differences in the exhaust region are much smaller now.
Figure 2.12: comp./exp. torque, with adapted exhaust pressure

This new compression/expansion torque has an average value different from zero. The average over one complete cycle can be expressed as follows [PENN02]:

\[
T_{\text{comp/exp,average}} = \frac{1}{4\pi} \int_{3\pi}^{4\pi} \left( P_{\text{exhaust}} - P_{\text{intake}} \right) \left( \sin(\phi) + \frac{\lambda \cdot \sin(2\phi)}{\sqrt{1 - \lambda^2 \sin^2(\phi)}} \right) d\phi
\]  

(2.18)

The purpose of this average value will become clear later on, when the combustion process is taken into account.

Hereby the analysis of the compression/expansion torque is completed. It is now possible to simulate this torque using geometry data and three parameters: correction factor, intake pressure and exhaust pressure.

\[
T_{\text{comp/exp}}(\phi) = T_{\text{comp/exp}}(\phi, \text{geometry}, cf, P_{\text{intake}}, P_{\text{exhaust}})
\]  

(2.19)

2.5 Combustion torque

Describing a combustion process using thermodynamics is difficult, since it depends on a large number of parameters. Therefore an empirical method, based on mathematics, is used. The original empirical model is known as the Isermann method. LuK has improved this method. The improved empirical model is called the Isermann/Pennec method. A short description of both methods will be given.

The crankshaft torque resulting from the combustion process can be found by calculating the difference between total gas torque and compression/expansion torque. The total gas torque is known from the cylinder pressure measurements; furthermore the compression/expansion
torque has been simulated. After subtracting both curves, the combustion torque remains. This is shown in the next figures.

![Combustion Torque Graph](image)

**Figure 2.13:** Combustion torque is the difference between both curves

![Torque Due to Combustion Graph](image)

**Figure 2.14:** Torque due to combustion

### 2.5.1 Isermann method

The Isermann method [ISER99] uses the following formula to describe the combustion torque shown in figure 2.14.

\[
T_{\text{combustion}}(\phi) = \frac{16 \cdot \pi \cdot T_{\text{static}}}{\phi_{\text{max}}} \cdot \phi^2 \cdot e^{-\frac{\phi}{\phi_{\text{max}}}} \quad \text{for} \quad 360^\circ < \phi < 540^\circ
\]  

(2.20)
It contains two parameters, $T_{\text{static}}$ and $\varphi_{\text{max}}$, which can be determined from the measured data. The static torque is the average value over one full cycle:

$$T_{\text{static}} = \frac{1}{4\pi} \int_0^{4\pi} T_{\text{measured}}(\varphi) d\varphi = \frac{1}{4\pi} \int_0^{4\pi} P \frac{dV}{d\varphi} d\varphi$$  \hspace{1cm} (2.21)

The angle of maximum combustion torque ($\varphi_{\text{max}}$) corresponds to the maximum in figure 2.14. The effect of both parameters on the combustion torque is shown in the next graphs [PENN02].

Experience has shown that the accuracy of the Isermann model is limited. Therefore the improved Isermann/Pennec model has been developed.
2.5.2 Isermann/Pennec method

This method uses two additional shape parameters to simulate the combustion torque more accurately. The derivation of this empirical model is beyond the scope of this report. The final Isermann/Pennec combustion torque is a function of the crank angle and four parameters:

\[ T_{\text{combustion}} = T_{\text{combustion}}(\varphi, T_{\text{stat}}, \varphi_{\text{max}}, \alpha, \beta) \]  

(2.22)

The effect of the additional shape parameters \( \alpha \) and \( \beta \) on the combustion torque is shown in the graphs below [PENN02].

![Figure 2.17: influence of \( \beta \) on the combustion torque](image1)

![Figure 2.18: influence of \( \alpha \) on the combustion torque](image2)
Having found the values for $T_{\text{static}}$ and $\phi_{\text{max}}$, the values for $\alpha$ and $\beta$ can be optimized. To compare different combinations, the standard deviation between measurement and simulation is calculated:

$$\sigma^2(\alpha, \beta) = \frac{1}{4\pi} \int_0^{4\pi} (T_{\text{measured}}(\phi) - T_{\text{simulation}}(\phi))^2 d\phi$$  \hspace{1cm} (2.23)

Experience has shown results similar to figure 2.19. There is no unique combination of $\alpha$ and $\beta$ where an absolute minimum standard deviation is reached. There is a line on which the results are comparably good. This simplifies the optimization process considerably. One shape parameter can have a fixed value, after which the other shape parameter is optimized to find the minimum standard deviation.

![Figure 2.19: standard deviation for different combinations of $\alpha$ and $\beta$ [PENN02]](image)

![Figure 2.20: simulated combustion torque](image)
Figure 2.20 gives an example of a combustion torque simulated with the Isermann/Pennec method. Here $\alpha$ was fixed, after which $\beta$ was optimized to find the best fit.

The Isermann/Pennec empirical model makes it possible to accurately describe the torque resulting from combustion, using only three parameters: static torque, angle of maximum combustion torque and beta.

\[
T_{\text{combustion}}(\varphi) = T_{\text{combustion}}(\varphi, T_{\text{static}}, \varphi_{\text{max}}, \beta) \tag{2.24}
\]

### 2.6 Total torque resulting from gas forces

The total crankshaft torque resulting from gas forces is the sum of compression/expansion torque and combustion torque. This dynamic torque can now be accurately simulated using only geometry data and six parameter values:

\[
T_{\text{comp/exp}} = T_{\text{comp/exp}}(\text{geometry}, cf, P_{\text{intake}}, P_{\text{exhaust}}) \tag{2.25}
\]

\[
T_{\text{combustion}} = T_{\text{combustion}}(T_{\text{static}}, \varphi_{\text{max}}, \beta) \tag{2.26}
\]

\[
T_{\text{gas,sim}} = T_{\text{gas,sim}}(\text{geometry}, T_{\text{static}}, P_{\text{intake}}, P_{\text{exhaust}}, cf, \varphi_{\text{max}}, \beta) \tag{2.27}
\]

A result of this method is displayed in figure 2.21. A comparison between measurement and simulation is shown.

