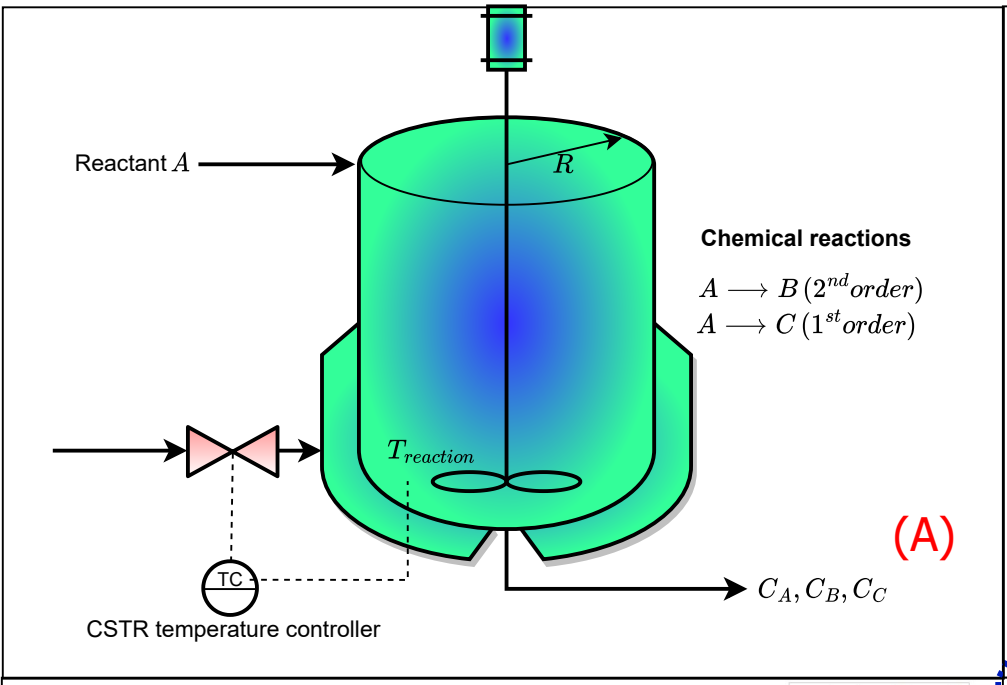
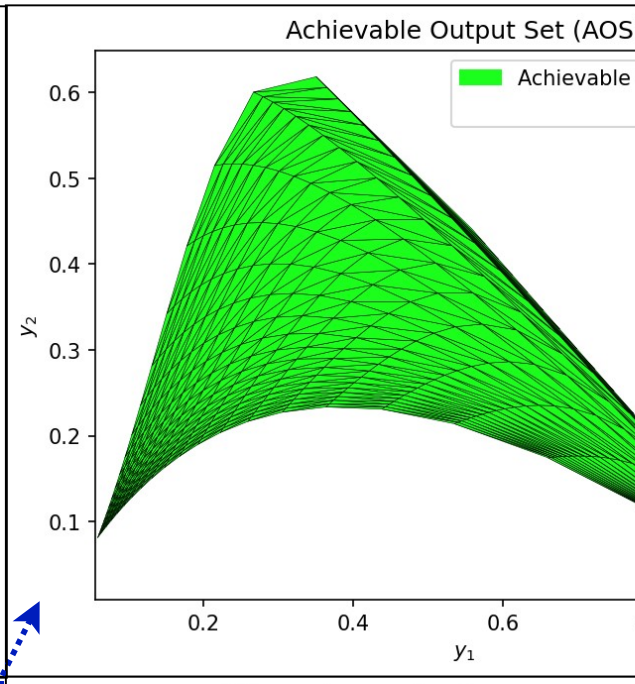
Seamless operability quantification using ***opyrability***: *multimodel_rep* and *OI*

