SSA 4

subject

Dolf Eck and Thomas Driessen

February 25, 2021

Goal

- The goal of this SSA is to improve the theoretical Matlab model.

Conclusion

- A different approach of making the model is used which is more accurate than the previous one.
- The non-ideal gas approach is not implemented yet, this will take some more time and effort.

Problems

- Had some difficulty with determining the Wiebe function, and correctly implementing it.

Follow up Steps

- Make the model better, there are still some points of improvement.
  - Temperature dependence of Cv
  - Different fuels
  - determine mass of fuel used.
  - Incorporate work and efficiency
- Look more at non- ideal gasses.

Work Division and time division

- Looking at possibilities for non ideal gasses approach. (Thomas 1.5 hour)
- Already working on the script before meeting together. (Dolf 1.5 hrs)
- Working on the script together (2.5 hour)
- SSA writing (2 hour each person)

Overleaf Link

1 The Improvements on the theoretical matlab script

The first script which was able to plot a theoretical PV-diagram of the Otto cycle was described in such a way to calculate the volume of each step, by the use of ideal gas law and divide this volume in smaller parts to calculate the belonging pressure. It however did not take into account the crank angles, and was hence difficult to adjust with real factors. This worked out, but in the Handbook a 'skeleton script' is provide which introduces an other way of approaching this problem. In the skeleton script the crank angle is taken as the variable on which the model will be calculated, so not only the volume as in our previous model. Also it describes the cycles as one function dependent on the crank angle, and not divided in four different functions dependent on the volume.
So first there will be looked to the skeleton script. This script can be seen below:

```
% Initialisation
p(1)=P0;T(1)=T0;
pad(1)=p(1);
Ca(1)=0.0;V(1)=Vcyl(Ca(1),S,B,l,rc); % Vcyl is a function that
                                     % computes cyl vol as fie of crank-
                                     % angle for given B,S,l and rc
m(1)=p(1)*V(1)/Rg/T(1);
%
%
% Loop over crank-angle, with 'for' construction
NCa=360;                    % Number of crank-angles
dCa=0.5;                    % Stepsize
NSteps=NCa/dCa;
for i=2:NSteps,
  Ca(i)=Ca(i-1)+dCa;
  V(i)=Vcyl(Ca(i),S,B,l,rc);  % New volume for current crank-angle
  m(i)=m(i-1);                % Mass is constant, valves are closed
  dV=V(i)-V(i-1);             % Volume change

  dQcom = YourModel(Ca(i));   % Heat Release by combustion
  dT=(dQcom-p(i-1)*dV)/Cv/m(i-1); % 1st Law dU=dQ-pdV (closed system)
                              % adiabatic closed system with constant
                              % gas composition and constant Cv
  T(i)=T(i-1)+dT;
  p(i)=m(i)*Rg*T(i)/V(i);     % Gaslaw
end;
```

Figure 1: The skeleton script

This script still requires a number of functions which need to be made by us. The primary two are the "Yourmodel", and "Vcyl" function. The latter is a function for describing the volume in terms of the crank angle. The following equation has been set up to do so:

$$V(\theta) = -\frac{V_d}{2}\cos\left(\theta\frac{\pi}{180} + \phi\right) + \frac{V_d}{2} + V_c \tag{1}$$

Where V is the volume at a give crank angle, θ, which is measured in a clockwise rotation from the Y-axis. V_d is the volume swept away by the piston head, and V_c is the clearance volume. Lastly ϕ can be used to describe the starting crank angle and translate the function over. It was chosen to calculate V_d and V_c from the data sheet instead of the measurements taken on the engine. This was done following maths.

$$V_t = V_d + V_c$$

Where V_t is total volume ($196cm^3$) and r the compression ratio, 8.5. R is calculated as followed.

$$R = \frac{V_d + V_c}{V_c} \tag{2}$$

The second function which needs to be worked on is "YourModel", which we re-named "HeatReleased" This uses a lot of the same components from the previous ideal model to describe the

combustion energy released. However it incorporates the Wiebe function to describe the heat release over a period of crank angles see Section 1.1 for full explanation. Using the same ideal scenario for the balance equations the Lower heating value is calculated based on the mass composition before and after combustion (50-52). (Function is included at end of SSA). And this value is then further used with the Wiebe's function described in the hand book.

