Calibration of a 4-cyl engine model

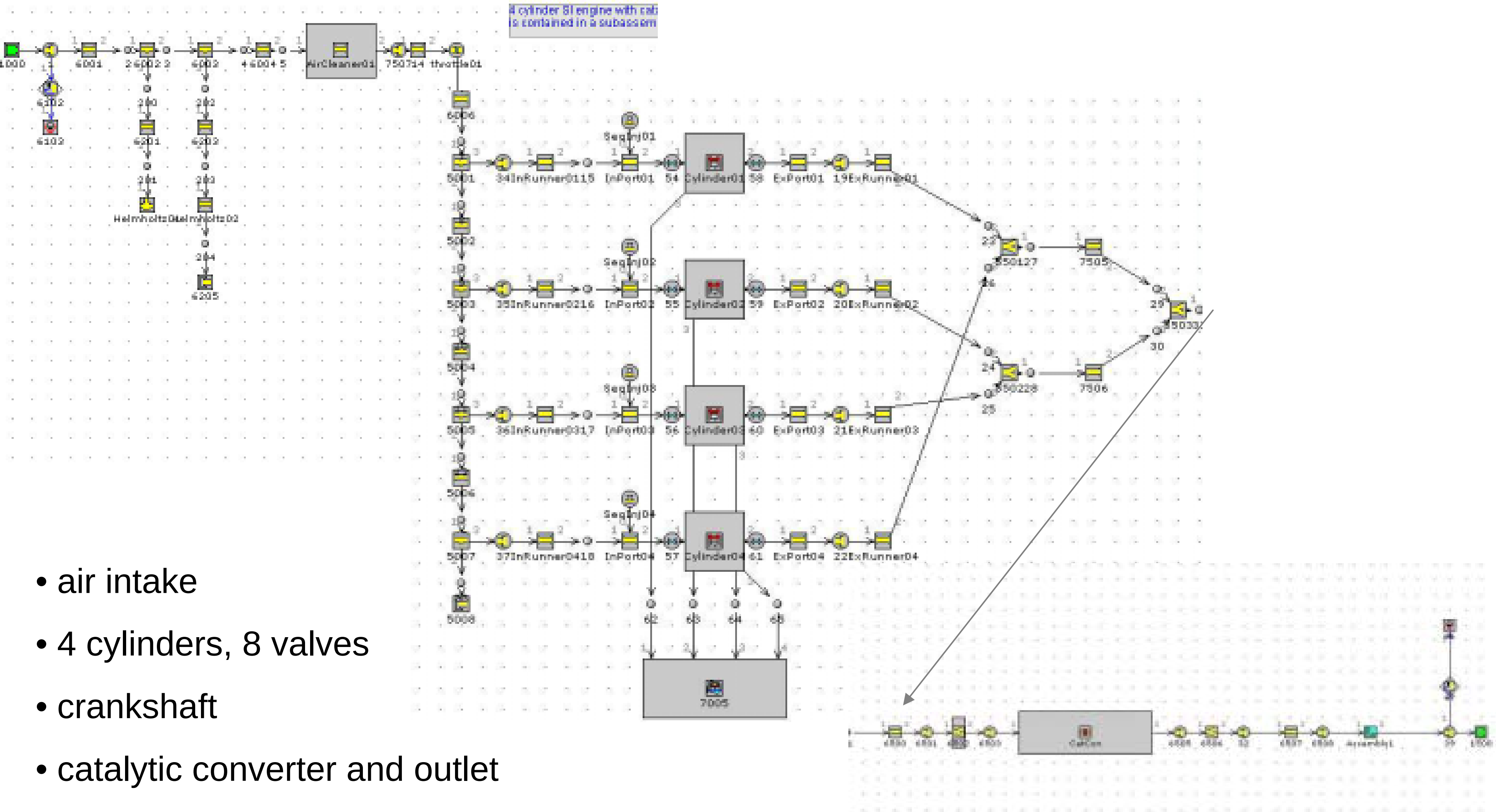
Integration of modeFRONTIER with GTPower

Summary

- **Model definition** (GTPower)
- **Optimisation workflow** (modeFRONTIER)
- **Analysis of results**

modeFRONTIER
a multi-objective optimization and design environment

4-cyl engine: GTPower model



- air intake
- 4 cylinders, 8 valves
- crankshaft
- catalytic converter and outlet

