THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF AGRICULTURAL AND BIOLOGICAL ENGINEERING

TECHNO-ECONOMICAL ANALYSIS OF A DAIRY COW FARM-SCALE ANAEROBIC DIGESTION SYSTEM

HUNTER PORCANO
SPRING 2024

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Biological Engineering
with honors in Biological Engineering

Reviewed and approved* by the following:

Juliana Vasco-Correa
Assistant Professor of Agricultural and Biological Engineering
Thesis Supervisor

Ali Demirci
Professor of Agricultural and Biological Engineering
Honors Adviser

* Electronic approvals are on file.

ABSTRACT

This research investigates the imperative to promote bio-based sustainable practices in global agricultural systems, with a focus on the significant contribution of dairy operations to annual greenhouse gas emissions in the United States. There is a need for techno-economic analysis to inform decision-making among dairy farmers regarding the integration of anaerobic digestion systems. The study highlights the potential of renewable natural gas to mitigate methane emissions from dairy operations while providing renewable fuel. Through modeling and evaluation of various scenarios, including herd sizes (1000, 2000, and 5000 heads) and switchgrass availability, the research aims to provide valuable insights into the feasibility and profitability of anaerobic digestion on Pennsylvania dairy cow farms. This study found that anaerobic digestion systems were more profitability on larger farms. Factors such as natural gas selling price and biogas yield from manure had the largest impact on profitability. Anaerobic digestion systems may have to be tailored to fit the Pennsylvania dairy industry to increase feasibility.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to express my gratitude to my thesis supervisor, Dr. Juliana Vasco-Correa, for their unwavering support, guidance, and encouragement throughout this journey. Their expertise, mentorship, and genuine interest in my research have been invaluable in shaping this thesis and my academic growth. I would also like to thank my honors advisor, Dr. Ali Demirci, for their unwavering commitment to pushing me beyond my comfort zone and holding me to high standards. I would also like to thank my colleagues, Camilla Valderrama and Elmin Rahic, for their assistance and support throughout this research. I am thankful for their mentorship and the valuable lessons learned along the way.

I would also like to send a huge thank you to my friends and family for all the reassurance and support throughout this journey. To my parents, Diana and Anthony Porcano, thank you for always saying “you got this” in the moments I needed it most. To my best friends, Paula Szymanowicz and Ashley Ohstrom, thank you for being the best support system any girl could dream of.

This work is supported by the Pennsylvania State University and USDA NIFA Project #PEN04848 Accession #7005952, C-CHANGE project and the Department of Agricultural and Biological Engineering at Penn State University.

Chapter 1

Introduction

There is a call to action to promote bio-based sustainable practices on agricultural systems globally. Dairy operations contribute approximately 2.5% to the annual greenhouse gas (GHG) emissions in the United States, positioning them as one of the major contributors to industrial GHG emissions (Wendt et al., 2016). Figure 1 illustrates the regions where dairy farming is prominent across the United States, with Pennsylvania ranking among the top 10 states in terms of dairy production (Anaerobic Digestion on Dairy Farms | US EPA, 2024).

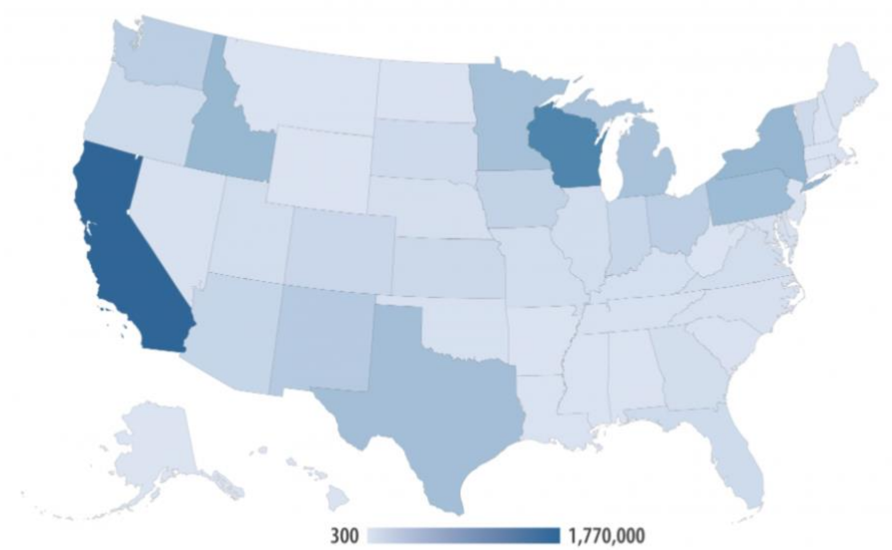

Figure 1: United States Dairy Cow Population, 2018 (U.S. EPA GHG Inventory of Greenhouse Gas Emissions and Sinks: 1990-2018).

