RA2311026011010
Siddhesh Khankhoje

Hybrid and Electric Vehicles

CLA-1 Assesment

Problem Statement:

Modern gasoline engines use **three-way catalytic converters (TWC)** to control exhaust emissions.
The **Oxygen Storage Capacity (OSC)** of the catalytic converter plays a critical role in maintaining stoichiometric air–fuel ratio by absorbing and releasing oxygen during transient engine operation.

The OSC of the oxidation catalyst depends on:

- **Catalyst volume (L)**
- **Washcoat loading (g/L)**
- **CeO_2 weight fraction in the washcoat**
- **Utilisation efficiency (η)**

The objective of this work is to:

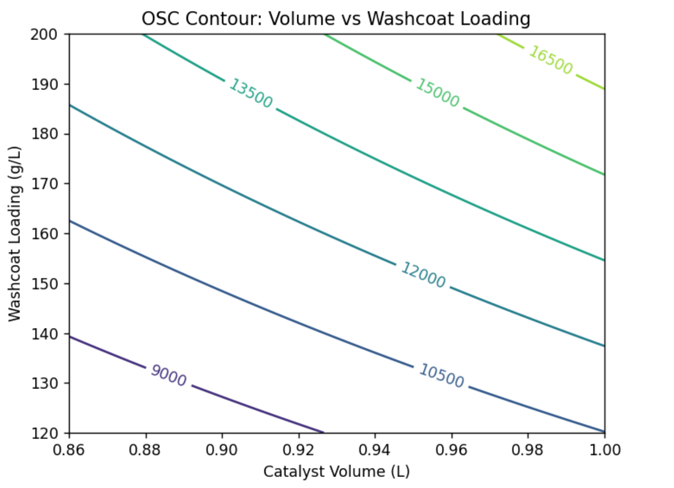
1. Develop a **physics-based OSC model**
2. Generate **lookup tables (LUTs)** for OSC
3. Visualize the sensitivity of OSC using:
    - Volume vs Washcoat loading
    - Volume vs CeO_2 wt. fraction
    - Volume vs Utilisation efficiency
4. Create **OSC contour plots** and a **3D surface plot** for calibration and injector-control applications.

Solution:

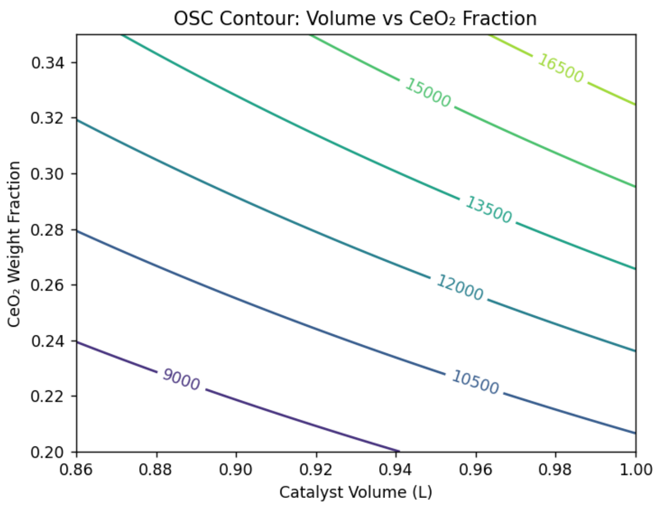
1. Lookup table:

Volume_L	Washcoat_	CeO2_frac	Efficiency_	OSC
0.86	120	0.2	0.65	5235.259
0.875556	128.8889	0.216667	0.661111	6421.938
0.891111	137.7778	0.233333	0.672222	7786.624
0.906667	146.6667	0.25	0.683333	9345.789
0.922222	155.5556	0.266667	0.694444	11116.79
0.937778	164.4444	0.283333	0.705556	13117.91
0.953333	173.3333	0.3	0.716667	15368.33
0.968889	182.2222	0.316667	0.727778	17888.22
0.984444	191.1111	0.333333	0.738889	20698.69
1	200	0.35	0.75	23821.88

