Computer Methods in Combustion

Project I

Solid rocket motor ignition system based on air-methane mixture

Grzegorz Pela

Szymon Małecki

supervisor

Mateusz Żbikowski Ph. D.

Warsaw, 2019

Contents

1. Introduction

1.1 Cantera

Cantera is an open-source tool for solving problems involving chemical kinetics, thermodynamics, and transport processes. In this project it will be used to simulate a single reactor for air-methane mixtures with various equivalence ratios.

1.2 Purpose of the project

The main goal is to determine the optimal equivalence ratio, amount of fuel (air-methane) mass and time needed to ignite rocket fuel using the outer reactor with fixed mass flow and volume. Requirement which has to be met is to provide to the rocket motor sufficient amount of heat.

2. Review of the literature

To create this project two sources where used. The first one is cantera.org – tutorials, examples and documentation allowed to generate the code of simulation. The second one is the Rocket Division's documentation from where A2 solid rocket motor data were delivered.

3. Model description

Air-methane mixture with specified equivalence ratio (range form 0,6 to 1,6) is delivered by inlet (with 0,025 kg/s mass flow and 10 atm pressure) to the reactor where it burns and is ejected through the exhaust to the rocket motor. From the chemical reactions in combustor the value of heat per second is obtained and then the amount of time and mass of fuel necessary to ignite the solid rocket motor is calculated. The required heat is 7,79 MJ.

4. Results and conclusion

The results plot of time and mass for specified equivalence ratio is shown below.

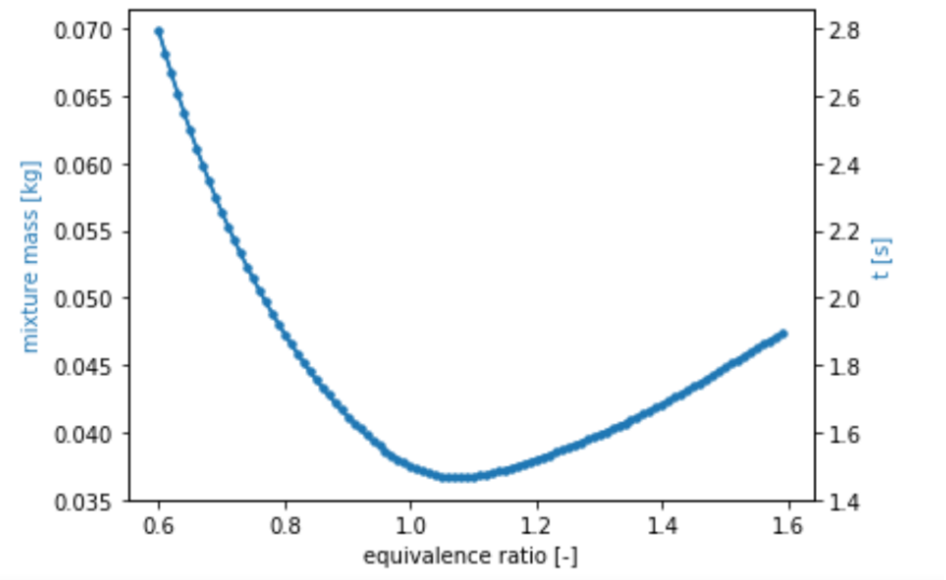

***Figure 1**: Air-methane mass and time needed to ignition of solid rocket motor as a functions of equivalence ratio*

The results show that the smallest amount of mass and time needed to ignite the rocket motor is for the equivalence ratio slightly above 1 (stoichiometry) – about 1,05 and their values are about 0,0365 kg (36,5 g) and 1,44 s. The reason for optimal equivalence ratio above stoichiometry and more steep course of function on lean combustion side is that even though rich combustion is usually less efficient it could produce more power so it is more applicable for this kind of usage. Comparing to ignition system based on solid particles (BKNO3 or black powder) air-methane mixture is rather poor since using BKNO3 1,2 gram is sufficient and ignition delay (time between starting the ignition system and starting of the rocket motor) is negligible (much less than a second).

5. Summary

The model created in Cantera allowed us to establish theoretical values for ignition system design based on air-methane mixture. The optimal equivalence ratio (1,05), the amount of fuel (36,5 g) and time needed to ignition (1,44 s) were determined. However the simulation has also indicated that this kind of ignition system (based on gaseous mixtures) is probably not suitable for this kind of usage. Still the future applications could be considered and optimized to obtain satisfactory results.

6. Code

```
import math as m

import cantera as ct

import numpy as np

import matplotlib.pyplot as plt


# Reaction mechanism GRI-Mech 3.0

gas = ct.Solution('gri30.xml')

# Energy which has to be provided in order to ignite the rocket fuel of A2 rocket motor

ignition_energy = 1.2*1550 # [kcal]

ign_energy = ignition_energy*4186.8 # [J]

# Outer combustion chamber (reactor) for ignition mixture will be used

d = 40.0 # mm diameter of the reactor

l = 100.0 # mm length of the reactor

Vr = m.pi*d*d*0.25*l/1000000000 # [m^3] volume of the reactor

mdot = 0.025 # [kg/s] mass flow in the reactor
```

```
mt = []
Tt = []
Qet = []
Eqt = []
tsim = 0.005 # [s] time spended in the reactor by flowing gases

eq_ratio = 0.6 # initial equivalence ratio
while eq_ratio < 1.6:
    print(eq_ratio)

    # gas definition, initial conditions and inlet
    gas.TP = 300.0, ct.one_atm*10
    gas.set_equivalence_ratio(eq_ratio, 'CH4:1.0', 'O2:1.0, N2:3.76')
    inlet = ct.Reservoir(gas)

    # filling combustor with a gas
    gas.equilibrate('HP')
    combustor = ct.IdealGasReactor(gas)
    combustor.volume = Vr

    # exhaust definition
    exhaust = ct.Reservoir(gas)

    # mass flow
    inlet_mfc = ct.MassFlowController(inlet, combustor, mdot=mdot)

    # simulation definition
    sim = ct.ReactorNet([combustor])
```

```
    # Reactor's states array
    states = ct.SolutionArray(gas)

    #Simulation
    sim.set_initial_time(0.0)  # reset the integrator
    sim.advance(tsim)
    states.append(combustor.thermo.state)
    V = mdot/combustor.density
    Q = -np.sum(states.net_production_rates * states.partial_molar_enthalpies)
    Qe = Q*V
    t = ign_energy/Qe
    mpal = mdot*t
    print('masa = {:.2f}; T = {:.1f};'.format(mpal, combustor.T))

    # writing results to arrays
    mt.append(mpal)
    Tt.append(t)
    Qet.append(Qe)
    Eqt.append(eq_ratio)
    eq_ratio += 0.01
    print('Qe = {:.2f}; mpal = {:.2f}; t = {:.2f}'.format(Qe, mpal, t))
    Q=0.0
    mpal=0.0

#plots
f, ax1 = plt.subplots(1,1)
ax1.plot(Eqt, mt, '.-', color='C0')
ax2 = ax1.twinx()
ax1.set_xlabel('equivalence ratio [-]')
```

```
ax1.set_ylabel('mixture mass [kg]', color='C0')
ax2.plot(Eqt,Tt, '.-', color='C0')
ax2.set_ylabel('t [s]', color='C0')
f.tight_layout()
plt.show()
```