There is an increasing demand for techno-economic analysis research to advance sustainability in the agricultural industry (Tan et al., 2022). There exists considerable research on integrating anaerobic digestion into dairy cow farms; however, a notable portion of this research lacks essential cost data necessary for farmers to make informed decisions (Nleya et al., 2023). To address this, there is a need to model anaerobic digestion systems on dairy cow farms to study the feasibility and profitability of these projects.

Chapter 2

Literature Review

2.1 Pennsylvania Dairy Industry

With a dairy cow population totaling approximately 9.4 million, the milk industry in the United States holds significant economic importance. Pennsylvania ranks second in the northeastern region for the highest number of dairy cows (USDA ERS - Dairy Data, 2024.).

The management styles and herd sizes vary widely among dairy cow farms in Pennsylvania. This leads to differences in manure collection, storage, and nutrient recovery across the state. Most farms grow crops such as corn, hay, and soybeans to produce cow feed (Holly et al., 2019). Homegrown feed production is an important cost cutting aspect of farms. The majority of farmers endeavor to utilize every aspect of their land efficiently for dairy production. The varying management strategies and land layouts make it difficult to standardize potential biogas collection systems. Many dairy operations also rely on the use of nonrenewable energy sources. Generating electricity on the farm could help make energy usage more sustainable (von Keyserlingk et al., 2013).

The dominant dairy cow manure management system in Pennsylvania is daily spreading of manure on to fields of crops (Niles & Wiltshire, 2019). Cow manure is considered a valuable fertilizer source. The prevalent method of dairy cow manure storage involves an open pit, followed by a significant portion of farms not employing any storage method at all (Holly et al., 2019a). It is logical for larger farm operations to utilize mechanized systems for manure collection.

2.2 Anaerobic Digestion

Anaerobic digestion (AD) encompasses the breakdown and stabilization of organic materials in the absence of oxygen by microbial organisms, resulting in the production of biogas and biomass. It is extensively employed within the wastewater treatment sector as an efficient method for water purification. Numerous agricultural wastes, such as animal manure, are well-suited for AD due to their abundance of readily biodegradable materials. (Chen et al., 2008).

When dairy cow manure undergoes AD, valuable biogas is released mainly in the form of methane and carbon dioxide. Manure that is spread on to a field as a fertilizer will eventually undergo anaerobic digestion naturally. This means that released biogas is unable to be used by the farmer (Nleya et al., 2023). AD allows biogas to be collected from the manure, while still producing a nutrient full digestate product that can be utilized by farmers. AD significantly lowers the manure pathogen levels making the digestate a better product for field spreading (Foxen & Fitzgerald, 2019). Utilizing AD to capture biogas from the dairy cow effluent offers a promising opportunity to harness a valuable energy resource and presents significant income-generating potential.

Perennial grasses are becoming popular for the carbon sequestration potential they hold (Agostini et al., 2015). Utilizing lignocellulosic materials for anaerobic digestion in the United States offers numerous advantages. Firstly, it enhances biogas output by supplementing nutrient-rich animal and human waste with a valuable carbon source. Additionally, it creates a potential interim market for energy crops, paving the way for future large-scale production of cellulosic ethanol (Jin et al., 2012).

Figure 2 illustrates the comprehensive overview of the potential anaerobic digestion (AD) setup on a Pennsylvania dairy cow farm, depicting the input of dairy cow manure and switchgrass into the digester. The outputs of the digester include valuable products such as biogas and digestate.

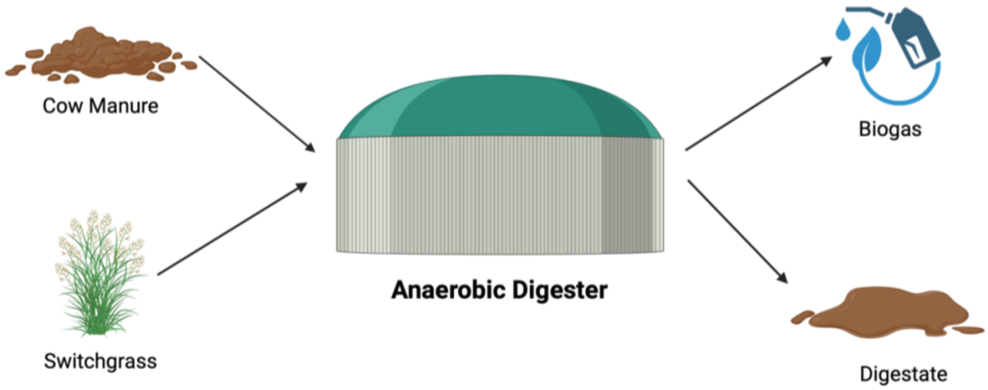

Figure 2: Input and Outputs of Anaerobic Digestion on Dairy Cow Farms.