![Figure 2.21: dynamic torque form measurement and simulation](image)
2.7 Parameter maps

The previous analysis is based on only one single operating point. The car manufacturer delivers pressure measurements in many operating points. For each pressure measurement, the torque resulting from gas forces is simulated using the Isermann/Penec method. The engine speed at which a cylinder pressure measurement is conducted is known. A value for the static torque results from the pressure analysis. Note that this is the static torque of only one cylinder. Multiplying this value with the number of cylinders delivers the total static engine torque.

\[ T_{\text{static,engine}} = T_{\text{static,cylinder}} \cdot N_{\text{cyl}} \]  \hspace{2cm} (2.29)

The position in the engine map, at which the pressure measurement is conducted, is now known. Five additional parameters belong to this particular operating point. Because pressure measurements at many different operating points are available, it is possible to create maps for each of these parameters. These maps describe parameter variation with engine speed and static engine torque:

\[ P_{\text{intake}} = P_{\text{intake}}(\alpha T_{\text{static}}) \]
\[ P_{\text{exhaust}} = P_{\text{exhaust}}(\alpha T_{\text{static}}) \]
\[ cf = cf(\alpha T_{\text{static}}) \] \hspace{2cm} (2.30)
\[ \phi_{\text{max}} = \phi_{\text{max}}(\alpha T_{\text{static}}) \]
\[ \beta = \beta(\alpha T_{\text{static}}) \]

With these five parameter maps (and geometry data), it is possible to simulate the dynamic engine torque in every operating point of the engine. An example will be used to show how the maps are constructed.
The next figure shows the engine map of a 2.0L gasoline engine. In this particular case the car manufacturer has delivered cylinder pressure data in over 220 operating points. For each of these measurements the dynamic torque is simulated, using the method described before. This results in the following engine map:

Figure 2.22: engine map, operating points in which pressure data is available

It can be seen that pressure measurements have been conducted at idle, coast, part-load and full-load operation. Five parameters, needed to reconstruct the dynamic engine torque, are found for each of these operating points. These parameters can be displayed as a function of engine speed and static engine torque. For this particular engine the following results are found.

Figure 2.23: intake pressure
Figure 2.24: exhaust pressure
These loose values need to be converted to maps, covering all possible engine operating points. Before these maps can be created, possible “holes” in the dataset need to be taken into account. The operating points at which pressure data is delivered are different for each engine. The example used above shows an ideal case, with over 220 operating points covering the engine map. Very often however, this is not the case. Some common problems, caused by a limited amount of measurements, are as follows:

- no cylinder pressure measurements at the start/idle region (300 - 750 rpm)
- missing cylinder pressure data at coast operation
- no, or little, cylinder pressure data at part-load operation
- missing cylinder pressure data at high engine speeds

Each of these cases has to be taken into account. This will be explained later on by using a second (worst-case) example. First the example of figure 2.22 is treated. Here only data at the start/idle region is missing. In this region a number of points are added, based on values of experience. Secondly, a line of
points at maximum engine torque is added. This is done to create a square engine map. The reason for this has to do with the DyFaSim software (the DyFaSim combustion engine module requires a square mesh for each parameter). The parameter values stay constant at torques higher than the full-load torque curve. After these additions, a linear interpolation along every load line is carried out. Interpolated points are added with a 50-rpm spacing. The results are displayed in the following engine map.

![Engine map](image)

**Figure 2.28**: engine map, after adding points and interpolation

There are now enough operating points to create the full parameter maps. A number of interpolation techniques have been considered. Experience has shown that basic linear interpolation between the operating points delivers the best results. Higher order techniques (cubic, spline) cause problems when the number of operating points is limited. The five parameter maps are displayed in the following figures.

![Intake pressure map](image)

**Figure 2.29**: intake pressure map

![Exhaust pressure map](image)

**Figure 2.30**: exhaust pressure map
These maps are generated using an engine speed spacing of 50 rpm and a static torque spacing of 2.5 Nm.
Now a second example is treated. It is used to illustrate some other techniques to “fill the map”. The previous example was a best-case scenario, with over 220 operating points available. Now a worst-case scenario is considered. Another 2.0L gasoline engine is considered. For this engine very little pressure data is delivered. Only full-load pressure measurements have been conducted, as can be seen in the next figures.

- Figure 2.34: engine map
- Figure 2.35: intake pressure
- Figure 2.36: exhaust pressure
- Figure 2.37: combustion angle
- Figure 2.38: correction factor
In this case more actions are needed to fill the engine map. Again operating points in the start/idle region are added, as well as the line at maximum torque. Furthermore additional points at coast operation are added. Parameter values at these added points are again based on values of experience. Next a linear interpolation along the load lines is made, after which the parameter maps can be created. The results are shown in the next figures.
Figure 2.41: intake pressure map

Figure 2.42: exhaust pressure map

Figure 2.43: combustion engine map

Figure 2.44: correction factor map

Figure 2.45: beta map
These results show that, although only limited data is available, parameter values are found for all engine operating points. This way a good estimate for the dynamic engine torque, based on these interpolated parameter values can be made.