$$\frac{dQ}{d\theta} = Q_{LHV} \cdot m_{fuel} \cdot \frac{dx_b}{d\theta} = Q_{LHV} \cdot m_{fuel} \cdot n \cdot a \frac{1 - x_b}{\theta_d} \cdot (\frac{\theta - \theta_s}{\theta_d})^(n-1) \tag{3}$$

There are still some things to be implemented in the script such as the fuel ratios, fuel amount (1kg right now), and heat losses, but this is the intermediary result is seen in figure 3. The intake and exhaust gasses are not represented as this was advised by discussion pages. The reason

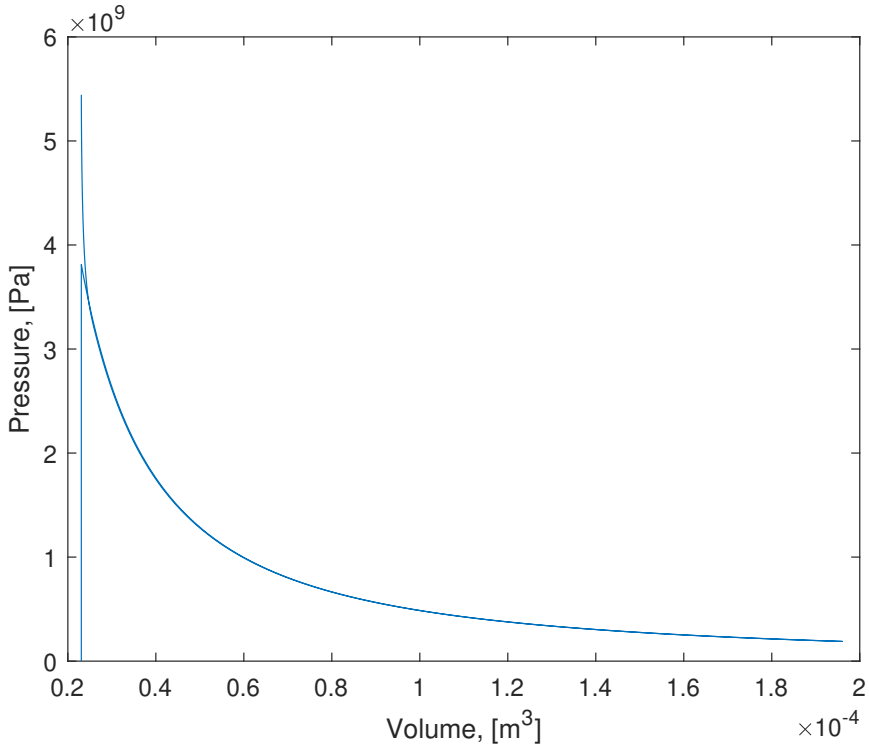
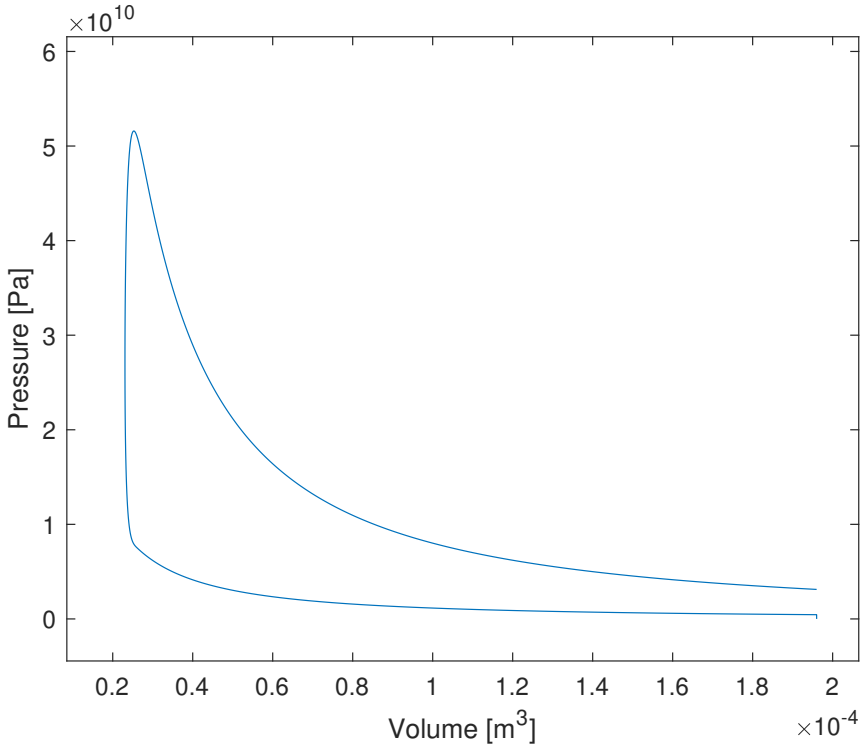