The biogas output from the anaerobic digester can be upgraded to become renewable natural gas (RNG). Redirecting dairy methane emissions towards energy consumption not only displaces the combustion of fossil gas but also results in the emission of carbon dioxide, which has a significantly lower climate impact compared to methane. RNG has the potential to offer significant amounts of renewable fuel for the transportation sector, pipeline gas, or renewable electricity(Parker et al., 2017). The California Low Carbon Fuel Standard program serves as the key state-run initiative driving the demand and market value of RNG (Foxen & Fitzgerald, 2019).

2.3 Techno-Economical Analysis

A high capital cost is expected with the implantation of anaerobic digestion and gas upgrading technologies on to a farm (Wilkie, 2005). It is essential to gather information for dairy farmers to make informed decisions to lower the potential of economic risk. The techno-economic analysis (TEA) of biorefineries offers crucial insights into the economic feasibility, technological challenges, and business risks associated with the biomass-to-fuel value chains (Cortes-Peña et al., 2020).

Chapter 3

Goals and Objectives

There is potential for Pennsylvania dairy cow farms to increase revenue streams with the incorporation of anaerobic digestion. The production of renewable natural gas allows farms to optimize resource utilization and promote sustainable practices. The newfound accessibility of carbon credits has sparked a widespread surge in interest across the nation towards the reduction of carbon emissions.

3.1 Goal

The goal of the research is to evaluate financially viable methods for integrating anaerobic digestion into Pennsylvania dairy cow farms. A TEA model will be prepared using an open-source python program called BioSTEAM. Scenarios with differing manure and switchgrass input amounts will be assessed using the model. The program will predict profit yields based on specific parameters such as resource availability, biogas conversion rates, natural gas prices, etc. The findings will provide valuable insights to steer future research endeavors aimed at investigating practical options for implementing anaerobic digestion systems among dairy cow farmers in Pennsylvania.

3.2 Objectives

Objective 1: Measure the impact of dairy cow herd size and switchgrass availability on the profitability of renewable gas production.

Objective 2: Assess the parameters that effect profitability of anaerobic digestion systems the most.

Chapter 4

Methodology

4.1 Model Development

The objective of this research is to generate a model that can be used as a tool to make informed decisions on the incorporation of anaerobic digestion systems on to Pennsylvania dairy cow farms. A TEA model was generated with python code using the open-source platform called BioSTEAM. The platform was designed to assess emerging technology under uncertainty. (Cortes-Peña et al., 2020)

The program seeks to model an anaerobic digestion system process for a Pennsylvania dairy cow farm of slightly above-average size. As shown in Figure 3, the system begins with the initial collection of switchgrass and manure on a farm and ends with the production of RNG, solid digestate, and liquid effluent. The RNG is sold by the farmer for profit. The process includes storage of manure, storage and shredder for switchgrass, pumps, anaerobic digester, gas upgrading system, and a screw press for solid and liquid separation. The detailed process flow can be seen in Figure 3. It is assumed equipment will operate in a consistent way throughout a given year.

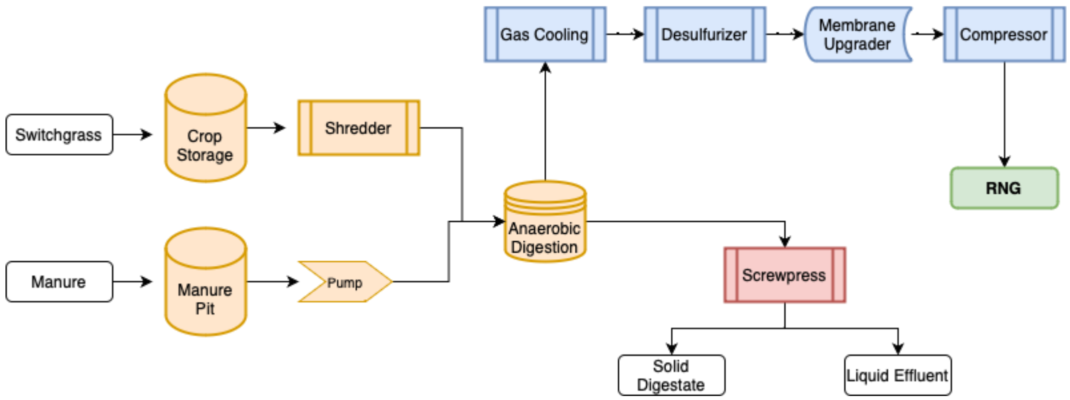

Figure 3: Anaerobic Digestion System on a Dairy Cow Farm Process Flow Diagram.

4.2 Modelling the Pennsylvania Dairy Cow Farm

Technical parameters set for the farm model were obtained from literature review of the Pennsylvania dairy industry. The model program that can be altered for different farm sizes. The main two inputs a user can customize is the number of cows on the farm and the annual yield of switchgrass. The program will construct the model parameters to fit the size of the input variables. For example, the size of the digester generated with the model will be based off the amount of manure and switchgrass available to load.