### 2.8 Reducing the parameter maps

The current parameter maps are very accurate, since they are created with a fine mesh:

- speed spacing = 50 rpm
- static torque spacing = 2.5 Nm

This results in a very large matrix for each parameter. Since the DyFaSim combustion engine module is not designed for matrices of this size, the amount of data will have to be reduced. A new spacing for engine speed and static torque can be defined, after which the matrices are reduced in size using this new spacing. In case a large spacing is chosen, accuracy of the engine model will be lost. A suitable compromise between data reduction and loss of accuracy will have to be made. Experience has shown that the following spacing delivers satisfying results:

- speed spacing = 250 rpm
- static torque spacing = 25 Nm

Original points, lying in between, are erased from the matrix. By implementing this new spacing, only 2% of the original amount of data remains.

To visually check the quality of this reduction process, both the full parameter maps and the reduced parameter maps are displayed. An example is given in the following figures. This reduction process belongs to the first example of the previous paragraph. In the MatLab program the axes of both figures are linked. This makes it possible to rotate the figures simultaneously.

Figure 2.46: reduction of the intake pressure map
Figure 2.47: reduction of the exhaust pressure map

Figure 2.48: reduction of the combustion angle map

Figure 2.49: reduction of the correction factor map
Experience has shown that a good visual quality check is possible this way. In case the reduced maps are unsatisfying, resulting from an excessively large spacing, a new (smaller) spacing can be defined.

2.9 Peak-to-peak comparison

Although the visual quality check can be very useful, a more quantitative method is developed. A comparison will be made between the measured dynamic torque (resulting from pressure measurements) and the simulated dynamic torque. For each operating point in which measurement data is available, the dynamic torque is simulated using the full as well as the reduced parameter maps.

Now a comparison between the peak-to-peak torques is made. The peak-to-peak torque is the difference between the minimum and maximum of the dynamic torque curve over two crankshaft revolutions. Here the dynamic torque of the complete engine is considered (figure 2.52). The total engine torque can be found by cascading the one-cylinder torque into a full engine. For an n-cylinder in-line engine the total engine torque is given by:

\[
T_{\text{engine}} = \sum_{i=0}^{n_{\text{cy}l}-1} T_{\text{cy}l} \left( \varphi + i \times \frac{4\pi}{n_{\text{cy}l}} \right)
\]

(2.31)

The next figures show the cylinder cascading for a 4-cylinder engine. The dynamic torques from measurements and simulation are displayed in one graph. Both simulation with the full maps and simulation with the reduced maps are displayed.
The peak-to-peak torque for these three curves is given by:

\[
\begin{align*}
T_{\text{pp,measured}} &= \max(T_{\text{eng,measured}}) - \min(T_{\text{eng,measured}}) \\
T_{\text{pp,sim,full}} &= \max(T_{\text{eng,sim,full}}) - \min(T_{\text{eng,sim,full}}) \\
T_{\text{pp,sim,reduced}} &= \max(T_{\text{eng,sim,reduced}}) - \min(T_{\text{eng,sim,reduced}})
\end{align*}
\]  

(2.32)
For the 220 operating points of the previous example, the 1-cylinder results are displayed in figure 2.53.

![Figure 2.53: dynamic torque of one cylinder, all available operating points](image1.png)

The full (4-cylinder) engine torque is displayed in figure 2.54. From these results, the peak-to-peak torques can be found.

![Figure 2.54: dynamic torque of full engine, all available operating points](image2.png)
The peak-to-peak values are determined and plotted as a function of engine speed and static engine torque. The graphs below show a three-dimensional view of the results, along with the two-dimensional side views.

Figure 2.55: absolute peak-to-peak values

Figure 2.56: absolute peak-to-peak values, side view 1

Figure 2.57: absolute peak-to-peak values, side view 2

Figure 2.57 displays the peak-to-peak torque as a function of static engine torque. A clear linear relation can be seen. This result is expected for all naturally aspirated engines.
Based on these absolute values, it is difficult to evaluate the quality of the simulated curves. Therefore the relative peak-to-peak differences between measurement and both simulations are calculated:

\[
\text{diff}_{\text{relative, sim, full}} = \left( \frac{\text{abs}(T_{\text{pp, sim, full}} - T_{\text{pp, measured}})}{T_{\text{pp, measured}}} \right) \cdot 100\% \tag{2.33}
\]

\[
\text{diff}_{\text{relative, sim, reduced}} = \left( \frac{\text{abs}(T_{\text{pp, sim, reduced}} - T_{\text{pp, measured}})}{T_{\text{pp, measured}}} \right) \cdot 100\% \tag{2.34}
\]

The results are displayed in the following graphs. The red points represent the relative difference between measurement and simulation with the accurate parameter maps. The green points represent the relative difference between measurement and simulation with the reduced maps.

Figure 2.58: relative peak-to-peak difference

Figure 2.59: relative peak-to-peak difference, side view 1

Figure 2.60: relative peak-to-peak difference, side view 2
The simulation with the accurate maps is clearly better than the simulation with the reduced maps. At high engine torques, this loss of accuracy is less obvious. Furthermore it can be seen that the results at coast are very bad. There is no combustion torque at coast operation, only compression/expansion torque. The cylinder pressure at coast cannot be described accurately by the method explained in chapter 2.4.

These relative differences are based on dynamic torque resulting from gas forces only. If the mass torque is also taken into account the absolute peak-to-peak values change, which has an effect on the relative error size. The next graphs show the results in case mass torque is included.

**Figure 2.61**: absolute peak-to-peak values, mass torque taken into account

**Figure 2.62**: absolute peak-to-peak values, side view 1

**Figure 2.63**: absolute peak-to-peak values, side view 2
In figure 2.62 the effect of the mass torque can be seen. At low engine speeds the combustion torque is dominant and at high engine speeds the mass torque is dominant. This occurs because combustion torque and mass torque are phase shifted and the mass torque increases quadratically with engine speed. Comparing figure 2.62 and figure 2.56 clearly shows the effect of the mass torque.

Again the relative differences are calculated, the results are displayed below.