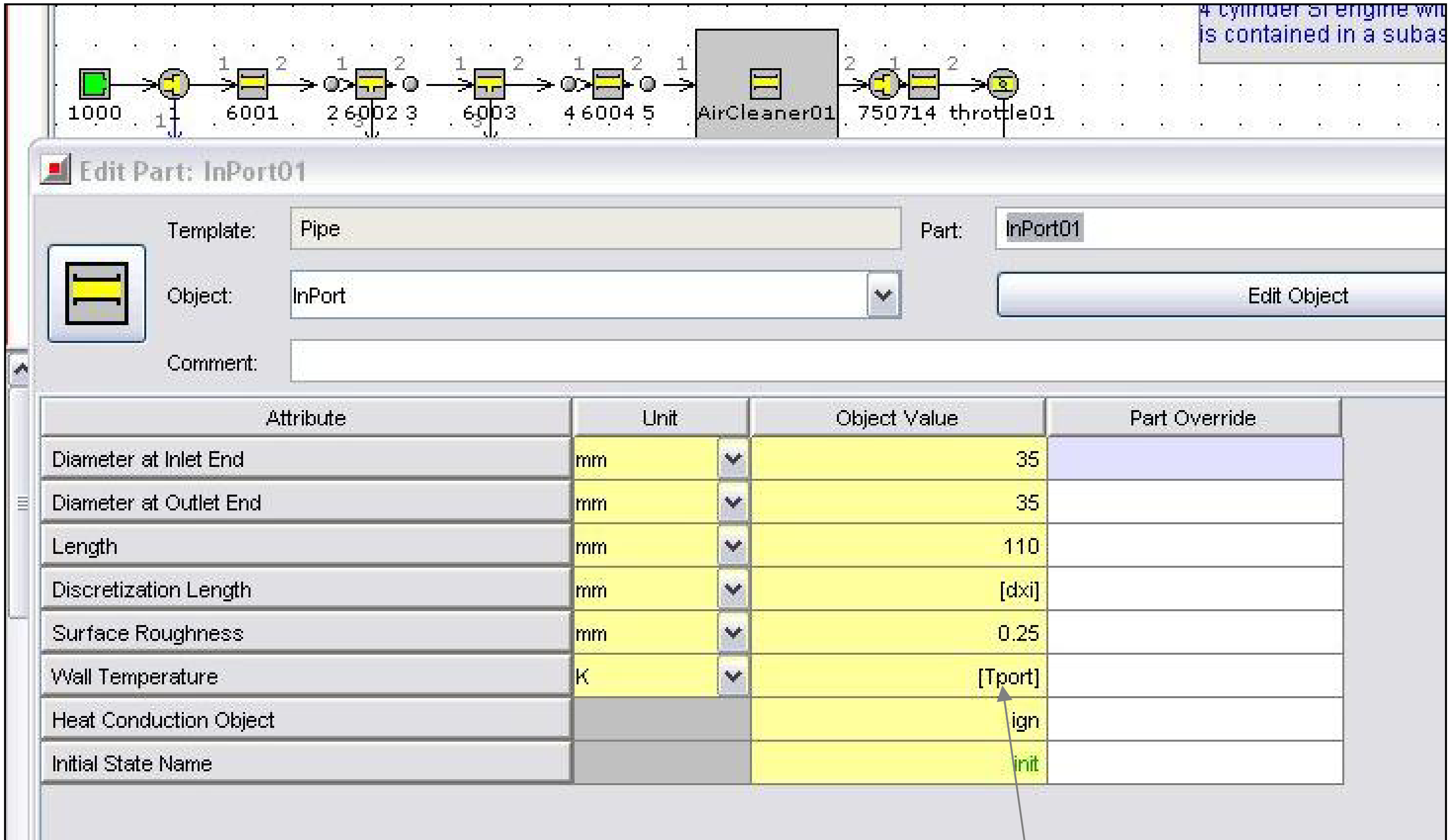
Engine data description

- 4-stroke 4 cylinder 2.2L
- Bore: 93mm; stroke: 81mm; compression ratio: 8.9
- 2 valves per cylinder
- Port injection
- Combustion: Wiebe function (with spark trimming simulated by the 50% burn point parameter)
- In-cylinder heat transfer modeled by Woschni model

Main GTPower model parameters

	Variable name	description	min	max
1	Tport	Intake port temperature [K]	300	500
2	Rlenght	Runners lenght [mm]	300	800
3	HrunM	Runners heat transfer multiplier	0.2	2
4	Hplenum	Plenum heat transfer multiplier	0.2	2
5	Elash	Exhaust valve lash [mm]	0.1	0.5
6	Etiming	Exhaust valve timing [°]	100	150
7	Ilash	Intake valve lash [mm]	0.1	0.5
8	Itiming	Intake valve timing [°]	200	260
9	Xmap	Scaling factor of Pressures ratio for lift-losses map	0.1	1.8
10	Ymap	Scaling factor of Lift for lift-losses map	1.2	2
11	Zmap	Scaling factor of Losses coefficients for lift-losses map	0.1	2

Main GTPower model parameters



Each parameter is defined indicating the variable name between [ ] in the definition of each part of the model

Main GTPower model parameters

Parameter	Unit	Label	...on)	2 (on)	3 (on)	4 (on)	5 (on)
BDURINIT		SiWebe burn duration	16	16	16	16	16
DXE	mm	Discretization length exhaust	51	51	51	51	51
DXI	mm	Discretization length intake	37	37	37	37	37
ELASH	mm	Exhaust valve lash	0.3	0.3	0.3	0.3	0.3
ETIMING	Cam Angle	Exhaust Valve Timing	126	126	126	126	126
EXHINITP	bar	Initial exhaust pressure	1.22	1.22	1.22	1.22	1.22
EXHINITT	K	Initial exhaust temperature	1120	1120	1120	1120	1120
EXHTWALL	K	Initial exhaust wall temperat...	1000	1000	1000	1000	1000
HPLENM		Heat Transfer Plenum Multip...	1	1	1	1	1
HRUNM		Runners Heat Transfer Multi...	1	1	1	1	1
HTMULT		Heat transfer multiplier used...	1	1	1	1	1
ILASH	mm	Intake valve lash	0.2	0.2	0.2	0.2	0.2
ITIMING	Cam Angle	Intake Valve Timing	239	239	239	239	239
MUFFINITP	bar	Muffler initial pressure	1.1	1.1	1.1	1.1	1.1
MUFFINITT	K	Muffler initial temperature	600	600	600	600	600
MUFTWALL	K	Muffler wall temperature	525	525	525	525	525
NCYC		Simulation duration	40	40	40	40	40
RLENGTH	mm	Runners lenght	500	500	500	500	500
RPM	RPM	Engine Speed	5500	5000	4500	4000	3500
THB50INIT		SiWebe anchor angle	10	10	10	10	10
THROTANG		Throttle angle	90	90	90	90	90

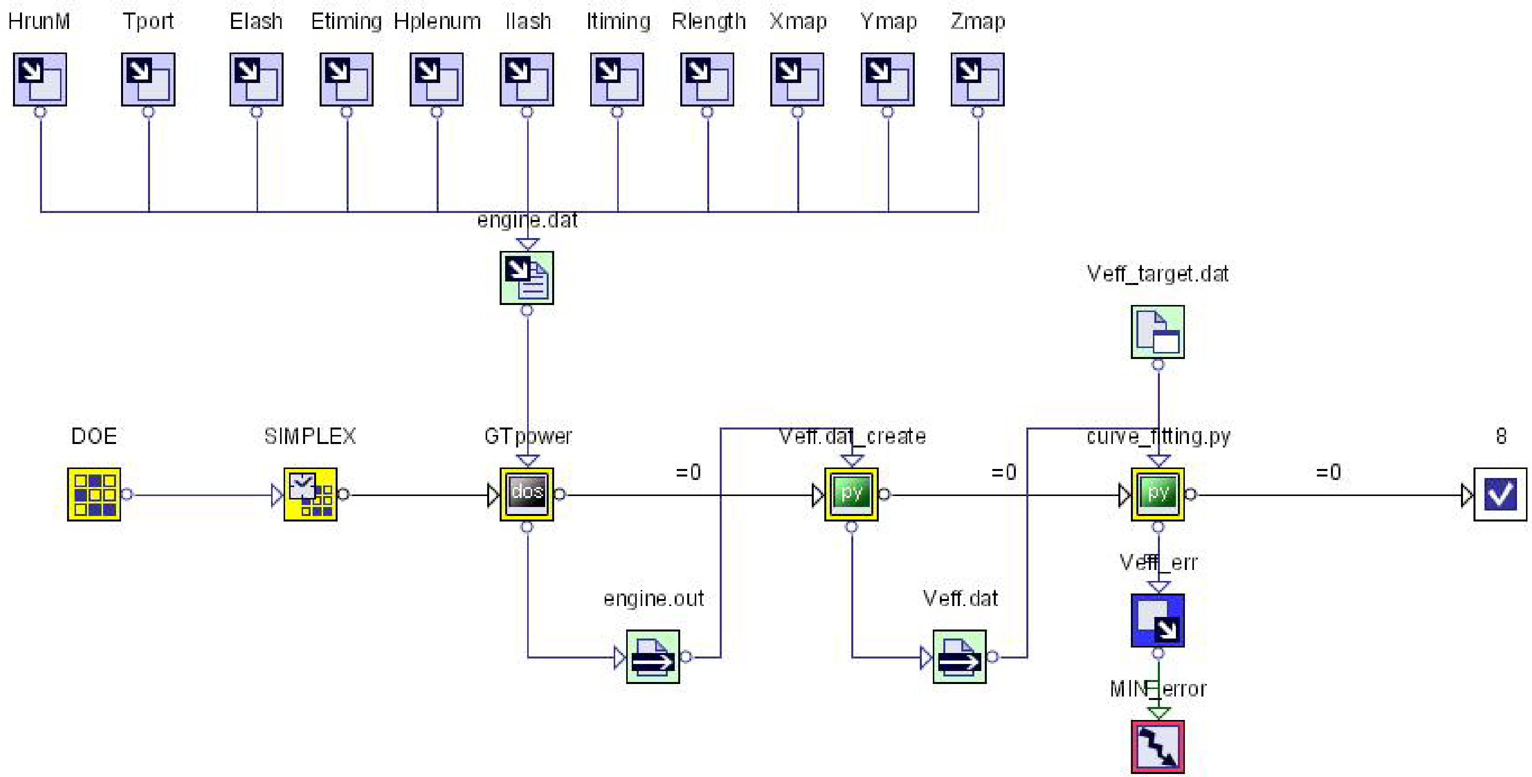
- The starting values of each parameter are defined by the CASE SETUP menu
- The engine speed is defined for 10 different rpm conditions
- All data are written by GTPower in **engine.dat** that becomes the modeFRONTIER input file