The base case scenario for the TEA is a 1000 head cow farm. The model was used to evaluate a 1000, 2000, and 5000 head dairy farm. The baseline farm size (1000 heads) was also modelled with 10% of land dedicated to switchgrass (estimate annual yield of 204240 kg). The four scenarios are laid out in Table 1. The capital cost, operational cost, and cash flows were evaluated for each scenario.

Table 1: TEA Model Scenarios.

	Number of Cows	Annual Switchgrass Yield (kg)
Scenario 1	1000	0
Scenario 2	1000	204240
Scenario 3	2000	0
Scenario 4	5000	0

The characteristics needed to assess biogas yields from the manure and switchgrass input were obtained from literature. Table 2 shows the specifications set for each input.

Table 2: Technical Parameters of Farm Inputs.

Input Type	Parameter	Value	Unit	Reference
Dairy Cow Manure	Production Rate	68	kg/day/cow	(Wendt et al., 2016)
	Density	1000	kg/m^3	(Wang et al., 2019)
	VS	84.6	% of TS	(Mcvoitte & Clark, 2017)
	MC	92	%	(Wilkie, 2005)
	Methane Yield	215	ml/g VS	(Abdallah et al., 2018)
Switchgrass	Production Rate	204240	kg/year	(Holly et al., 2019b)
	Density	150	kg/m^3	(Lam et al., 2008)
	VS	96.7	% of TS	(Jin et al., 2012)
	MC	30.13	%	(Jin et al., 2012)
	Methane Yield	300	ml/g VS	(U.S. Department of Energy, 2016)

4.3 Modelling the Anaerobic Digestion Process Flow System

The modeling and economic estimations were conducted using the default settings in version 2.28.0 of the BioSTEAM software, developed in Python v3.11 (Python Software Foundation, Wilmington, DE, USA, 2018), unless specified otherwise (Cortes-Peña et al., 2020). The models and economic estimations were adapted from the Bioindustrial-Park GitHub repository of examples provided by the software developers (Cortes-Peña et al., 2024). Therefore, only the variations in modeling and integration of new models are elaborated in detail below.

4.3.1 Anaerobic Digestion System

The anaerobic digester unit is built from the BioSTEAM continuous stir tank reactor unit. The biogas conversion reactions were modified so that gas yield came from values found from literature. The methane and carbon dioxide reaction yields are calculated based on the input of volatile solids in the program. Table 3 shows other technical parameters in place for the anaerobic digester unit. Calculations were done to assess heat loss by making assumptions on the materials used for the digester.

Table 3: Technical Parameters of Anaerobic Digester Unit.

Parameter	Value	Unit	Reference
Hydraulic Retention Time	25	Days	(Wilkie, 2005)
Temperature	35	℃	(Wilkie, 2005)
Influent MC	92	%	(Wilkie, 2005)
Cap Factor	0.9	-	-

4.3.2 Biogas Upgrading System

The biogas undergoes purification by passing through a gas membrane filtration system. The membrane filter is built off the splitter unit in BioSTEAM. The splitter is mapped to have the methane separate from the water and carbon dioxide in the biogas.

4.4 Economic Parameters

Most of the economic estimations mirrored those found in the original BioSTEAM software (Seider et al., 2017). However, the economic estimation parameters for the anaerobic digester, membrane filtration system, and screw press were added to the original software.

The capital cost estimation of the anaerobic digester was based off a cost sheet in a biogas opportunities progress report released by the USDA, EPA, and DOE (USDA et al., 2015). Electricity, heat requirements, and heat loss are calculated based off the input values of manure and switchgrass. The capital cost estimation of the gas membrane filtration system was based off a TEA of biogas membrane filtration for farm scale (Deng & Hägg, 2010). The capital cost estimation of the screw press was based off a product specification sheet (*Dairy Farm Waste Manure Dewatering Machine/Screw Press Cow Dung Slurry Separator/Cow Dung Cleaner Animal Manure Dryer*, 2024). The other costs of the gas membrane filtration and screw press system were estimated using original software.

Based on literature review, other major assumptions important to the TEA are listed below.

Major Cost Assumptions:

- Manure is free for the farmer to use.
    - Justification: Most Pennsylvania dairy operations utilize manure for other in-house farm operations. Even though the manure goes through anaerobic digestion, solid digestate and liquid effluent are valuable products to the farmer to continue to use for the same functions.
- The expense of labor accounts for only one additional operator being hired to work the digester system.
    - Justification: Due to the scale of the AD system, it is assumed that the farmer would utilize current employees for some operation necessities.

A sensitivity analysis was performed to assess how natural gas prices, biogas yields, and cost of equipment effect the profitability of the system. The minimum and maximum values chosen for analysis were obtained from literature.