Figure 2: Caption

that this graph (Figure 2) looked so odd is that it is because it starts at a crank angle of zero which would mean that the valves open and gasses would enter. However due to the given advice to not model the gas exchange. Therefore the starting crank angle should be not zero but π. With this small adjustment made the graph looks as followed by shifting the



h!

Figure 3: Caption

1.1 Crank angles

As the thermodynamics of the systems are all a function of the crank angle, some conditions of certain thermodynamics at a specific crank angle need to be defined. Two certain main definitions for the crank angle are the TDC (Top Dead Centre) and the BDC (Bottom Dead Center). In figure 4 the crank angle relative to TDC and BDC at which process occurs can be seen. For the model the most important part is the timimg of ignition. As can be seen in the figure this timings just before the TDC. In general this value is 15 degrees before TDC. Furthermore it is important to know till which value the combustion will take place. This will be important for implementing the Wiebe function. This function will be explained later on in this SSA. It is important to hence the fact that it will take two rounds on the diagram for one full cycle of the engine.

The combustion time in terms of the crank angle will be about 20 degrees after TDC. So the total combustion will be therefor 35 degrees. [1]

Furthermore the valve timing can also be expressed in terms of the crank angle. This can be seen in figure 5. As can be seen the intake valve opens at 21 degrees before TDC and closes at 51 degrees after BDC.

[1] https://en.wikipedia.org/wiki/Ignition_timing

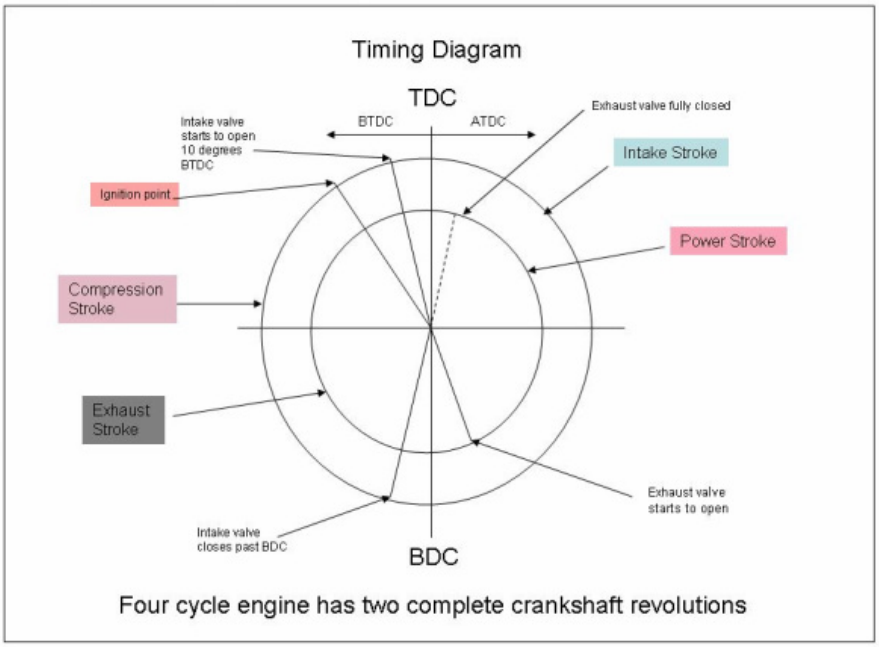


Figure 4: The timing diagram of a four stroke engine [1]