Calibration purpose

- The purpose is to **set the 11 model parameters** in order to **minimise the simulation error** with respect to the **experimental data** (relatively to volume efficiency vs rpm chart)

- For this goal, we use modeFRONTIER to define an **optimisation** case with **minimisation** of **volume efficiency error** as objective

modeFRONTIER workflow



Design parameters

Input variables

11 parameters

Output variables

Volume efficiency error

Design goals

Minimize Veff error

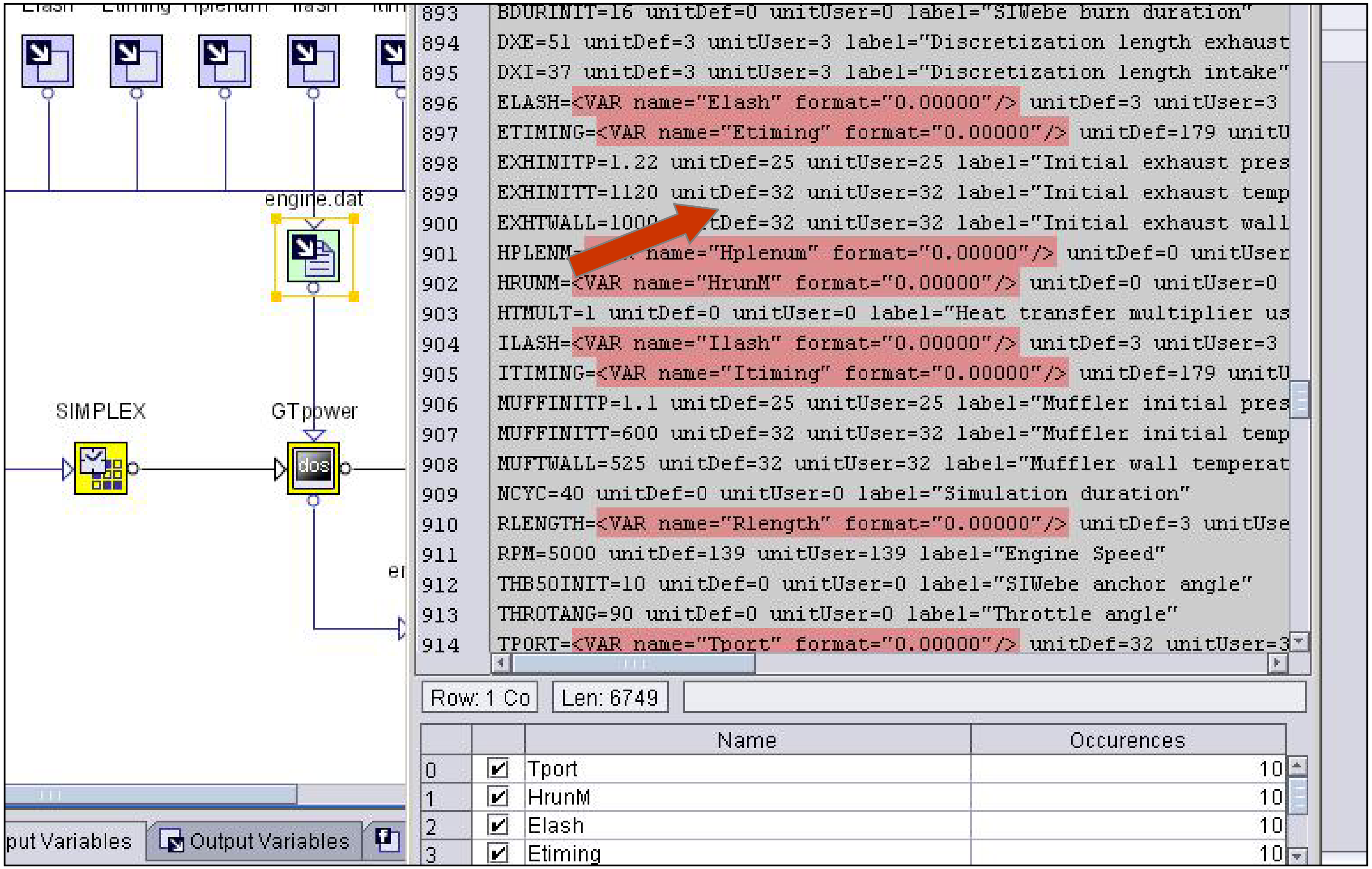
Sampling phase:

Random Doe – nr initial individuals: 12

Exploration phase:

SIMPLEX initial phase – NLPQLP refinement

modeFRONTIER workflow



Each input parameters is indicated by modeFRONTIER inside **engine.dat** file with correspondent format

modeFRONTIER workflow (GTPower script)

```
gtpower engine.dat

exit 0
```

This very simple UNIX command is used to run the GTPower model reading the parameters modified by modeFRONTIER

modeFRONTIER workflow (Veff.dat_create python program)

```
from java.lang import String as string
f = []
infile_name = CWD + '¥¥engine.out'
outfile_name = CWD + '¥¥Veff.dat'
keyword = "VOLEF(tot)    VOLEF(man)"
find = string.indexOf
infile= open(infile_name,'r')
while 1:
        line = infile.readline()
        if not line:
            break
        if find(line, keyword) <> -1:
            line = infile.readline()
            s = line.split()
            f.append(float(s[1]))
infile.close()
outfile=open(outfile_name,'w')
for i in range(len(f)):
    print f[i]
    outfile.write("%s¥n" % f[i])
outfile.close()
```

In engine.out GTPower output file, the data relative to Volumetric efficiency are found and reported in Veff.dat file

