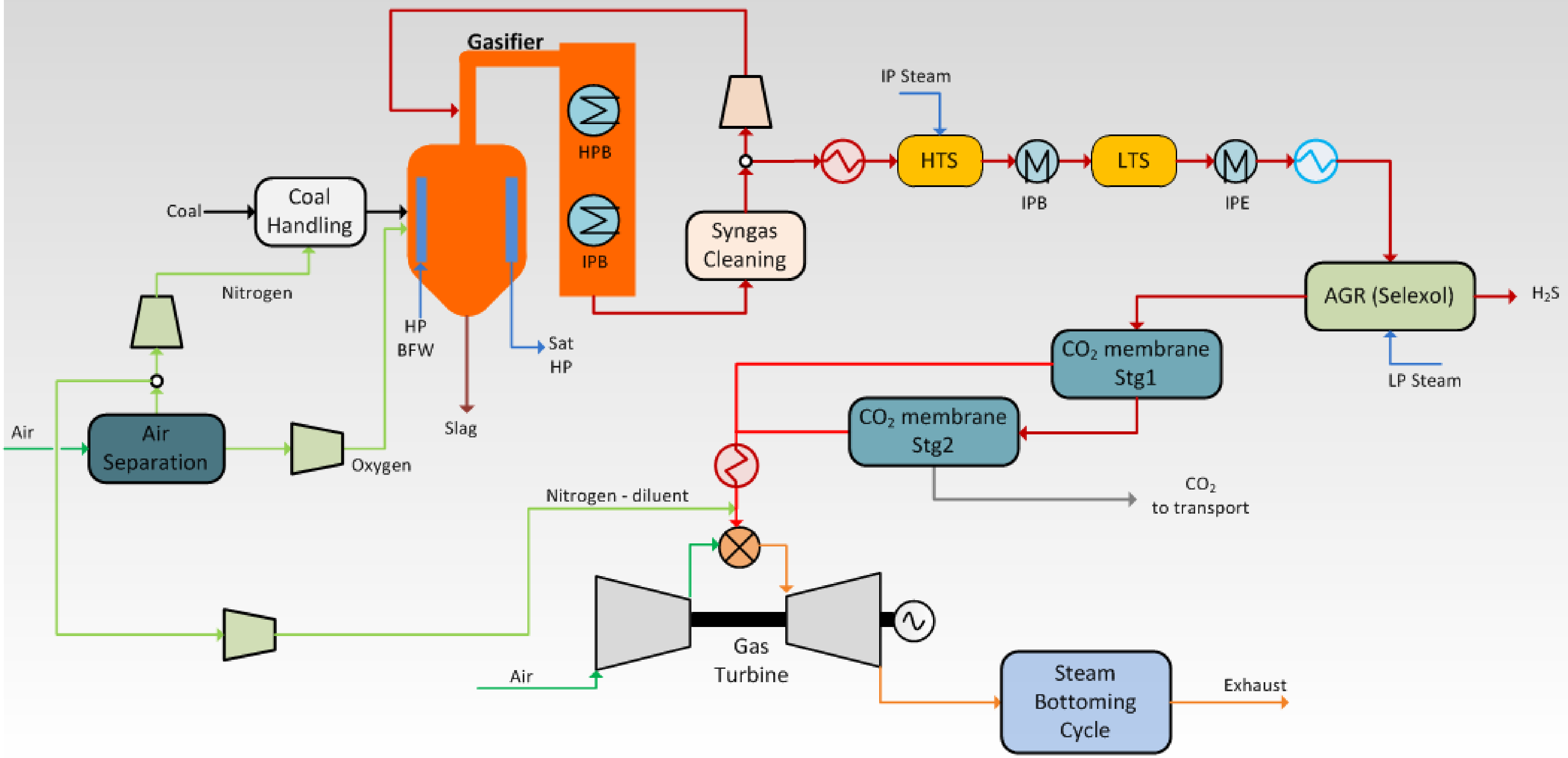
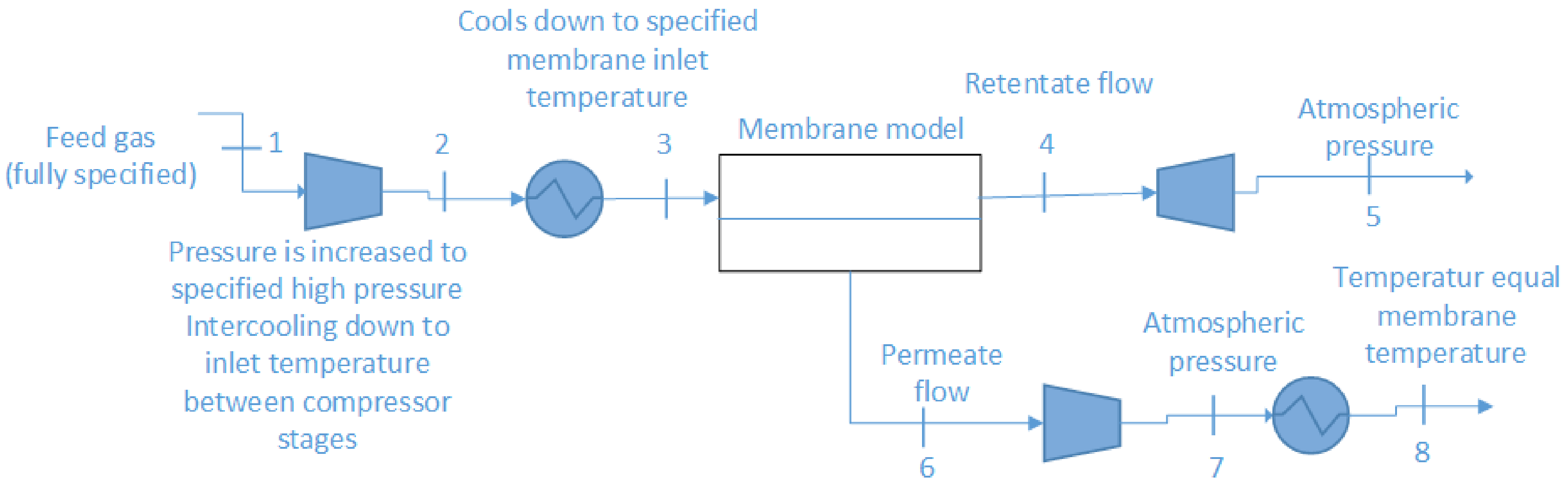
Membrane for pre-combustion capture from a IGCC