Now all relative differences are smaller than 10%. The large error at coast operation has also disappeared. Since mass torque is very dominant at coast, the bad accuracy of the gas torque simulation is of minimum importance.
2.10 Output for DyFaSim software

If the peak-to-peak analysis is satisfactory, the created parameter maps need to be stored in a format suitable for the DyFaSim software. Therefore, an output file is created in ASCII-format, containing the reduced parameter maps along with the geometry data of the engine. This file can be uploaded into the DyFaSim combustion engine module, after which the engine module is ready for use.

The figure below shows the main window of the DyFaSim combustion engine module. In this case a generated ASCII file of a 6-cylinder engine has been uploaded.

![Figure 2.67: DyFaSim combustion engine module](image)

For each cylinder, the geometry data and parameter maps are defined (inside the red square). With these values, DyFaSim is able to simulate the dynamic engine torque in every operating point.
The figures below show the windows containing the engine data. Geometry data is collected in figure 2.68. The five parameter maps are defined in figure 2.69. Each “edit” button opens a parameter map. An example is shown in figure 2.70.

Figure 2.68: geometry data

Figure 2.69: parameter maps

Figure 2.70 shows an intake pressure map in DyFaSim. In this case it is a map with a 250-rpm speed spacing and a 50-Nm torque spacing. In the bottom left corner the intake pressure matrix is shown. On the right side a two-dimensional graph of this matrix is displayed.

Figure 2.70: intake pressure map
3. Generation of standard engine models

Although pressure-based engine models deliver very accurate results, cylinder pressure data is not always available. In many cases only basic engine data is known. For these situations a second operating mode has been developed: standard engine model generation.

This operating mode needs only basic engine data to create a full model. The model-input consists of:

- engine geometry data
- net torque curve at full load
- drag torque curve
- intake pressure curve at full load

3.1 Combustion engine data sheet

LuK has developed a standardised combustion engine data sheet. This Excel data sheet is filled out by the car manufacturer. An example of the data sheet is given in figure 3.1.
Since the data sheet has a standard layout, it is possible to extract information in an automated way. MatLab creates an interface with the Excel data sheet and imports the required data. The car manufacturer often returns a data sheet in which data is missing. In this case the missing data is replaced with experience values based on a similar engine. This process will be explained in the next paragraph.

3.2 Replacing missing data by estimates

As stated in the beginning of this chapter, there are four essential inputs. In case any of these inputs are unavailable, they need to be replaced by a good estimate. How these estimates are determined will be explained in this paragraph.

3.2.1 Geometry data

In case geometry data is unavailable, the user has to define:
- engine type (gasoline or diesel)
- displacement volume
- number of cylinders

All other geometry data is replaced by values of experience. These experience values are different for gasoline and diesel engines, as can be seen in the tables below.

<table>
<thead>
<tr>
<th>Gasoline</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal displacement</td>
<td>User defined</td>
<td>[liter]</td>
</tr>
<tr>
<td>Number of cylinders</td>
<td>User defined</td>
<td>[-]</td>
</tr>
<tr>
<td>Bore</td>
<td>0,080</td>
<td>[m]</td>
</tr>
<tr>
<td>Stroke</td>
<td>0,100</td>
<td>[m]</td>
</tr>
<tr>
<td>Compression ratio</td>
<td>10,5</td>
<td>[-]</td>
</tr>
<tr>
<td>Connecting rod length</td>
<td>0,150</td>
<td>[m]</td>
</tr>
<tr>
<td>Oscillating mass</td>
<td>0,5</td>
<td>[kg]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Diesel</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal displacement</td>
<td>User defined</td>
<td>[liter]</td>
</tr>
<tr>
<td>Number of cylinders</td>
<td>User defined</td>
<td>[-]</td>
</tr>
<tr>
<td>Bore</td>
<td>0,080</td>
<td>[m]</td>
</tr>
<tr>
<td>Stroke</td>
<td>0,100</td>
<td>[m]</td>
</tr>
<tr>
<td>Compression ratio</td>
<td>18</td>
<td>[-]</td>
</tr>
<tr>
<td>Connecting rod length</td>
<td>0,150</td>
<td>[m]</td>
</tr>
<tr>
<td>Oscillating mass</td>
<td>1,0</td>
<td>[kg]</td>
</tr>
</tbody>
</table>

3.2.2 Net torque curve at full load

In case the net torque curve at full load is unavailable, the user needs to define a maximum value for the net torque. A predefined torque curve is first scaled up or down, depending on the displacement volume. Secondly this scaled curve will be adapted to match the user-defined maximum.

Net torque curves are predefined for a four different situations:
- gasoline, naturally aspirated
- gasoline, turbocharged
- diesel, naturally aspirated
- diesel, turbocharged
The four predefined curves are displayed in figure 3.2.

These are experience curves for 2.0L engines of different configurations. Depending on the chosen displacement volume these curves are scaled up or down:

\[ T_{net, scaled}(\omega) = T_{net, predefined}(\omega) \cdot \frac{V_d}{2} \]  \hspace{1cm} (3.1)

This scaled curve is then compared to the user-defined maximum torque. In case the scaled torque curve exceeds the user-defined maximum, all values above this maximum are cut off:

\[ \max(T_{net, scaled}(\omega)) \equiv T_{max} \]  \hspace{1cm} (3.2)

In case the scaled torque curve lies below the user-defined maximum it is scaled up once again:

\[ T_{net, scaled}(\omega) = T_{net, scaled}(\omega) \cdot \frac{T_{max}}{\max(T_{net, scaled}(\omega))} \]  \hspace{1cm} (3.3)

This method has proven to deliver a good estimate for the net torque curve.

### 3.2.3 Drag torque curve

The car manufacturer delivers net torque data. However the combustion engine model is based on gross torque. Therefore drag torque has to be taken into consideration.