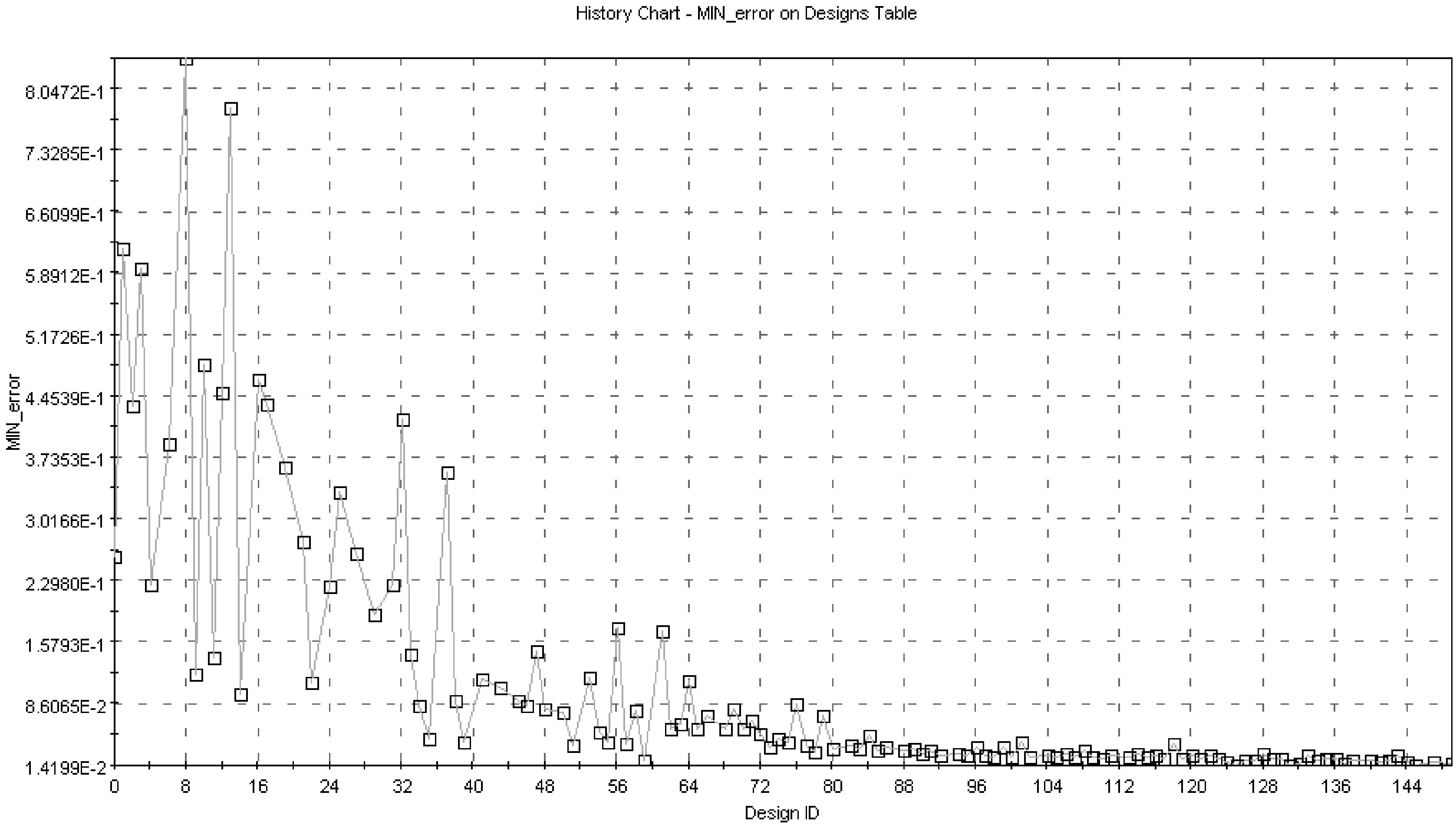
modeFRONTIER workflow (curve_fitting.pyt)

```
f = []
ft = []
target_file = CWD + '¥¥Veff_target.dat'
real_file = CWD + '¥¥Veff.dat'
# load TARGET data: one column file
infile = open(target_file,'r')
while infile:
    line = infile.readline()
    s = line.split()
    n = len(s)
    if n == 0:
        break
    ft.append(float(s[0]))
infile.close()
# load REAL data: one column file
infile = open(real_file,'r')
while infile:
    line = infile.readline()
    s = line.split()
    n = len(s)
    if n == 0:
        break
    f.append(float(s[0]))
infile.close()
```

```
# calculate absolute difference
d = []
dr = []
for i in range(len(ft)):
    d.append(abs(f[i]-ft[i]))
    dr.append(abs((f[i]-ft[i])/ft[i]))
# numerical integration: trapezoidal rule
integral = 0.0
for i in range(len(d)-1):
    integral = integral+(d[i]+d[i+1])/2.0
integral = integral/(len(d)-1)
print integral
outfile=open('error.dat','w')
outfile.write("%s¥n" % integral)
outfile.close()
```

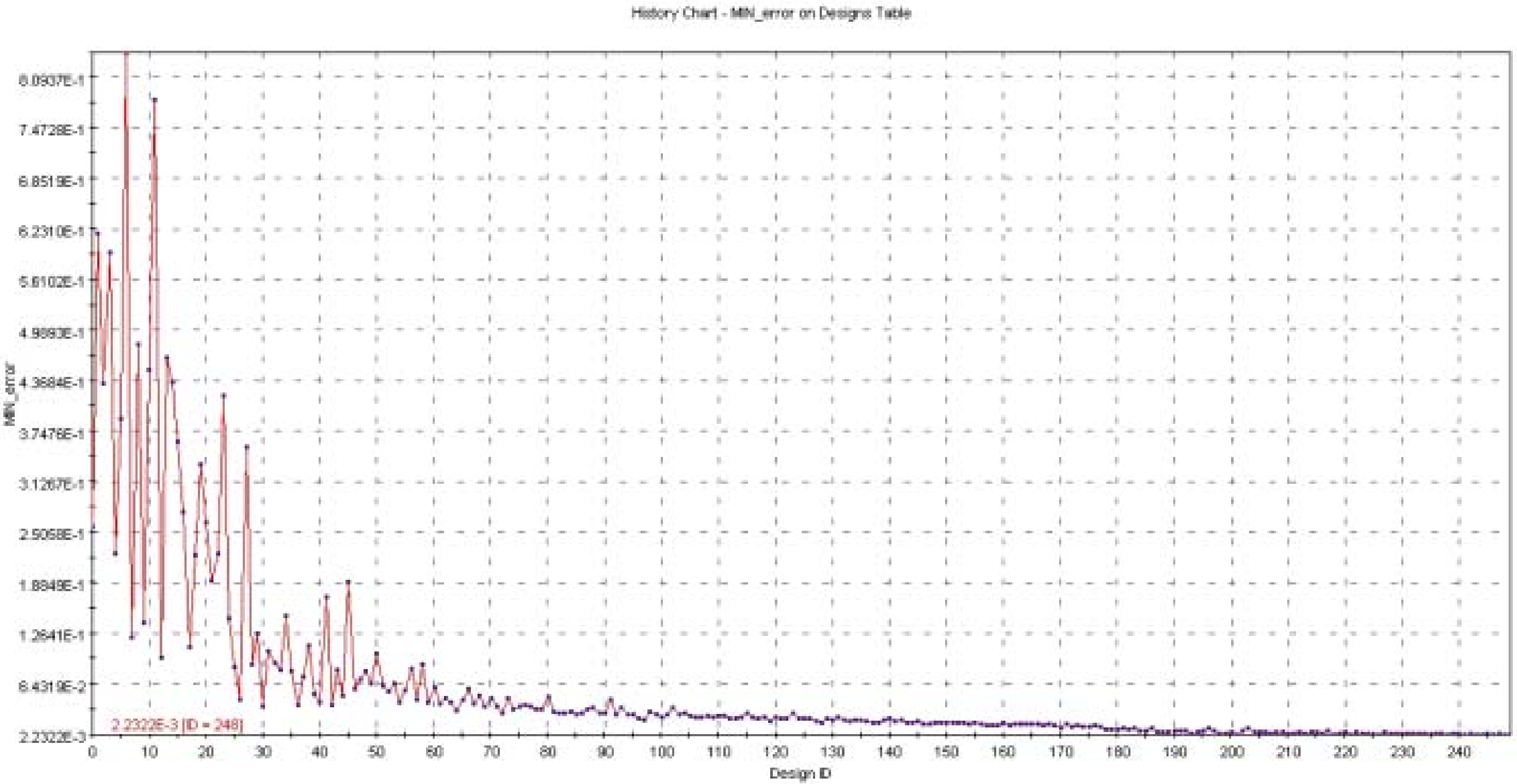
The Volume efficiency data read in Veff.dat file are compared to the target data, and the mean relative error is written in error.dat file

