A Proposal to engineer visibility into outbound goods as they are manufactured, assembled and distributed

Introduction

Manufacturing companies are implementing strategies to make calculated risks, in line with the COVID-19 pandemic trend, in order to ensure continued profits. These companies are particularly seeking technology that will provide greater operational visibility into their supply chains. That way, they can predict which goods and materials are in the supply chain, where they are in the chain and when their need will become urgent. Such Value of Information (VOI) is priceless in times like this. Manufacturers will be able to predict market prices and estimate how much to spend on raw materials and how much to produce, at any given instant in time.

This document proposes a design for a Robust Control system based on fundamental Chemical Engineering principles of Process Control and Dynamic, to engineer visibility into outbound goods as they are manufactured, assembled and shipped to customers through the transportation network.

Discussion of the Underlying Engineering Principles

The system is programmed using Python, and takes advantage of the modelling properties of “Pyomo”. The optimization model is designed as a linear function and specified by decision variables, constraints and an optimization objective. Furthermore, the model uses a GNU Linear Programming Kit (glpk) solver and displays the optimal solution in its interface, to enhance modelling of complex systems such as this. The optimization problem to be solved is represented in the ConcreteModel() object while the mathematical programming software used for calculating a solution is represented in the SolverFactory() object of the Pyomo application.

The mechanism of this control system is best described using the example of refinery blending process, where two types of crude oil are used to produce various products (gasoline, kerosene etc.). It is assumed that the production distribution from the refinery is different for the two types of crude oil. The goal is to determine the optimal combination of the two types of crude to buy per day, and the optimal revenue, under various conditions that one may encounter during the current COVID-19 pandemic.

The decision variables are the flow rates, in barrels per day (bbl./day) of the two types of crude oil and four products:

$$x_1 - crude\ 1$$

$$x_2 - crude\ 2$$

$$x_3 - \ gasoline$$

$$x_4 - \ kerosine$$

$$x_5 - \ fuel\ oil$$

$$x_6 - \ residual$$

The objective of this system is to maximize profit. It is defined as a linear objective function below, as the difference between income and costs:

$$Profit = Income - Raw\ Material\ cost - Processing\ cost$$

Linear relationships between the decision variables are formulated to define the parameters of income, raw material cost and processing cost.

$$Income = 72x_3 + 48x_4 + 42x_5 + 20x_6$$

$$Raw\ material\ cost = 48x_1 + 30x_2$$

$$Processing\ cost = 1x_1 + 2x_2$$

The final objective function to optimize is therefore:

$$f = \ - 49x_1 - 32x_2 + 72x_3 + 48x_4 + 45x_5 + 20x_6$$

The material balance equations are as follows:

$$Gasoline{:}\ x_3 = 0.8x_1 + 0.44x_2$$

$$Kerosine{:}\ x_4 = 0.05x_1 + 0.1x_2$$

$$Fuel\ Oil{:}\ x_5 = 0.10x_1 + 0.36x_2$$

$$Residual{:}\ x_6 = 0.05x_1 + 0.10x_2$$

The production limits are:

$$Gasoline{:}\ x_3 \leq 24,000$$

$$Kerosine{:}\ x_4 \leq 2,000$$

$$Fuel\ Oil{:}\ x_5 \leq 6,000$$

A schematic can then be developed to depict the associated finances as flow ins and flow outs.

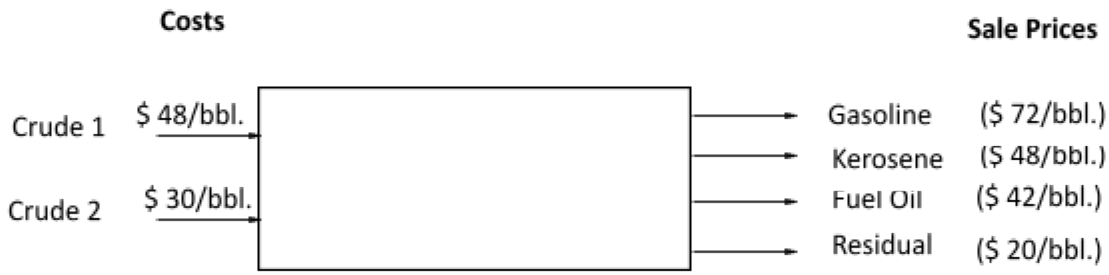

Figure 1: Refinery input and output schematic

Table 1 summarizes the system constraints. An additional constraint of nonnegativity is also included. All process variables must be greater than or equal to 0. This is because it is meaningless to have negative production rates.