In case no drag torque data is available a similar process as for the net torque curve is used. Again, predefined curves form the basis. For the drag torque there is one curve for a 2.0L diesel engine and one curve for a 2.0L gasoline engine (figure 3.3).
3.2.4 Intake pressure curve at full load

The intake pressure at full load is also predefined for four different situations. Again the division is made between diesel and gasoline, turbocharged and naturally aspirated. In case the engine is naturally aspirated, the intake pressure will be equal to the surrounding atmospheric pressure, approximately 1 bar. For turbocharged engines, predefined curves are used. The curves for turbocharged gasoline and diesel engines are shown in figure 3.4.
4. **Graphical User Interface in MatLab**

Both methods described in the previous chapters have been implemented in MatLab. The main goal of the program is easy operation. This has been accomplished by using the Graphical User Interface tools available in MatLab. For both the pressure-based and the standard engine model generation an example will be given.

A compiled version of the program is available. This version can be installed and used on every computer within LuK, without the need for a MatLab license. Because MatLab licenses are expensive and scarce, a stand-alone version created with MatLab compiler has many advantages.

### 4.1 Graphical User Interface, pressure-based engine model

To explain the graphical user interface for the pressure-based operating mode, an example is given. A DyFaSim engine model is needed for a new engine. The manufacturer has delivered the geometry data and has conducted cylinder pressure measurements.

When the program is started, the user has the possibility to select a language:

![Figure 4.1: language selection](image)

After selecting a language, the user can choose between the two operating modes. Since cylinder pressure data is available, the pressure-based model generation is chosen.

![Figure 4.2: selecting an operating mode](image)
MatLab now shows the database in which the geometry data of all engines is stored. To protect customer data, a nameless example-engine is chosen.

![Image](image1.png)

**Figure 4.3: manufacturer / engine selection**

The engine named “example.txt” is selected. A new window opens in which all available cylinder pressure data for this engine is shown. There is cylinder pressure available at idle, 6 part-loads and full-load.

![Image](image2.png)

**Figure 4.4: pressure data selection**

In this case all pressure data is selected. Of course it is also possible to select less data.
An overview of the selected engine and pressure data is displayed in two info boxes.

![Info box showing selected engine](image1)

![Info box showing selected pressure data](image2)

**Figure 4.5:** info box showing selected engine  
**Figure 4.6:** info box showing selected pressure data

The program asks the user whether pressure data at coast is available. If coast data is not available, which is the case for this example, a dialog box appears. In it are predefined experience values for the intake pressure and exhaust pressure at coast. The user has the possibility to adapt these values, in case this is necessary.

![Question dialog, coast data](image3)

![Dialog box for coast values](image4)

**Figure 4.7:** question dialog, coast data  
**Figure 4.8:** dialog box for coast values

At this point the program is ready to start the calculation of the parameter values for all selected cylinder pressure measurements. The user has the possibility to view the results of the calculation process for every selected pressure measurement. Because the plot actions slow the program down, one can also choose not to view these initial results.

![Question dialog, plot actions](image5)

**Figure 4.9:** question dialog, plot actions
If “Yes” is chosen, the following figure appears. On the left side the dynamic torque from measurement and simulation is shown. These curves are continuously updated during the calculation process. On the right side, the engine map shows the (black) operating points that have already been calculated. The purple point corresponds with the current dynamic torque shown on the left.

![Figure 4.10: dynamic torque plot, along with position in engine map](image)

When the calculation process is completed, the user can adapt the simulated dynamic idle torque. Pressure measurements in this operating point are often unreliable. Therefore the simulated static torque at idle and a value known from experience are shown. The user can decide which value should be used.

![Figure 4.11: possibility to adapt idle simulation](image)

The parameter maps can now be calculated. A dialog box appears in which the user can insert values for the speed and torque spacing. These values will be used to generate the reduced maps.
The final results are displayed using six figures, i.e. one for the engine map and five for the parameter maps. The user now has the ability to visually evaluate the results. The parameter graphs show the accurate and the reduced maps. The axes of both graphs are linked, so they can be turned simultaneously.

Figure 4.12: dialog box for spacing values

Figure 4.13: graphs showing engine map and parameter maps
The program continues after pressing a random key on the keyboard. If the reduction process is unsatisfying, due to large spacing values, new spacing values can be entered.

If the user is satisfied with the results, the peak-to-peak analysis is made. The results are displayed in six figures.

Figure 4.14: possibility to redefine spacing values

Figure 4.15: single cylinder dynamic torque

Figure 4.16: dynamic torque of full engine

Figure 4.17: absolute peak-to-peak values, only gas torque

Figure 4.18: relative difference, only gas torque
After viewing the results of the peak-to-peak analysis, a DyFaSim engine model is created. The user can select a location to save the final model.

![Figure 4.19: absolute peak-to-peak values, gas & mass torque](image1)

![Figure 4.20: relative difference, gas & mass torque](image2)

Finally there is a possibility to save a 1-cylinder representation of the engine model. These single cylinder models are used in case crankshaft dynamics are taken into account. In this case each piston is a separate mass, on which a 1-cylinder combustion engine is mounted.

![Figure 4.21: select a location to save the DyFaSim engine model](image3)

![Figure 4.22: possibility to create a 1-cylinder engine model](image4)
After saving the DyFaSim model(s), the operation is completed. The following window appears:

![Operation completed notification](image)

**Figure 4.23:** info box, informing user that the operation is completed

There is a possibility to make a print-out of the final results. For this purpose, all of the generated maps are displayed in one figure.