Optimisation results



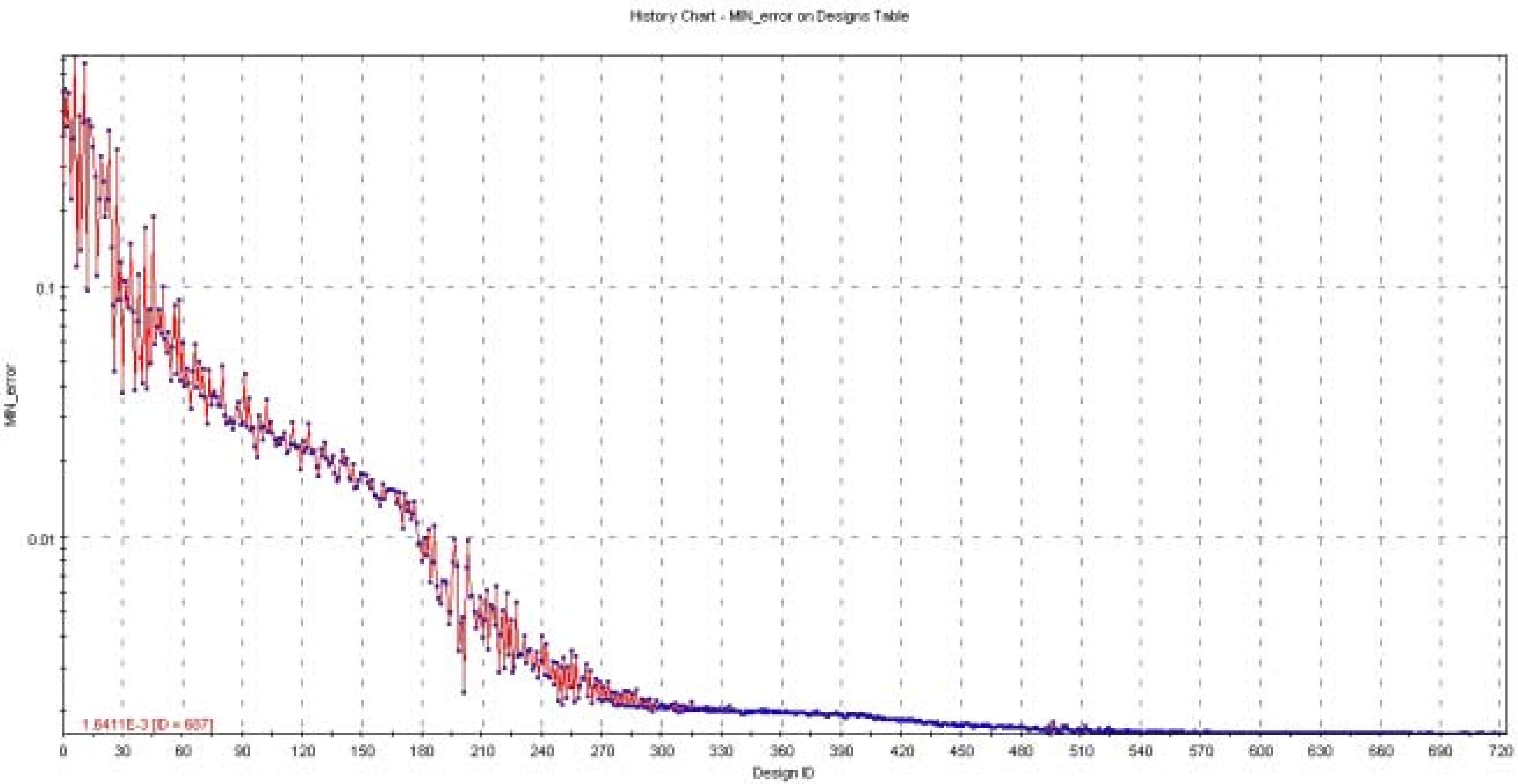
After 150 SIMPLEX iterations, the Veff error has been reduced from 30% to 1.4%

Optimisation results



After a total of 250 SIMPLEX iterations, the Veff error has been reduced to 0.23%