1.2 Wiebe function

The Wiebe function is a function which can be used to describe the amount of fuel which will be combust at a certain crank angle. This function can be seen in equation 5.

$$x_b(\theta) = 1 - e^{-a(\frac{\theta-\theta_s}{\theta_d}^n)} \tag{4}$$

In which θ is the crank angle, θ_s the crank angle at which the ignition starts, θ_d the total crank angle at which the ignition take place. n can be set as 3 and a as 5 by definition. For determining the heat which will be exerted during this combustion as a function of the crank angle the derivative to the angle of the Wiebe will be taken and multiplied by Q_{LHV}.

$$\frac{dQ}{d\theta} = Q_{LHV} \cdot m_{fuel} \cdot n \cdot a \cdot \frac{1 - x_b}{\theta_d} \cdot (\frac{\theta - \theta_s}{\theta_d})^{n-1} \tag{5}$$

1.3 Van der Waals gas

When looking at a method to describe the thermodynamics of a non ideal gas, the van der Waals gas equations can be used. This behaviour in general can be described as in equation **??**. In this equation a and b are the Van der Waals constant.

$$(P + \frac{n^2 a}{V^2})(V - nb) = nRT \tag{6}$$

For an adiabatic process equation 7 holds. Where $\Gamma = \frac{R}{C_v} + 1$

$$(P + \frac{n^2 a}{V^2})(V - nb)^{\Gamma} = constant \tag{7}$$

Some more research for this needs to be done, see [2].

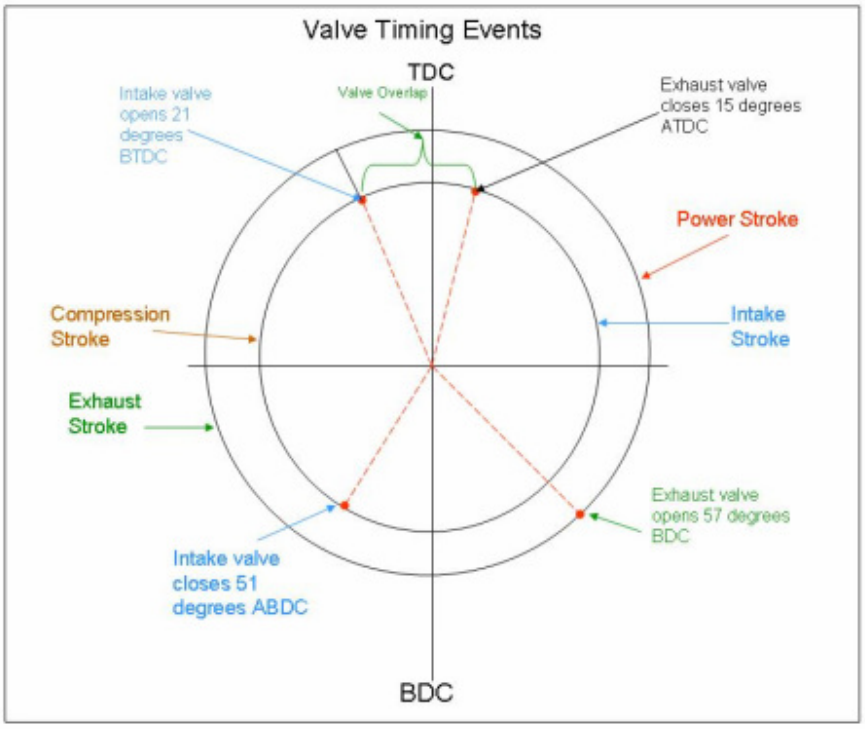


Figure 5: The timing diagram for the valve timing [1]