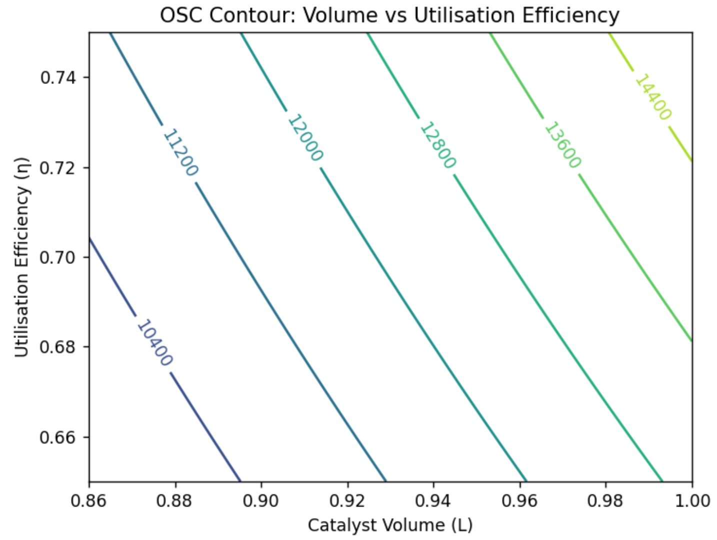
2. Volume v/s Wash coat loading



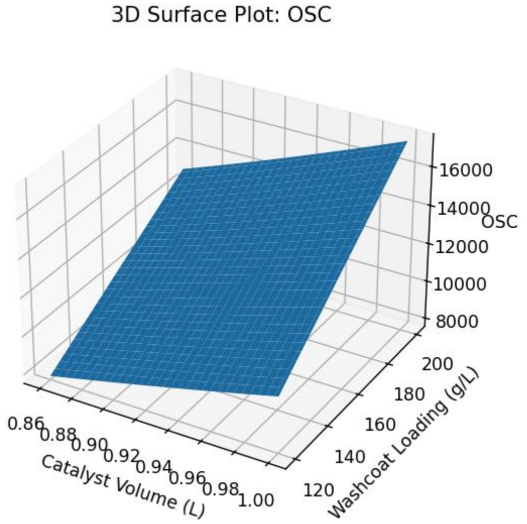
3. Volume v/s Ceria (CeO2) %weight fraction



4. Volume v/s Utilisation efficiency



5. 3D Surface Plot: OSC



6. Program (Python code):

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# ===============================
# CONSTANTS
# ===============================
CPSI = 750
k_factor = 5.5
```

```python
O2_PER_G_CEO2 = 0.11

# Nominal values
washcoat_nom = 160        # g/L
ceo2_nom = 0.275          # wt fraction
eta_nom = 0.70

# ===============================
# AXES RANGES
# ===============================
volume = np.linspace(0.86, 1.0, 60)
washcoat = np.linspace(120, 200, 60)
ceo2_frac = np.linspace(0.20, 0.35, 60)
eta = np.linspace(0.65, 0.75, 60)

# ===============================
# OSC FUNCTION
# ===============================
def compute_osc(volume, washcoat, ceo2_frac, eta):
    washcoat_mass = volume * washcoat
    ceo2_mass = washcoat_mass * ceo2_frac
    stored_o2 = ceo2_mass * O2_PER_G_CEO2 * eta
    osc_geom = CPSI * k_factor * volume
    return osc_geom * stored_o2

# ===============================
# 1. Volume vs Washcoat Loading
# ===============================
V1, W = np.meshgrid(volume, washcoat)
OSC_vw = compute_osc(V1, W, ceo2_nom, eta_nom)

plt.figure()
c1 = plt.contour(V1, W, OSC_vw)
plt.clabel(c1)
plt.xlabel("Catalyst Volume (L)")
plt.ylabel("Washcoat Loading (g/L)")
plt.title("OSC Contour: Volume vs Washcoat Loading")
plt.show()

# ===============================
# 2. Volume vs CeO2 wt fraction
# ===============================
V2, C = np.meshgrid(volume, ceo2_frac)
OSC_vc = compute_osc(V2, washcoat_nom, C, eta_nom)

plt.figure()
c2 = plt.contour(V2, C, OSC_vc)
plt.clabel(c2)
```

```
plt.xlabel("Catalyst Volume (L)")
plt.ylabel("CeO₂ Weight Fraction")
plt.title("OSC Contour: Volume vs CeO₂ Fraction")
plt.show()

# ================================
# 3. Volume vs Utilisation Efficiency
# ================================
V3, E = np.meshgrid(volume, eta)
OSC_ve = compute_osc(V3, washcoat_nom, ceo2_nom, E)

plt.figure()
c3 = plt.contour(V3, E, OSC_ve)
plt.clabel(c3)
plt.xlabel("Catalyst Volume (L)")
plt.ylabel("Utilisation Efficiency (η)")
plt.title("OSC Contour: Volume vs Utilisation Efficiency")
plt.show()

# ================================
# 4. FILLED OSC CONTOUR
# ================================
plt.figure()
plt.contourf(V1, W, OSC_vw)
plt.colorbar(label="OSC")
plt.xlabel("Catalyst Volume (L)")
plt.ylabel("Washcoat Loading (g/L)")
plt.title("Filled OSC Contour: Volume vs Washcoat")
plt.show()

# ================================
# 5. 3D SURFACE PLOT
# ================================
fig = plt.figure()
ax = fig.add_subplot(111, projection="3d")
ax.plot_surface(V1, W, OSC_vw)
ax.set_xlabel("Catalyst Volume (L)")
ax.set_ylabel("Washcoat Loading (g/L)")
ax.set_zlabel("OSC")
ax.set_title("3D Surface Plot: OSC")
plt.show()

# ================================
# 6. LOOKUP TABLE (LUT)
# ================================
volume_lut = np.linspace(0.86, 1.0, 10)
washcoat_lut = np.linspace(120, 200, 10)
ceo2_lut = np.linspace(0.20, 0.35, 10)
```

```
    eta_lut = np.linspace(0.65, 0.75, 10)

    osc_lut = compute_osc(volume_lut, washcoat_lut, ceo2_lut, eta_lut)

    lut = pd.DataFrame({
        "Volume_L": volume_lut,
        "Washcoat_g_per_L": washcoat_lut,
        "CeO2_fraction": ceo2_lut,
        "Efficiency_eta": eta_lut,
        "OSC": osc_lut
    })

    print("\n=== OSC LOOKUP TABLE ===\n")
    print(lut)

    lut.to_csv("osc_lookup_table.csv", index=False)
```