Chapter 5

Results and Discussion

5.1 Scenario Comparisons

The net present values (NPV) for a 20-year facility lifetime can be seen in Figure 4 for the baseline scenario with and without switchgrass incorporated. The NPV is positive for both cases, but considerably low considering that large capital investments required for the facility. The NPV for the 1000 head scenario is $113,443 without the incorporation of switchgrass and $803,394 with switchgrass added. The inclusion of switchgrass increased the NPV of the baseline scenario. This suggests that the incorporation of switchgrass could be advantageous to farmers to increase revenue. More research is needed to fully assess the effect switchgrass has on the profitability of AD on dairy cow farms in Pennsylvania.

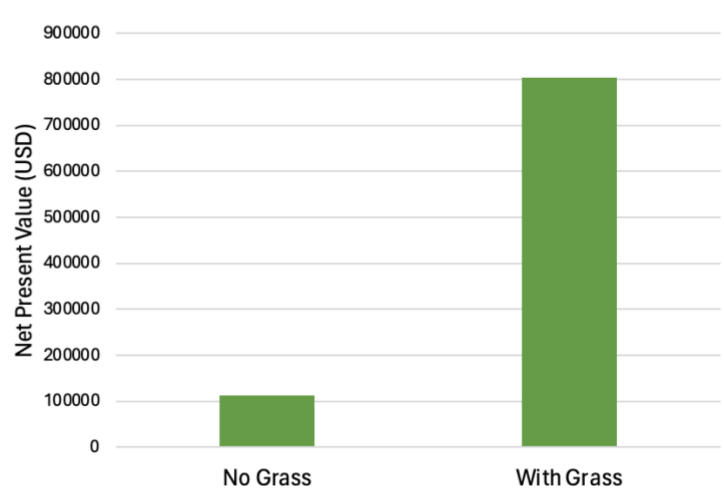

Figure 4: Net Present Value of Baseline Scenario with and without Switchgrass (20 Years Facility Lifetime).

The net present values for a 20-year facility lifetime can be seen in Figure 5 for different herd sizes. This figure shows how the difference in herd size heavily impacts the profitability of the AD system. The NPV for the 2000 head scenario is $3,415,153 for the 2000 head scenario and $15,142,432 for the 5000 head scenario.

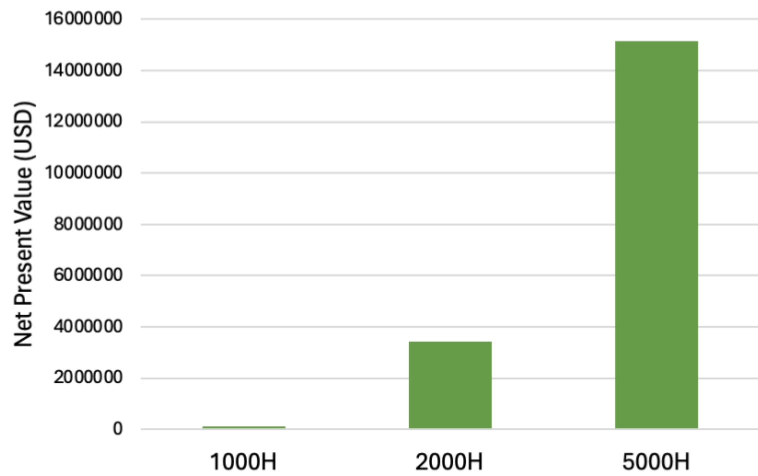

Figure 5: Net Present Value of Herd Size Scenarios (20 Years Facility Lifetime).

Figures 6 and 7 show the purchase cost summaries for the baseline scenario with and without switchgrass incorporated. There is not a large difference between the two cost breakdowns. There is a 4% purchase cost increase to add switchgrass to the farm. The purchase cost for the anaerobic digester is ~60% of the total cost for each scenario making it the largest expense. The boiler used for heating requirements is the second largest expense at ~20% of the total purchase cost. Figures 8 and 9 show the purchase cost summary of the 2000 and 5000 head farm scenarios. There is no significant difference between the distribution of equipment costs. The anaerobic digester and the heating equipment continue to be the largest expenses for these scenarios.

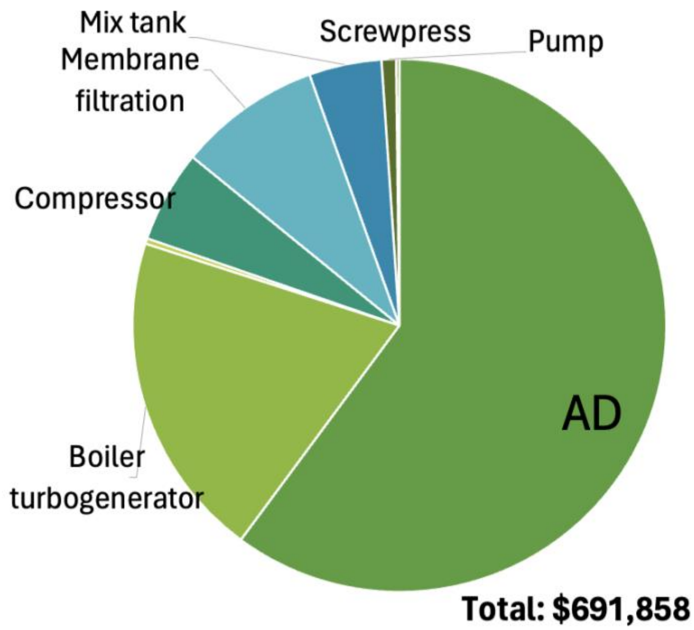

Figure 6: Purchase Cost Summary of 1000 Cow Farm.