Simon Roussanaly, SINTEF Energy Research
Rahul Anantharaman, SINTEF Energy Research
Brede Hagen, SINTEF Energy Research

Conceptual process diagram



Membrane process model



Screen shot of the model

- MembraneCarbonCapture
  - src
    - structures
      - __init__.py
      - attainableRegion.py
      - compressor.py
      - economy.py
      - expander.py
      - heatExchanger.py
      - idealGas.py
      - membrane.py
      - membraneSystem.py
      - optimization.py
    - attainableRegionPlotting.py
    - compressorTester.py
    - costModel.py
    - expanderTester.py
    - heatExchangerTester.py
    - ideal_gas_tester.py
    - membraneSystemTester.py
    - membraneSytemAndCostModel.py
    - membraneTester.py
    - processOptimization.py
  - Anaconda Python ... aconda\python.e
  - brede
- RemoteSystemsTempFiles

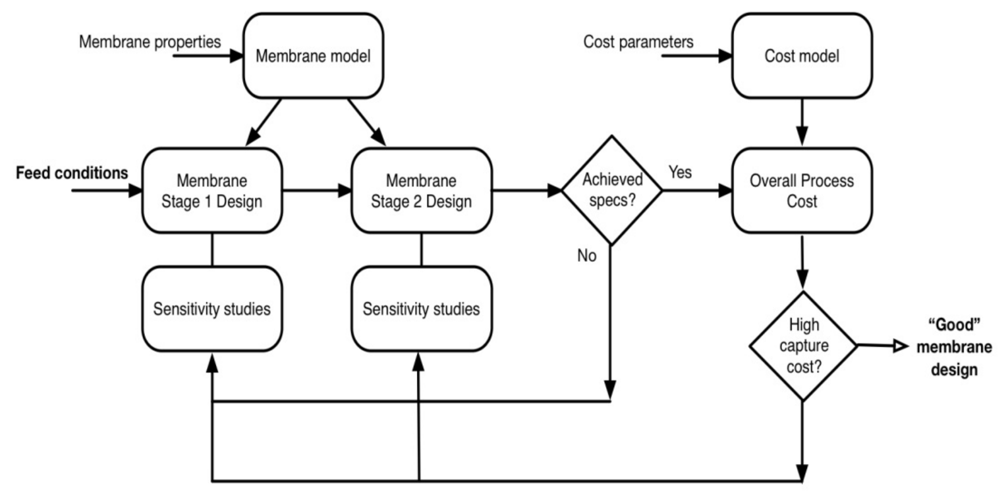
```
        for gas in self.streamArr:
            i += 1
            p = gas.p/self.kToUnity
            T = gas.T - self.CtoK
            mfl = gas.mfl
            zCO2 = gas.getComponentByName("CO2").z
            zN2 = gas.getComponentByName("N2").z
            print str(i).rjust(width),str("%.1f" % p).rjust(wid

    def calculate(self):
        self.netPower = 0
        # Compressor