MATLAB

```
function [q_lhv,Qcomb] = HeatReleased(Ca, AF, mfurate, Yfuel, Yair, Mi, Runiv, ...
    Elements, Tref)

% Qcomb     - Energy released during combustion

mairrate         = mfurate*AF;               % [kg/s]    Air rate coming into the ...
    carb
mrate_in         = mairrate + mfurate;       % {kg/s]    Rate of intake air

%Mass pre-combustion
mO2rate_in               = mairrate*Yair(6);     % [kg/s]
mN2rate_in               = mairrate*Yair(5);     % [kg/s]
mC2H5OHrate_in           = mfurate*Yfuel(1);
mGASOLINErate_in         = mfurate*Yfuel(2);
Ypre_comb                = [mC2H5OHrate_in, mGASOLINErate_in, 0, 0, mN2rate_in, ...
    mO2rate_in]./mrate_in; %mass fraction

% moles in
MoleO2rate_in = mO2rate_in/Mi(6);
MoleN2rate_in = mN2rate_in/Mi(5);
MoleC2H5OHrate_in = mC2H5OHrate_in/Mi(1);
MoleGASOLINErate_in = mGASOLINErate_in/Mi(2);
Molerate_in = MoleO2rate_in + MoleN2rate_in + MoleC2H5OHrate_in + ...
    MoleGASOLINErate_in;
Xpre_comb =[MoleGASOLINErate_in, MoleC2H5OHrate_in, 0, 0, MoleN2rate_in, ...
    MoleO2rate_in]./Molerate_in;
Mpre_comb = Xpre_comb*Mi';

Rpre_comb = Runiv/Mpre_comb';

% after combustion;
%mass out
mH2Orate_out = 3*Mi(3)/Mi(1)*mC2H5OHrate_in + 9*Mi(3)/Mi(2)*mGASOLINErate_in;
```

```
mO2rate_out = mO2rate_in - 3*Mi(6)/Mi(1)*mC2H5OHrate_in - ...
    12.5*Mi(6)/Mi(2)*mGASOLINErate_in;
mCO2rate_out = 2*Mi(4)/Mi(1)*mC2H5OHrate_in + 8*Mi(4)/Mi(2)*mGASOLINErate_in;
mN2rate_out = mN2rate_in;
moutrate = [0 0 mH2Orate_out mCO2rate_out, mN2rate_out, mO2rate_out];
check = sum(moutrate) - mrate_in;
Yafter_comb = moutrate/sum(moutrate);

%moles out
MoleO2rate_out = mO2rate_out/Mi(6);
MoleN2rate_out = mN2rate_out/Mi(5);
MoleCO2rate_out = mCO2rate_out/Mi(4);
MoleH20rate_out = mH2Orate_out/Mi(3);
Molerate_out = MoleO2rate_out + MoleN2rate_out + MoleCO2rate_out + MoleH20rate_out;
Xaft_comb =[0, 0, MoleH20rate_out, MoleCO2rate_out, MoleN2rate_out, ...
    MoleO2rate_out]./Molerate_out;
Maft_comb = Xaft_comb*Mi';
Raft_comb = Runiv/Maft_comb';

for i=1:length(Elements)
    h_comb(i) = HNasa(Tref, Elements(i));
end

hpre_comb = h_comb*Ypre_comb';
hafter_comb = h_comb*Yafter_comb';
q_lhv = hafter_comb - hpre_comb;

n =3;
a = 5;
Theta_d = 35;
Theta_s = (360-15);


if Ca ≥ Theta_s
    Xb = 1 - exp(-a*((Ca-Theta_s)/Theta_d)^n);
    dQcomb_dTheta = q_lhv*mfurate*n*a*(1-Xb)/Theta_d*((Ca-Theta_s)/Theta_d)^(n-1);
    Qcomb = dQcomb_dTheta*1;
elseif Ca < Theta_s
    Qcomb =0;
end
end
```

References

[1] http://www.streetrod101.com/four-stroke-timing-diagram.html#

[2] https://arxiv.org/pdf/1802.01474.pdf

[3]