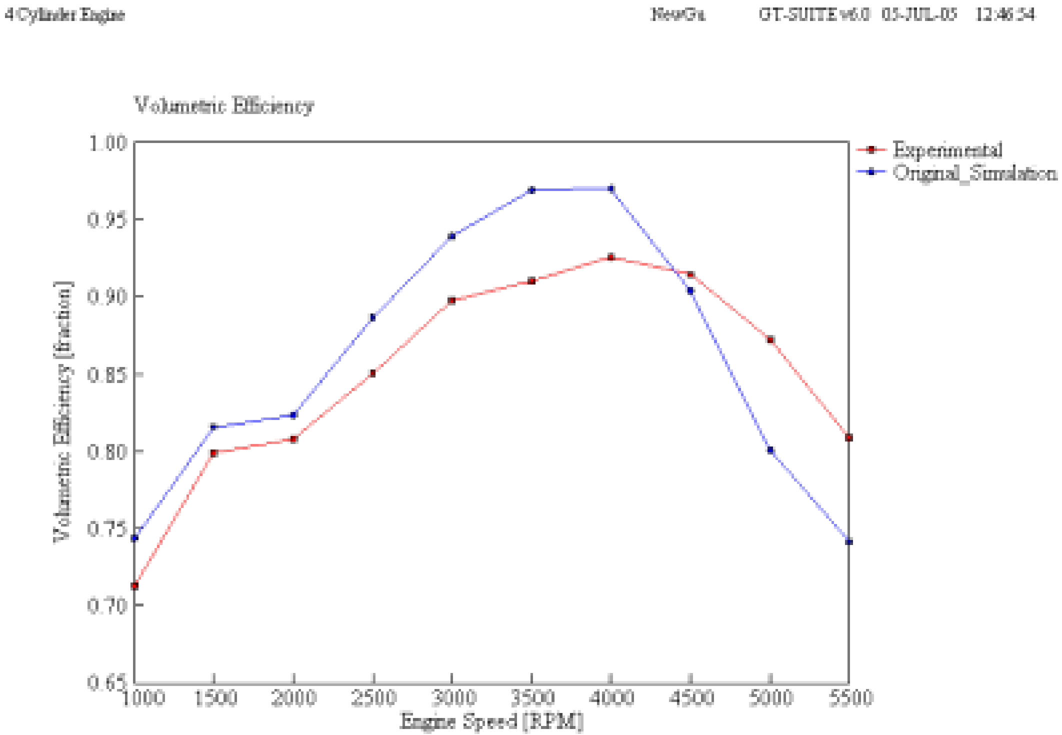
Optimisation results

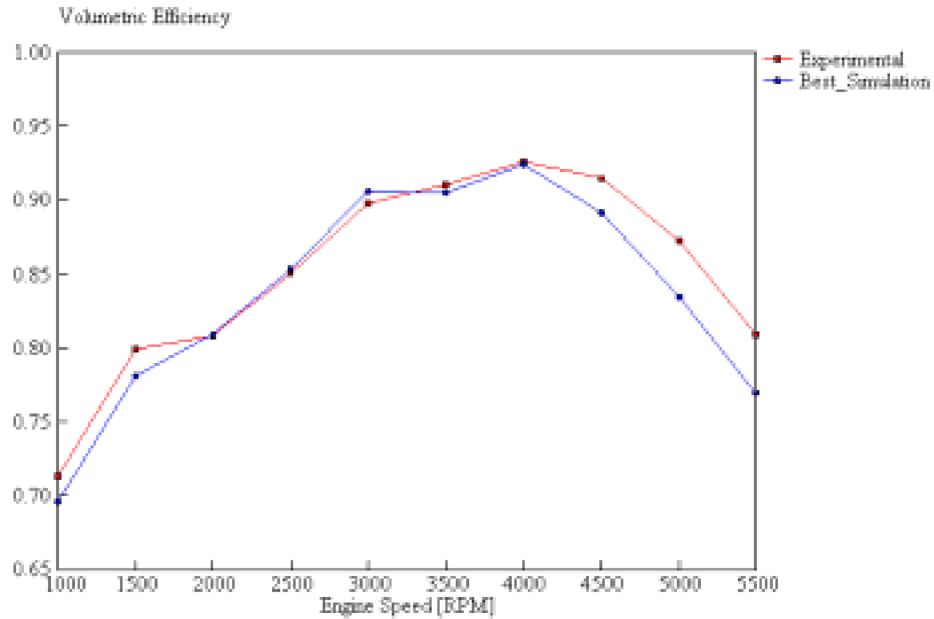


And after a total of 750 SIMPLEX iterations, the Veff error has been reduced to 0.16%

Optimisation results

comparison between target data and simulation (Veff vs RPM)



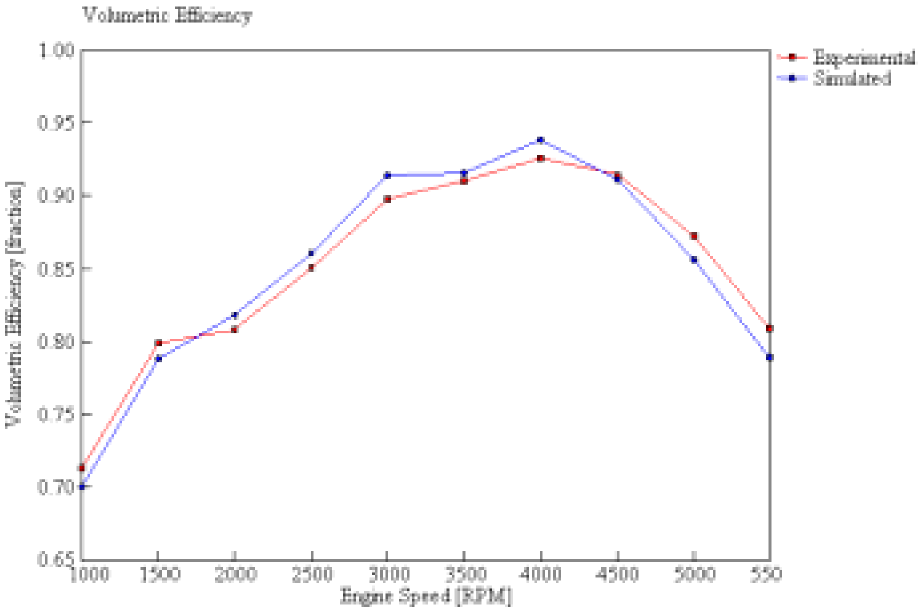


Initial guess: Mean error=30%

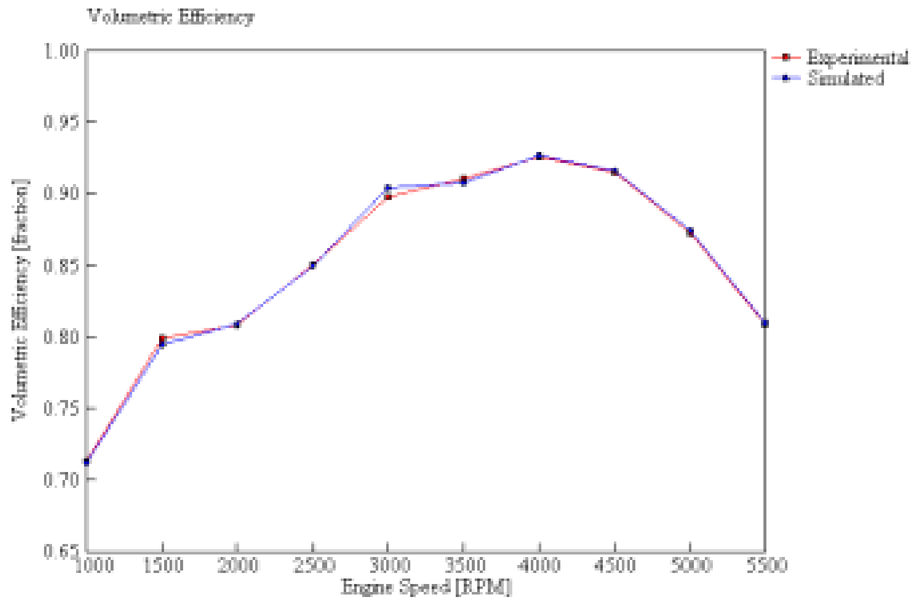
Optimised (150 run) : Mean error=1.42%

Optimisation results

comparison between target data and simulation (Veff vs RPM)