        self.compressor.calculate()
        self.compPower = self.compressor.power
        self.netPower += sum(self.compressor.power)

        # Cooler and intercooler
        self.coolerArea = 0.0
        self.coolerMfl = 0.0

        # Intercooler
        self.coolerArea += self.compressor.interCoolerArea
        self.coolerMfl += self.compressor.intercoolerWaterMfl


        # Coling after last compression stage
        self.cooler.calculate()
        self.coolerArea += self.cooler.area
        self.coolerMfl += self.cooler.extMfl
        #self.cooler.printHx()
        # Membrane
        self.membrane.solveMembrane(self.inputParamName, self.i
        self.membraneArea = self.membrane.area
        self.stageCut = self.membrane.theta
        self.CCR = self.membrane.CCR
        self.captureRate = self.membrane.captureRate
        self.purity = self.membrane.purity
```

What membrane properties for post-combustion CO_2 capture from a coal power plant?

- Identify which membrane properties and process configuration which can compete with solvent based capture
  - Which will give feedback for membrane development

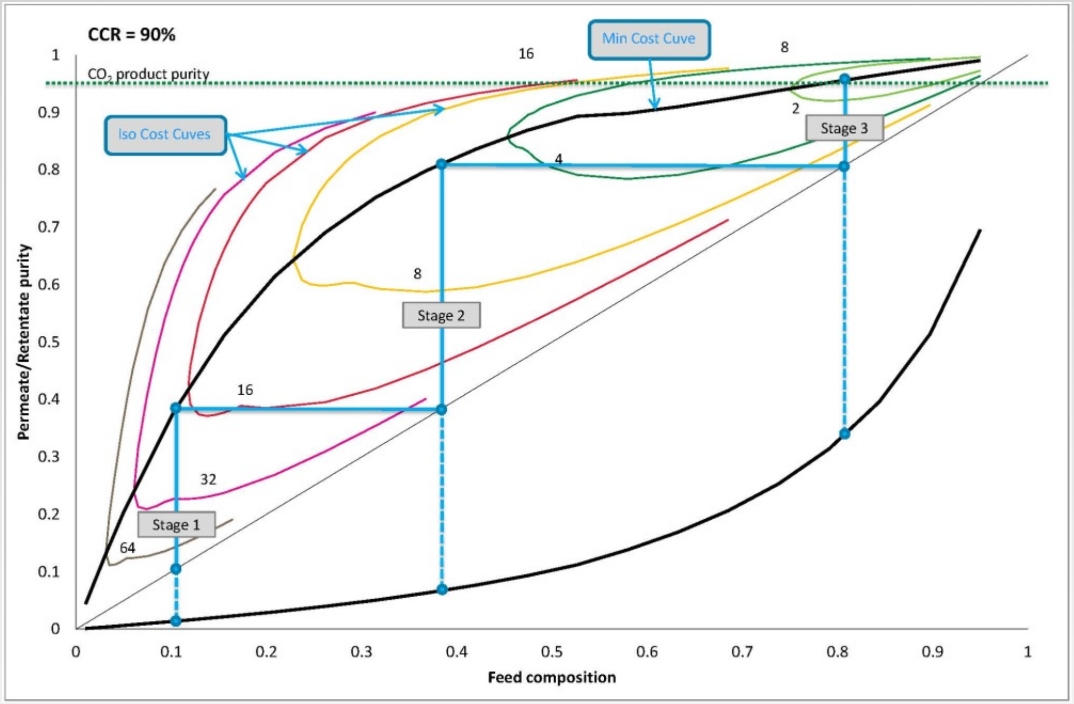
- Common methodoly for evaluation is mainly based on sensitivity analyses
  - Not suitable to identify desired properties