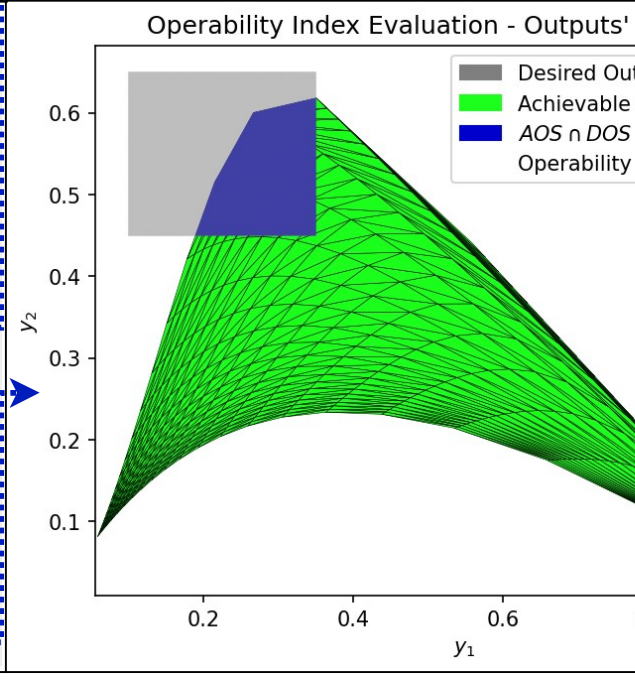




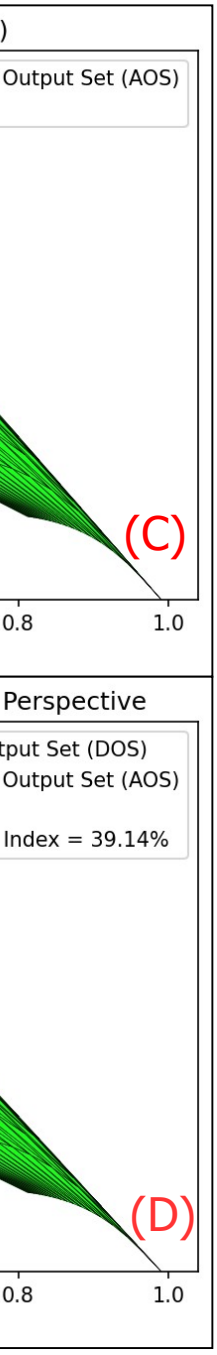

```python
from opyrability import multimodel_rep, OI_eval

# AIS Bounds
AIS_bounds = np.array([[0.25,  3.00],
                       [1.00, 15.00]])

# Discretization Resolution
AIS_resolution = [20, 20]

# Obtain AOS
AOS_region = multimodel_rep(CSTR,
                            AIS_bounds,
                            AIS_resolution)

# DOS Bounds and resolution
DOS_bounds = np.array([[0.10,   0.35],
                       [0.45,   0.65]])

# Obtain Operability Index (OI)
OI = OI_eval(AOS_region, DOS_bounds)
```

(B)

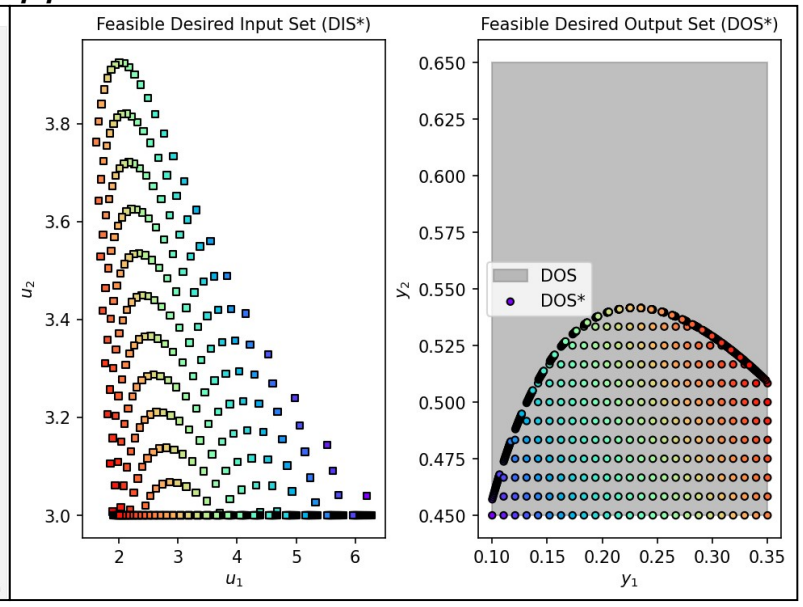
[3] ✓ 4.3s Python



I_eval


)
Output Set (AOS)
(C)
0.8
1.0
Perspective
tput Set (DOS)
Output Set (AOS)
Index = 39.14%
(D)
0.8
1.0

Inverse mapping with automatic differentiation and warm-start, using ***opyrability***: *nlp_based_approach*

```python
from opyrability import nlp_based_approach

# Initial estimate for inverse mapping, lower/upper bounds.
u0 = np.array([3 , 10])
lb = np.array([0.25, 3])
ub = np.array([50, 50])

# DOS Bounds and resolution
DOS_bounds = np.array([[0.1 ,   0.35],
                       [0.45,   0.65]])
DOS_resolution = [25, 25]

# Obtaining inverse map of the CSTR design
fDIS, fDOS, message = nlp_based_approach(CSTR,
                                         DOS_bounds,
                                         DOS_resolution,
                                         u0,
                                         lb,
                                         ub,
                                         method='ipopt',
                                         plot=True,
                                         ad=True,
                                         warmstart=True)
```

[4] ✓ 3m 14.7s Python