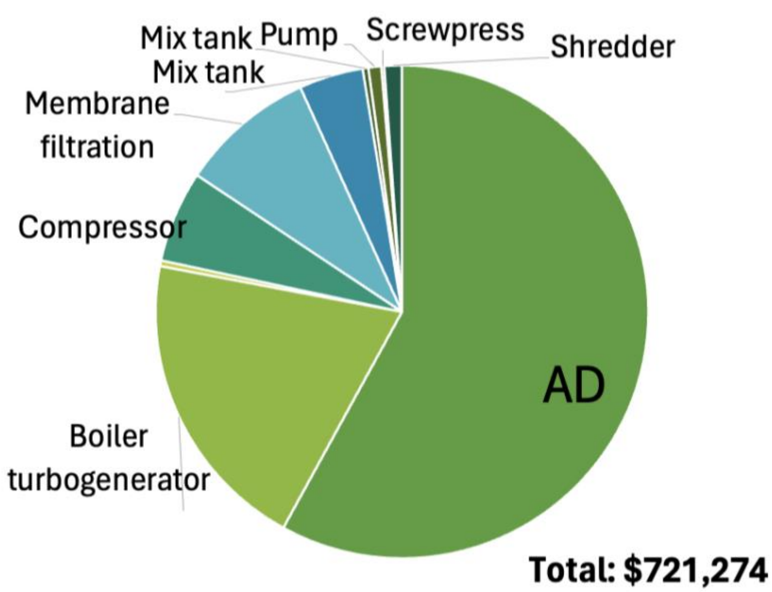

Figure 7: Purchase Cost Summary of 1000 Cow Farm with Switchgrass.

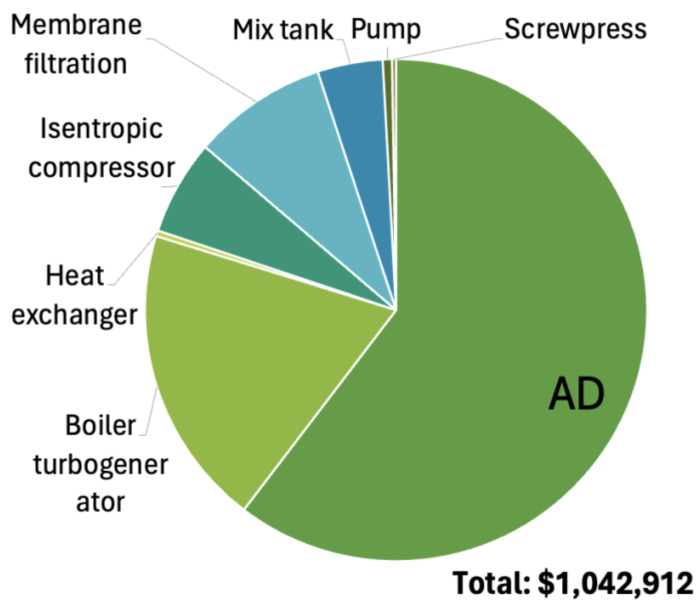

Figure 8: Purchase Cost Summary of 2000 Head Farm.

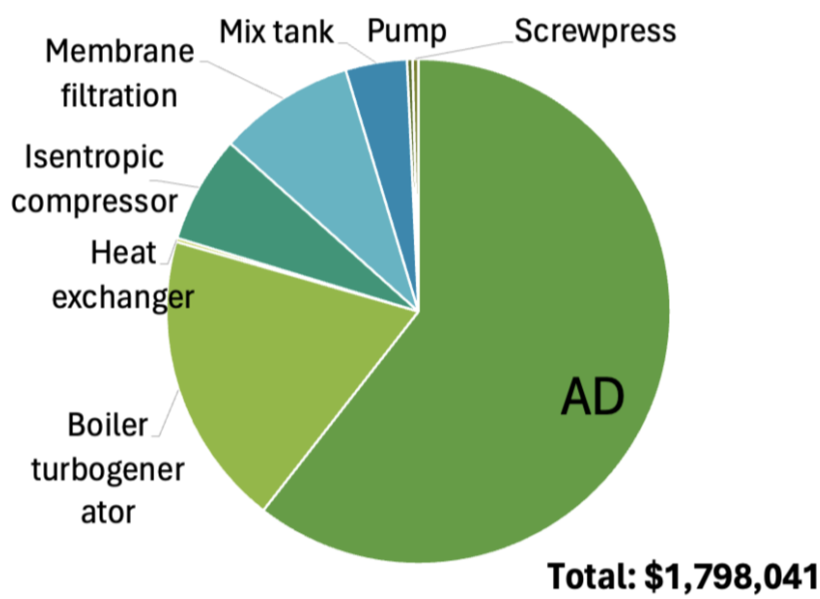

Figure 9: Purchase Cost Summary of 5000 Head Farm.