Table 1: Data for the Refinery Feeds and Products

	Volume Percent Yield		Maximum allowable
	Crude 1	Crude 2	production (bbl./day)
Gasoline	80	44	24 000
Kerosene	5	10	2 000
Fuel Oil	10	36	6 000
Processing cost ($/bbl.)	1	2	

Figure 2 shows the base python code for programming the described system.

```python
# Import Pyomo
from pyomo.environ import *
import numpy as np

# problem data
FEEDS = ['Crude #1', 'Crude #2']
PRODUCTS = ['Gasoline', 'Kerosine', 'Fuel Oil', 'Residual']

# feed costs
feed_costs = {'Crude #1': 48,
              'Crude #2': 30}

# processing costs
processing_costs = {'Crude #1': 1.00,
                    'Crude #2': 2.00}

# yield data
product_yield = {('Crude #1', 'Gasoline'): 0.80,
                 ('Crude #1', 'Kerosine'): 0.05,
                 ('Crude #1', 'Fuel Oil'): 0.10,
                 ('Crude #1', 'Residual'): 0.05,
                 ('Crude #2', 'Gasoline'): 0.44,
                 ('Crude #2', 'Kerosine'): 0.10,
                 ('Crude #2', 'Fuel Oil'): 0.36,
                 ('Crude #2', 'Residual'): 0.10}

# product sales prices
sales_price = {'Gasoline': 72,
               'Kerosine': 48,
               'Fuel Oil': 42,
               'Residual': 20}

# production limits
max_production = {'Gasoline': 24000,
                  'Kerosine': 2000,
                  'Fuel Oil': 6000,
                  'Residual': 100000}
```

```python
# model formulation
model = ConcreteModel()

# variables
model.x = Var(FEEDS, domain=NonNegativeReals)
model.y = Var(PRODUCTS, domain=NonNegativeReals)

# objective
income = sum(sales_price[p] * model.y[p] for p in PRODUCTS)
raw_materials_cost = sum(feed_costs[f] * model.x[f] for f in FEEDS)
processing_cost = sum(processing_costs[f] * model.x[f] for f in FEEDS)

profit = income - raw_materials_cost - processing_cost
model.objective = Objective(expr = profit, sense=maximize)

# constraints
model.constraints = ConstraintList()
for p in PRODUCTS:
    model.constraints.add(0 <= model.y[p] <= max_production[p])
for p in PRODUCTS:
    model.constraints.add(model.y[p] == sum(model.x[f] * product_yield[(f,p)] for f in FEEDS))

results = SolverFactory('glpk').solve(model)
results.write()
if results.solver.status:
    model.pprint()

# display solution
print('\nProfit = ', model.objective())
print('\nDecision Variables')
print('x = ', model.x['Crude #1'](), model.x['Crude #2']())
```

Figure 2: The base python code for the simulation

The optimal combinations of crude 1 and crude 2 to buy, and the resultant profits are determined under various scenarios. The first optimization scenario to be considered reveals the sensitivity of the supply price by considering two scenarios.

1) The cost of crude 1 increases by 50% while the cost of crude 2 remains the same
   Combination = 0 bbl./day of crude 1 and 16,667 bbl./day of crude 2, Profit = $ 360,000

2) The cost of crude 1 remains the same and the cost of crude 2 increases by 50%.
   Combination = 30,000 bbl./day of crude 1 and 0 bbl./day of crude 2, Profit = $486,000

This analysis reveals that a steady supply of crude 1 is more important than a steady supply of crude 2. This is because crude 1 results in a 35% greater profit than crude 2.

The second scenario to be considered reveals the sensitivity of the demand price. It is assumed that the demand price is known before raw materials are bought and that the products will be bought. Each of the situations considered below has a 50% probability of being realized during this pandemic:

1) The sale price of gasoline is $36/bbl. and not the usual, $72/bbl. This situation could occur when there is a full lockdown imposed and so, the demand for gasoline decreases significantly.
   Combination = 0 bbl./day of crude 1 and 16,667 bbl./day of crude 2, Profit = $96,000

2) The sale price of gasoline is $108/bbl. and not the usual $72/bbl. This situation could occur when there is the pandemic is over and the demand for gasoline increases significantly.
   Combination = 26,207 bbl./day of crude 1 and 6,897 bbl./day of crude 2., Profit = $1,437,517.2414

The third scenario discusses a more realistic application of the control system, where the demand price is not known before raw materials are bought. It is however assumed that gasoline will either sell at $36/bbl. or $108/bbl. and that the Oil and Gas company in question, is risk neutral. The amount of crude oils to buy are therefore the same in both situations considered below. Normally, when gasoline sells at $72/bbl., the optimal combination is 26,206 bbl./day of crude 1 and 6896 bbl./day of crude 2.

1) The sale price of gasoline is $36/bbl.: Profit = -$290, 475
   If the company buys the usual amount of raw materials and the price of gasoline drops unexpectedly, the company makes a loss.

2) The sale price of gasoline is $108/bbl.: Profit = $1,437,455
   If the company buys the usual amount of raw materials and the price of gasoline increases, the company makes a great profit.

The final scenario is similar to that of the third scenario but considers a situation whereby the company in question, is risk averse. As such, the amount of crude oils bought must ensure that a loss is not made regardless of how the price of gasoline varies. As such, it is best to devise the optimal conditions that will maximize the worst-case profit, which occurs when gasoline sale prices are lowest ($36/bbl.).

Combination = 0 bbl./day of crude 1 and 16,667 bbl./day of crude 2, Profit = $96,000

Conclusion

The determination of a threshold ensures that manufacturers have a robust framework that ensures they do not make a loss, regardless of the price at which their products are being sold. The system proposed can be adjusted for the many companies that are repurposing their supply lines to make medical grade masks, for instance, during the pandemic.

Such a system will help companies engineer visibility into outbound goods as they are manufactured, assembled and shipped to customers through the transportation network, which is the challenge that they are facing today.