![Final results overview](image)

**Figure 4.24:** overview of results, possibility to make a print-out
4.2 Graphical User Interface, standard engine model

To show the graphical user interface for the standard operating mode, the following example is used: an engine model is needed for a 2.0L 4-cylinder diesel engine. For this engine no cylinder pressure data is available. The only known data is:

- geometry data
- torque curve at full load
- drag torque curve
- intake pressure at full load

The data is collected in an Excel data sheet (this standard data sheet is filled out by the car manufacturer). When the program is started, the user has the possibility to select a language:

![Figure 4.25: language selection](image)

Now the user can choose between the two operating modes. Since no cylinder pressure data is available, the standard model generation is needed.

![Figure 4.26: selecting an operating mode](image)

Now the user can define whether geometry data is available for this engine. In case no geometry data is available, estimated values are used.

![Figure 4.27: dialog box, geometry data](image)
For this example geometry data is available. The user can now select the data sheet in which the geometry data is collected. MatLab opens the data sheet, reads the data and displays it in a dialog box. The user now has the ability to check and/or change the displayed data.

The next question is related to the air supply. In this example a turbocharger is present.
The program asks the user whether full load data is available. This is the case, so the data can be extracted from the data sheet. In case no full load data is available, a predefined full load curve is scaled up or down.

![Figure 4.31: question dialog, full load characteristic](image1)

The drag torque is treated in a similar way. In case no data is available, a predefined drag torque curve is scaled up or down, depending on the displacement volume. For this example drag torque data is available in the data sheet.

![Figure 4.32: question dialog, drag torque](image2)

When no intake pressure data (at full load) is available, predefined values of a similar engine are used. In this case the intake pressure characteristic is filled out in the data sheet.

![Figure 4.33: question dialog, intake pressure](image3)

The last question is related to the behavior of the throttle valve at engine shutdown (stop). This information is needed to estimate the parameter maps in the start/stop region.

![Figure 4.34: question dialog, throttle valve](image4)
Before creating the script file for DyFaSim, the resulting engine map and intake pressure map are shown (figure 4.35). On the left side the net and gross (full-load) torque curves are shown, along with the drag torque curve. Of course the drag torque is the difference between the first two. The power output is also shown, scaled on the right Y-axis.

![Figure 4.35: resulting engine map and intake pressure map](image)

After pressing a random key on the keyboard the program continues. The DyFaSim script file is generated, after which the user can select a location to store it.

![Figure 4.36: select a location to save the DyFaSim engine](image)
Again there is a possibility to create a 1-cylinder representation of the engine.

![Image](image1.png)

**Figure 4.37:** possibility to create a 1-cylinder engine model

After saving the DyFaSim model(s), the operation is completed. The next window appears, after which the program automatically shuts down.

![Image](image2.png)

**Figure 4.38:** info box, informing user that the operation is completed
5. **Comparison between pressure-based and standard models**

To investigate the differences between pressure-based and standard engine models, a harmonic analysis will be made. The harmonic analysis will be based on the full-load torque, since this is the most critical operating point.

Each engine model is loaded into DyFaSim and mounted on a large rotational mass. A simulation is performed, during which the mass is accelerated with a large constant torque. Upon this large constant torque, the dynamic full-load engine torque is superimposed. The applied torque accelerates the mass from 1000 rpm until maximum engine speed. A harmonic analysis of the torque signal is made. The torque amplitude of the main excitation order is compared. For 4-cylinder engine this is the second order (two excitations per revolution). For a 6-cylinder engine this is the third order (three excitations per revolution).

The comparison is made for two very different engines:
- 2.0 L, 4-cylinder diesel, turbocharged
- 3.0 L, 6-cylinder gasoline, naturally aspirated

For both engines a pressure-based model and a standard model has been generated. The resulting DyFaSim engine models are used to perform the harmonic analysis.

Figure 5.1 shows the result for the 4-cylinder (turbocharged) diesel engine. Here the second order torque amplitude is compared.

![Harmonic analysis, 2.0L diesel 4-cylinder (turbocharged)](image-url)
Figure 5.2 shows the result for the 6-cylinder (naturally aspirated) gasoline engine. Here the third order torque amplitude is compared.

In both graphs the effect of the mass torque can clearly be recognised. At low engine speeds the combustion torque is dominant. Since mass torque and combustion torque are phase shifted and mass torque increases quadratically with engine speed, both components cancel each other out at a certain speed. At higher engine speeds the mass torque is dominant, causing the combined amplitude to increase again.

These results show a clear similarity between the pressure-based and standard engine models. At part-load the differences will probably be larger, since the standard model is largely an estimate there. For the most critical (and therefore the most important) full-load operation, the results of the standard model are acceptable.
6. Idle drive train model in MatLab / Simulink

The goal of this second assignment is to build an idle drive train model in Simulink. This model will be used to make special simulations. In case complicated engine management components need to be simulated, this cannot be done immediately in the DyFaSim software. In Simulink it is easier and faster to build and test new components. Furthermore Simulink is the industry standard, making it possible to exchange models with the customer.

The main goal is to construct an idle model that delivers the same results as its DyFaSim counterpart. This model will then be used to investigate the unwanted phenomenon of sub-harmonic vibrations at idle operation. Experience has shown that cylinder-balancing controllers can contribute to sub-harmonic vibrations of the dual-mass flywheel. It is much more practical to build and test a cylinder-balancing controller in Simulink, than it is in DyFaSim.

6.1 Virtual test bench for Simulink representation of LuK arc springs

Before considering the complete drive train model, a new Simulink representation of the LuK arc springs needs to be tested. A virtual test bench has been built in Simulink (figure 6.1). The test bench contains two masses (primary and secondary) with an arc spring set in between.

![Virtual test bench diagram]

MatLab Simulink S-function Representation of LuK DMF Arc Springs

Warning: model contains no basic friction, no friction control plate, no flange inertia & no inner damper

Created by Dr. Stephen Jones, Dr. Mathias Müller & Luc Römers, XB Department, LuK GmbH

Version 2.2 - Beta

Figure 6.1: virtual test bench containing Simulink arc spring representation
The subsystem containing the arc springs is shown in figure 6.2. This particular dual-mass flywheel contains two inner arc springs and two outer arc springs.