5.2 Sensitivity Analysis

A sensitivity analysis was conducted to assess the influence of various variables on the model. The selected variables were those with the highest degree of uncertainty. Most of the low and high values for these variables listed in Table 4 were derived from a review of the literature. The low and high values for the cost of the digester and gas upgrading system were varied by changing the base cost by 30%.

Table 4: Sensitivity Analysis Variable Change.

Variable	Base	Low Value	High Value	Units	Reference
Natural Gas Selling Price	1.64	1.20	2.08	USD/kg	(U.S. Energy Information Administration (EIA), 2023)
Biogas Yield from Manure	215	185	245	ml/g VS	(Abdallah et al., 2018)
Base Cost of Digester	418,333	302,219	534,447	USD	-
Base Cost of Gas Upgrade System	64,001	44,801	83,202	USD	-
Biogas Yield from Switchgrass	300	270	330	ml/g VS	(U.S. Department of Energy, 2016)

Figure 10 shows the tornado diagram for the sensitivity analysis. The selling price of the natural gas generated has the largest impact on the profitability of the anaerobic digestion system. The price of natural gas has a direct relationship with the NPV. For the base case scenario, varying the natural gas price from $1.20-$2.08 per kg changed the NPV from -$1.64 -

$2.64 million. Small fluctuations in fuel cost can have drastic impact on the feasibility of AD on PA dairy farms.

The biogas yield from manure had the second largest impact on the NPV of the baseline scenario. Varying the manure biogas yield from 185 to 245 ml/g VS changed the NPV from -$0.1-$1.7 million. Biogas yield can fluctuate easily with a variety of outside conditions. It will be important to monitor this value closely when implementing AD systems on to a farm. As shown in earlier figures, the anaerobic digester is the largest purchase expense. Changing the base cost of the digester by ±30% caused the NPV to change from $0.08-$1.7 million. The cost of the digester and the gas upgrading system have an inverse relationship with the NPV. Variations in the cost of the gas upgrading system as well as the biogas yield from switchgrass did have a large effect on the NPV.

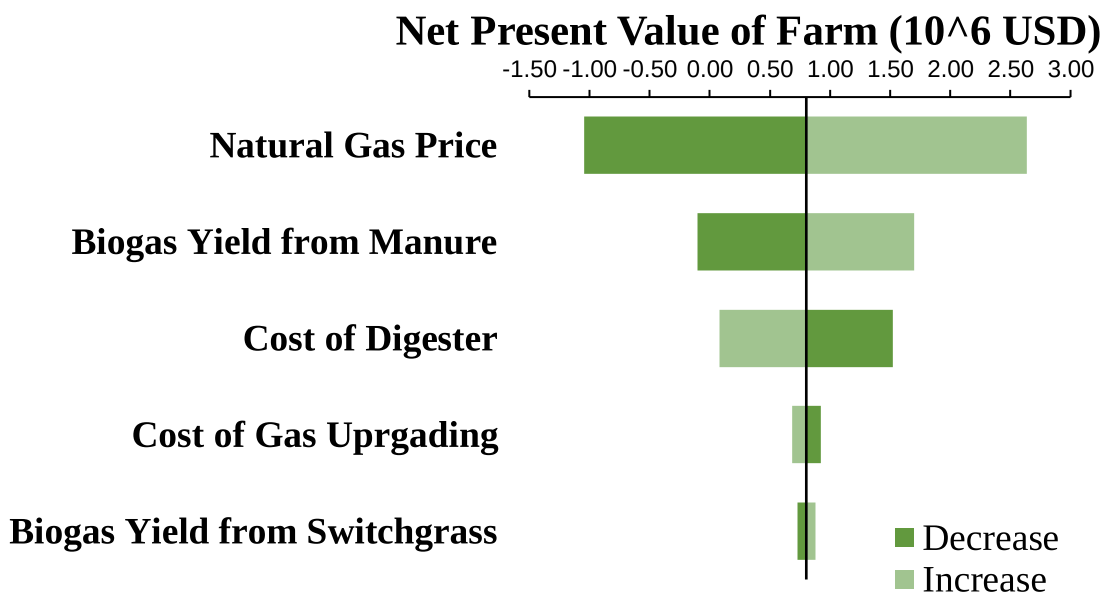

Figure 10: Sensitivity Analysis of 1000 Cow Farm with Switchgrass.