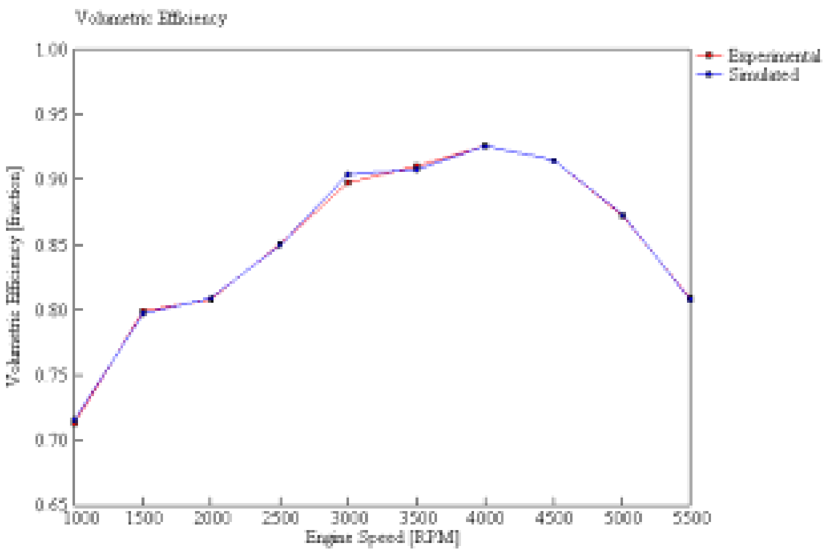
Optimised (200 run) : Mean error= 1.12%

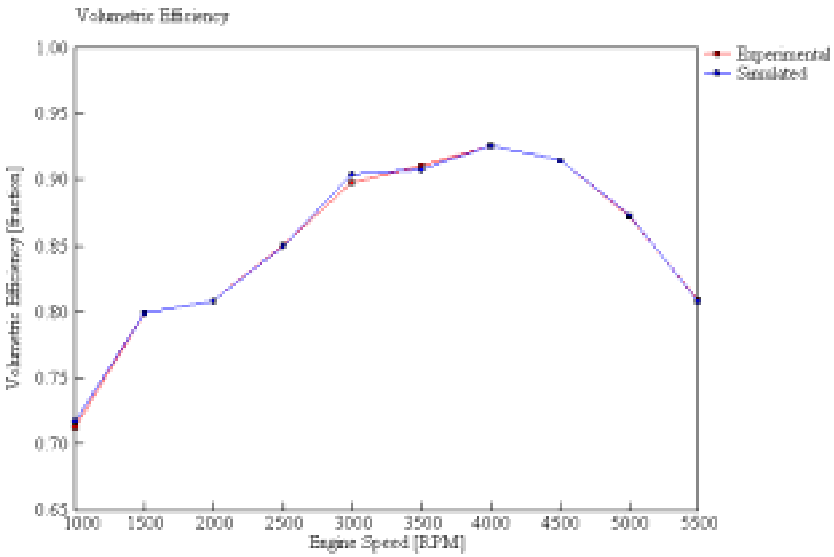


Optimised (250 run) : Mean error= 0.24%

Optimisation results

comparison between target data and simulation (Veff vs RPM)





Optimised (500 run) : Mean error= 0.18%

Optimised (750 run) : Mean error= 0.16%

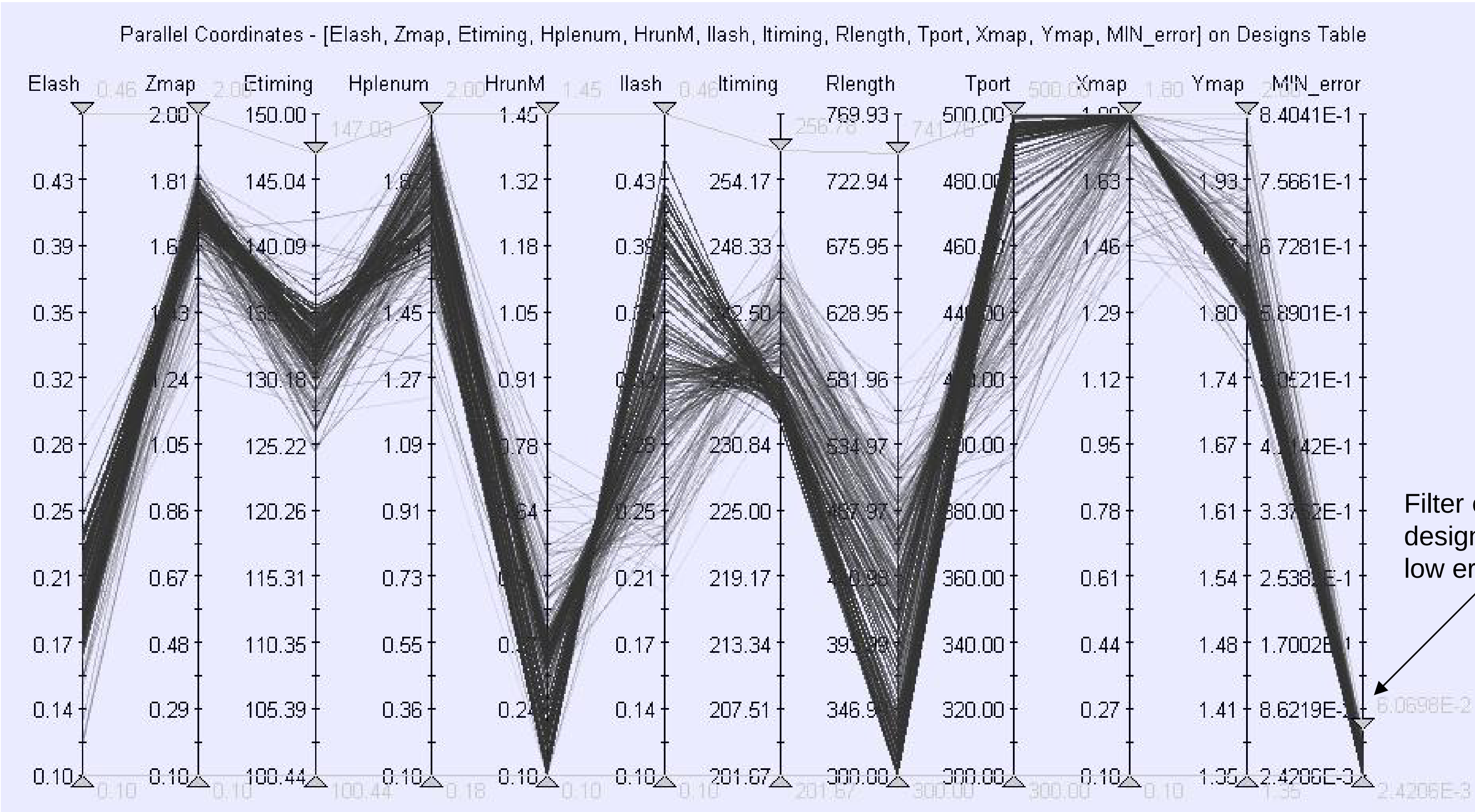
Statistical correlation

A positive number reveals a direct correlation, a negative reveals an inverse correlation, 0 indicates that the variables are not correlated