![Diagram of arc springs](image)

Figure 6.2: S-functions describing arc springs

An increasing positive torque is applied on the primary mass (see figure 6.1), representing the engine torque. Simultaneously an increasing negative torque is applied on the secondary mass, representing load torque. This way the dual-mass flywheel is twisted, thereby compressing the arc springs. Having reached a maximum value, the torque on both sides is decreased again.

After simulation the overall arc spring reaction torque can be plotted against the relative angle. This results in the combined arc spring characteristic. One static simulation and multiple dynamic simulations are performed. This is done because the characteristic is speed dependent.

The same test bench has been built in DyFaSim, making it possible to compare results. Figure 6.3 and 6.4 show the Simulink and DyFaSim results. It can be seen that the results of both models are 99% the same. Having validated this new Simulink arc spring model, it can be implemented into the model of the complete drive train.
Figure 6.3: overall arc spring characteristic, Simulink (left) & DyFaSim (right)

Figure 6.4: overall arc spring characteristic, DyFaSim & Simulink
6.2 Drive train model for idle simulation

The drive train model described in this paragraph consists of a combustion engine, idle drive train and an engine management module. Figure 6.5 shows the top layer of the model. The following Simulink modules were already developed by LuK:

- combustion engine module (idle only)
- arc spring module

Additional elements have been built to complete the idle drive train:

- flange in dual-mass flywheel
- inner damper in dual-mass flywheel
- basic friction in dual-mass flywheel
- idle speed controller (including speed filter)
- oscillating inertia of the combustion engine

Furthermore the engine module is improved by correcting some small errors.

Figure 6.5: idle drive train model, top layer
Figure 6.6 shows the combustion engine model (orange subsystem in figure 6.5). Four subsystems produce the different torque components. From top to bottom, the components are:

- mass torque
- compression / expansion torque
- combustion torque
- friction torque

For the combustion torque, the Isermann empirical model is used. All torque components are summed, after which the total torque is sent to the drive train (primary mass).

The engine is controlled by the idle speed controller (inside the engine management subsystem). The oscillating primary speed is filtered, resulting in an average value. This average value is fed into a PI-controller, which controls the static value of the combustion torque. For this particular model the P and I-gains are shaped (a function of the error-size).
Figure 6.7 gives an overview of the idle drive train model (yellow subsystem in figure 6.5). For idle operation the drive train can be modeled using three masses:

- crankshaft & primary mass DMF
- flange
- secondary mass DMF

The equation of motion of these three masses is represented by the rectangular subsystems in the figure below. Between the primary mass and the flange, the arc springs are situated (subsystem with DMF picture). The inner damper is mounted between the flange and the secondary mass. The final component is the basic friction working between the primary and secondary mass.

The inertia of the first mass is a function of the crank angle. It consists of a static inertia (belonging to primary DMF mass and crankshaft) and a dynamic inertia (belonging to the pistons).
6.3 Comparison with DyFaSim counterpart

The complete idle drive train model is compared to its DyFaSim counterpart. Both models are given the same initial conditions and are simulated for five seconds. A Runge-Kutta (ode4) fixed-step solver is used.

The speed-signals of the primary and secondary mass are compared. The results are displayed in figure 6.8 until 6.11.

These results show that both models are 99% identical. The remaining small differences are probably caused by the solver methods. Even though Runge-Kutta is used in both programs, the solver techniques are not completely the same.
7. Conclusions

*Automatic generation of combustion engine models using MatLab*

A tool has been created that enables all engineers at the dynamic simulation department of LuK to generate engine models for the DyFaSim simulation software. It forms an automatic interface between manufacturer engine data and the DyFaSim software.

Accurate engine models can be generated based on cylinder pressure data. In case no cylinder pressure data is available, an engine model based on standard engine data can be generated. Both methods have been compared by means of a harmonic analysis. Based on this analysis it can be concluded that the standard model generation also delivers satisfying results.

The MatLab-based tool is available as a stand-alone package, without the need for a MatLab license. The graphical user interface provides easy operation. The current version makes it possible to choose between two languages. Further expansion to three or more languages is a possibility.

Furthermore a guideline has been written, describing the use of the DyFaSim combustion engine module and the new MatLab-tool. A training course will be organized to familiarize all engineers with this new tool.

*Idle drive train model in MatLab / Simulink*

An idle drive train model has been built in MatLab / Simulink. The model contains a combustion engine module, drive train module (including dual-mass flywheel) and an engine management module.