- A new methodology called the Attainable region is here used to
  - Optimise multi-stage capture processes
  - Identify which properties are required to compete with solvent based capture
  - This have already been done for post-combustion and it will be done for the pre-combustion in PilotCCS



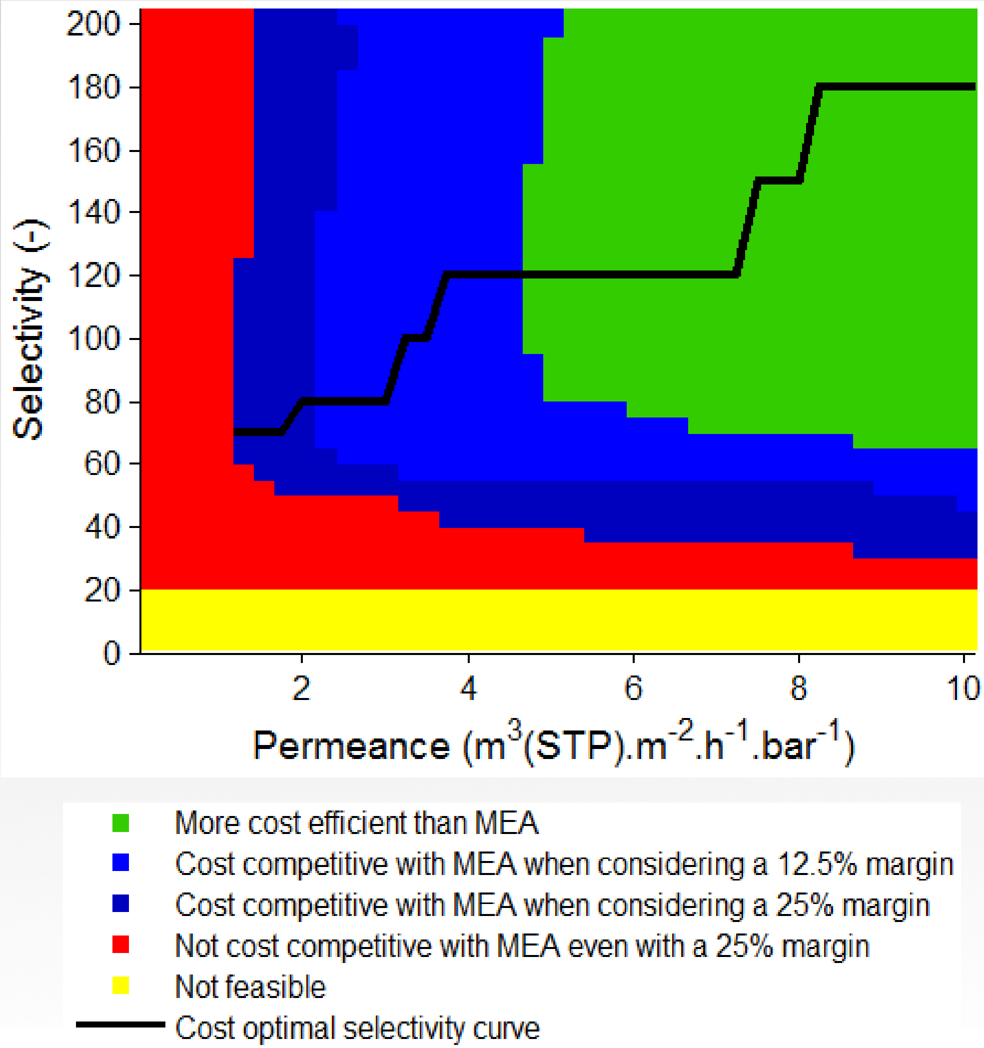
Sensitivity analyses based approach

The Attainable Region approach

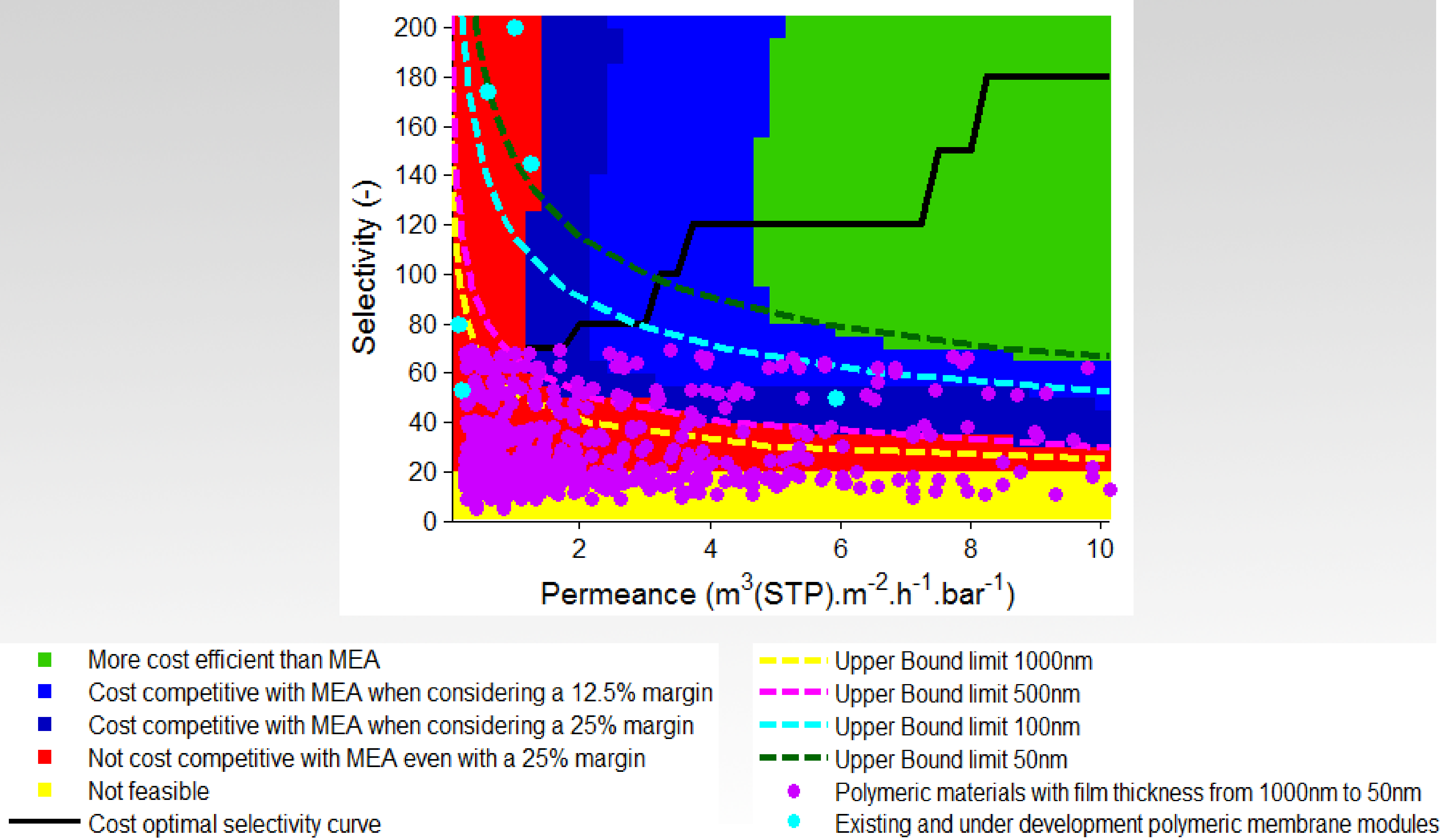


- The methodology produces a curve that is simple to use and also gives feedback on potential of particular membranes.
- The design procedure has been automated to provide optimal designs for a given flue gas and membrane

What membrane properties for post-combustion CO_2 capture from a coal power plant?



What materials for post-combustion CO_2 capture from a coal power plant?



Conclusions

- The methodology developed at SINTEF Energy Research allows to identify the membrane properties required for CO_2 capture from a specific application
- The results identify:
  - When advanced configurations will be required
  - The trade-off between membrane properties
  - Membrane materials and modules relevant
- Results can:
  - Guide the development of membrane materials for cost-effective CO_2 capture
  - Help the industry to select membranes that can compete with solvent-based capture systems.
- In the PilotCCS project, we plan to perform the same assessment and comparison for pre-combustion CO_2 capture from an IGCC power plant