	Elash	Zmap	Etiming	Hplenum	HrunM	Ilash	Itiming	Rlength	Tport	Xmap	Ymap	MIN_error
Elash	1.00	-0.49	-0.30	-0.69	0.21	0.22	-0.41	0.11	-0.35	-0.54	-0.40	0.55
Zmap	-0.49	1.00	-0.04	0.66	-0.73	0.40	0.17	-0.47	0.72	0.65	0.29	-0.79
Etiming	-0.30	-0.04	1.00	-0.24	-0.15	-0.01	0.12	-0.05	0.06	0.02	0.34	-0.04
Hplenum	-0.69	0.66	-0.24	1.00	-0.26	0.17	0.01	-0.39	0.65	0.62	0.08	-0.62
HrunM	0.21	-0.73	-0.15	-0.26	1.00	-0.67	-0.19	0.65	-0.74	-0.64	-0.24	0.67
Ilash	0.22	0.40	-0.01	0.17	-0.67	1.00	-0.46	-0.89	0.59	0.26	-0.24	-0.26
Itiming	-0.41	0.17	0.12	0.01	-0.19	-0.46	1.00	0.33	0.04	0.49	0.43	-0.42
Rlength	0.11	-0.47	-0.05	-0.39	0.65	-0.89	0.33	1.00	-0.63	-0.45	0.23	0.43
Tport	-0.35	0.72	0.06	0.65	-0.74	0.59	0.04	-0.63	1.00	0.73	0.00	-0.72
Xmap	-0.54	0.65	0.02	0.62	-0.64	0.26	0.49	-0.45	0.73	1.00	0.21	-0.87
Ymap	-0.40	0.29	0.34	0.08	-0.24	-0.24	0.43	0.23	0.00	0.21	1.00	-0.20
MIN_error	0.55	-0.79	-0.04	-0.62	0.67	-0.26	-0.42	0.43	-0.72	-0.87	-0.20	1.00

- 3 variables are **low** correleted with objective: Etiming, Ilash, Ymap
- 5 variables are highly **inversely** correlated: Zmap, Hplenum, Itiming, Tport, Xmap
- 3 variables are highly **directly** correlated: Elash, HrunM, Rlenght

Parallel chart

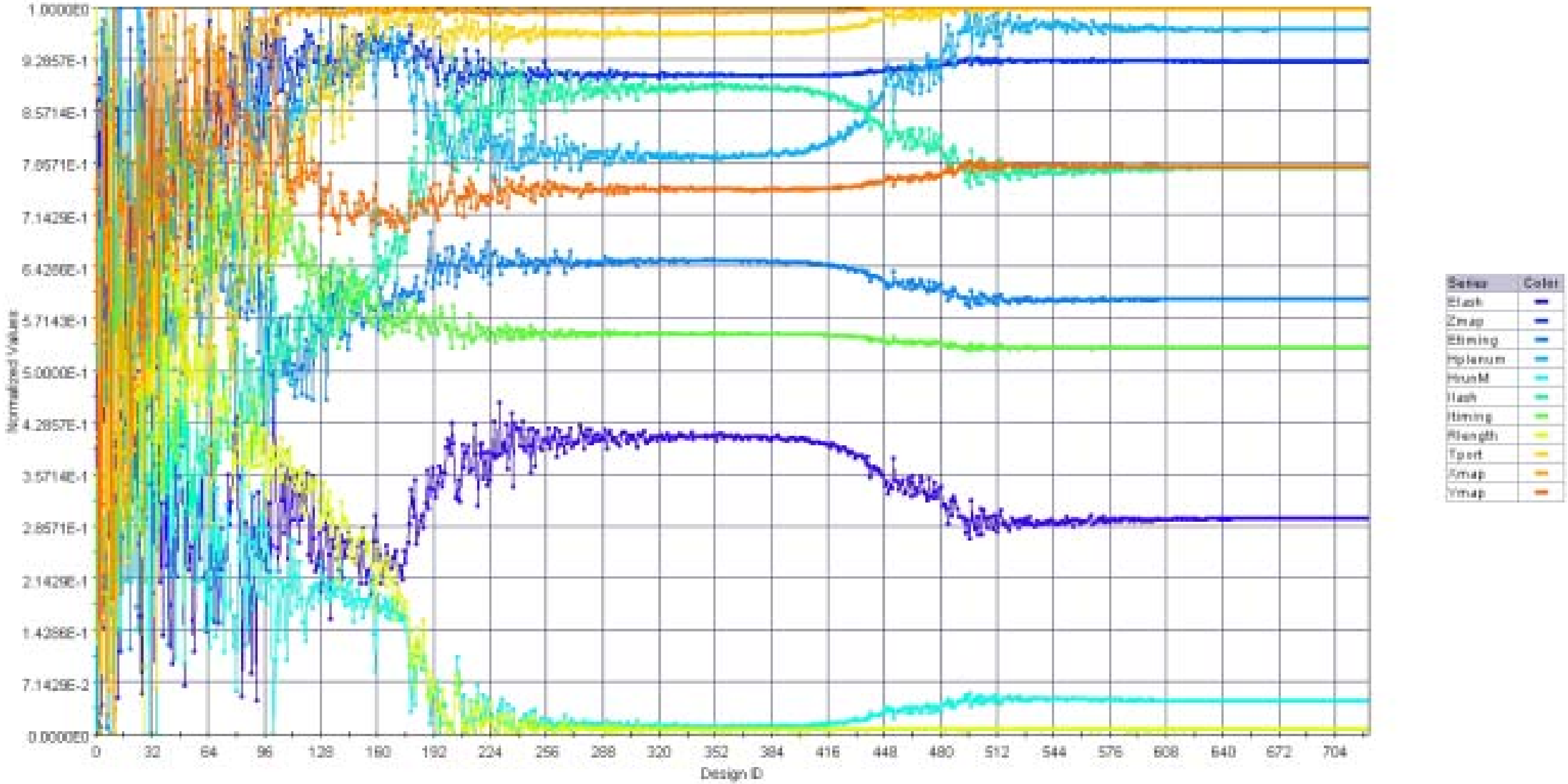


Filter on the designs for low error

To produce low value of error, most of designs have:

- **High** values for the variables: Zmap, Hplenum, Tport, Xmap
- **Low** values for the variables : Elash, HrunM, Rlenght

Multi-History Chart



Conclusion

- modeFRONTIER has been used to set properly the parameters of a GTPower model, relative to a 4-cyl engine, in order to match the experimental data (inverse design)

- A Simplex optimisation strategy has allowed to reproduce the experimental Volume efficiency vs RPM chart with less than 0.2% of relative error

- Statistical analysis tools of modeFRONTIER has been used to find which are the most significant parameters for the calibration