The main goal was to create a model that is similar to its DyFaSim counterpart. Comparison of output signals has shown that this goal has been accomplished. The Simulink model will be used for special simulations, involving complex engine management components. A starter motor could be added to the model, making it possible to do start / stop simulations.
### List of symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Unit</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDC</td>
<td>[-]</td>
<td>Bottom dead centre</td>
</tr>
<tr>
<td>cf</td>
<td>[-]</td>
<td>Correcting factor</td>
</tr>
<tr>
<td>d</td>
<td>[m]</td>
<td>Bore diameter</td>
</tr>
<tr>
<td>diff relative, sim, full</td>
<td>[%]</td>
<td>Relative difference, measurement &amp; full map simulation</td>
</tr>
<tr>
<td>diff relative, sim, reduced</td>
<td>[%]</td>
<td>Relative difference, measurement &amp; reduced map simulation</td>
</tr>
<tr>
<td>$F_{\text{applied}}$</td>
<td>[N]</td>
<td>Force applied on piston</td>
</tr>
<tr>
<td>$F_{c,r}$</td>
<td>[N]</td>
<td>Radial force on crank</td>
</tr>
<tr>
<td>$F_{c,t}$</td>
<td>[N]</td>
<td>Tangential force on crank</td>
</tr>
<tr>
<td>$F_{\text{piston}}$</td>
<td>[N]</td>
<td>Force acting on piston</td>
</tr>
<tr>
<td>l</td>
<td>[m]</td>
<td>Connecting rod length</td>
</tr>
<tr>
<td>$m_{\text{osc}}$</td>
<td>[kg]</td>
<td>Oscillating mass</td>
</tr>
<tr>
<td>n$_{\text{cyl}}$</td>
<td>[-]</td>
<td>Number of cylinders</td>
</tr>
<tr>
<td>P</td>
<td>[Pa]</td>
<td>Cylinder pressure</td>
</tr>
<tr>
<td>$P_{\text{comp/exp}}$</td>
<td>[Pa]</td>
<td>Compression / expansion pressure</td>
</tr>
<tr>
<td>$P_{\text{exhaust}}$</td>
<td>[Pa]</td>
<td>Exhaust pressure</td>
</tr>
<tr>
<td>$P_{\text{intake}}$</td>
<td>[Pa]</td>
<td>Intake pressure</td>
</tr>
<tr>
<td>$P_{\text{surrounding}}$</td>
<td>[Pa]</td>
<td>Surrounding pressure</td>
</tr>
<tr>
<td>r</td>
<td>[m]</td>
<td>Crank radius</td>
</tr>
<tr>
<td>$T_{\text{combustion}}$</td>
<td>[Nm]</td>
<td>Torque due to combustion</td>
</tr>
<tr>
<td>$T_{\text{comp/exp}}$</td>
<td>[Nm]</td>
<td>Torque due to compression / expansion</td>
</tr>
<tr>
<td>$T_{\text{crank}}$</td>
<td>[Nm]</td>
<td>Crankshaft torque</td>
</tr>
<tr>
<td>$T_{\text{drag,scaled}}$</td>
<td>[Nm]</td>
<td>Scaled drag torque curve</td>
</tr>
<tr>
<td>$T_{\text{drag,predefined}}$</td>
<td>[Nm]</td>
<td>Predefined drag torque curve</td>
</tr>
<tr>
<td>$T_{\text{engine}}$</td>
<td>[Nm]</td>
<td>Engine torque</td>
</tr>
<tr>
<td>$T_{\text{gas,sim}}$</td>
<td>[Nm]</td>
<td>Simulated gas torque</td>
</tr>
<tr>
<td>$T_{\text{mass}}$</td>
<td>[Nm]</td>
<td>Torque due to mass forces</td>
</tr>
<tr>
<td>$T_{\text{max}}$</td>
<td>[Nm]</td>
<td>Maximum torque</td>
</tr>
<tr>
<td>$T_{\text{measured}}$</td>
<td>[Nm]</td>
<td>Torque from measurement</td>
</tr>
<tr>
<td>$T_{\text{net,scaled}}$</td>
<td>[Nm]</td>
<td>Scaled net torque curve</td>
</tr>
<tr>
<td>$T_{\text{net,predefined}}$</td>
<td>[Nm]</td>
<td>Predefined net torque curve</td>
</tr>
<tr>
<td>$T_{\text{pp,measured}}$</td>
<td>[Nm]</td>
<td>Peak-to-peak torque, measurement</td>
</tr>
<tr>
<td>$T_{\text{pp,sim,full}}$</td>
<td>[Nm]</td>
<td>Peak-to-peak torque, full map simulation</td>
</tr>
<tr>
<td>$T_{\text{pp,sim,reduced}}$</td>
<td>[Nm]</td>
<td>Peak-to-peak torque, reduced map simulation</td>
</tr>
<tr>
<td>$T_{\text{simulation}}$</td>
<td>[Nm]</td>
<td>Simulated torque</td>
</tr>
<tr>
<td>$T_{\text{static}}$</td>
<td>[Nm]</td>
<td>Static engine torque</td>
</tr>
<tr>
<td>$T_{\text{static,cylinder}}$</td>
<td>[Nm]</td>
<td>Static cylinder torque</td>
</tr>
<tr>
<td>$T_{\text{static,engine}}$</td>
<td>[Nm]</td>
<td>Static engine torque</td>
</tr>
<tr>
<td>TDC</td>
<td>[-]</td>
<td>Top dead centre</td>
</tr>
<tr>
<td>V</td>
<td>[m$^3$]</td>
<td>Cylinder volume</td>
</tr>
<tr>
<td>$V_c$</td>
<td>[m$^3$]</td>
<td>Clearance volume</td>
</tr>
<tr>
<td>$V_d$</td>
<td>[m$^3$]</td>
<td>Displacement volume</td>
</tr>
<tr>
<td>x</td>
<td>[m]</td>
<td>Piston position from top dead centre</td>
</tr>
<tr>
<td>α</td>
<td>[-]</td>
<td>Shape factor</td>
</tr>
<tr>
<td>β</td>
<td>[-]</td>
<td>Shape factor</td>
</tr>
<tr>
<td>Symbol</td>
<td>Unit</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>$\varepsilon$</td>
<td>[-]</td>
<td>Compression ratio</td>
</tr>
<tr>
<td>$\varphi$</td>
<td>[rad]</td>
<td>Crank angle</td>
</tr>
<tr>
<td>$\varphi_{\text{max}}$</td>
<td>[rad]</td>
<td>Angle of maximum combustion torque</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>[-]</td>
<td>Isentropic coefficient</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>[-]</td>
<td>Crank/rod ratio</td>
</tr>
<tr>
<td>$\omega$</td>
<td>[rad/s]</td>
<td>Engine speed</td>
</tr>
</tbody>
</table>
References

[ENCA98] Microsoft Encarta 98, Multimedia Encyclopaedia