Chapter 6

Conclusion

Through comprehensive techno-economic analysis, this research aims to evaluate financially viable methods for integrating anaerobic digestion onto Pennsylvania dairy cow farms. By assessing scenarios with varying herd sizes and switchgrass availability, the study endeavors to provide valuable insights to guide future research and implementation efforts.

There was not a desirable profitability found for incorporating an AD system on to a 1000 head dairy cow farm with the model estimations. However, good profitability results were found for 2000 and 5000 head dairy cow farms. The purchase cost summary did not vary much between scenarios. The anaerobic digester was the largest purchase expense for the system. To fit the uniqueness of the Pennsylvania dairy industry, AD systems will have to be tailored for the most profitability.

The fluctuation of natural gas prices will play a major role in the profitability of farm scale AD systems. The higher the selling price of RNG, the more profit the farmer will make. Selling the natural gas at a premium due to the sustainable methods to generate the energy may make these projects more feasible. The biogas yield from manure was the second largest factor that could impact profitability. It will also be critical for farmers to maximize biogas yields from manure.

Further investigation is warranted to determine the optimal integration of AD systems into Pennsylvania dairy cow farms. The potential benefits of incorporating switchgrass into these systems pose significant questions. Collaboration between dairy and crop farms could facilitate the implementation of AD systems, while communal digesters shared among multiple farmers might offer greater economic viability. It is essential to model and evaluate these scenarios thoroughly.

The Pennsylvania dairy industry's diverse management styles and herd sizes present challenges in standardizing biogas collection systems, accentuating the necessity for tailored solutions. Anaerobic digestion emerges as a promising avenue, offering opportunities to capture valuable biogas from dairy cow effluent while simultaneously producing nutrient-rich digestate for agricultural use. Ultimately, the pursuit of sustainable practices in dairy farming holds immense potential to optimize resource utilization, increase revenue streams, and foster resilient agricultural communities.

Appendix A

Python Script

```
import biosteam as bst
from biosteam import settings

bst.nbtutorial() # Ignore warnings and reset local BioSTEAM preferences
# Farm Model
cows = int(input("Enter the number of cows: ")) # Number of cows on farm
manure_per_cow = 68 # Avg amount in kg of manure produced by one dairy cow per day
manure_per_day = cows*manure_per_cow # Avg manure in kg per day
# Manure Characteristics
manure_water = .92 # Manure water content %
manure_vsolids = (1-manure_water)*.846 # Manure volatile solids %
manure_ash = 1-manure_vsolids # Manure ash content
manure_flowrate = manure_per_day/24 # Manure flow rate in kg/hr
manure_slurry_mc = .92 # Desired moisture content for influent manure slurry
manure_density = 998 # Assuming same as water (kg/m3)
# Grass Characteristics
grass_annual_yield = 204240 #Annual amount of switchgrass harvested in kg
grass_per_day = grass_annual_yield/365 # Avg switchgrass available per day in kg
grass_ash = .039 # Grass ash content %
grass_water = .3013 # Grass water content %
grass_vsolids = (1-grass_water)*.9677 # Grass volatile solids %
grass_flowrate = grass_per_day/24 # Avg grass in kg/hr
grass_density = 150 # Grass density (kg/m3)
# Reactor Variables
hrt_day = 22.5 # Desired Hydraulic Retention Time in days
hrt_hr = hrt_day*24 # Conversion to hours
cap_factor = 0.9 # Assumption of the capacity factor of the plant with employees on site
load_weight_per_day = manure_per_day+grass_per_day # Amt of manure and grass available to load everyday (kg)
digester_flowrate = load_weight_per_day/24 #kg/hr
digestate_density = ((manure_density*manure_per_day)+(grass_density*grass_per_day))/load_weight_per_day
digester_volume = ((load_weight_per_day*hrt_day)/digestate_density)*1.428 # Digester volume based off amount off manure and grass; takes into account that reactor can only be 70% filled
```

```
# Influent and Flow Characteristics
influent_mc = 0.92  #This is the desired moisture content of the combined feedstock being fed into the digester.
water_flowrate = ((influent_mc * grass_flowrate) + (influent_mc * manure_flowrate) - (grass_flowrate * grass_water) - (manure_flowrate * manure_water))/(1-influent_mc)  #This calculates the necessary water flow to reach the desired moisture content for feeding the digester.
organic_load = (((grass_flowrate*grass_vsolids) + (manure_flowrate*manure_vsolids)) / (grass_flowrate + manure_flowrate + water_flowrate)) * 100     #This calculates the %VS being fed into the digester. This variable is used for my regression equations.
g_m_ratio = (grass_flowrate*grass_vsolids) / (manure_flowrate*grass_vsolids)  #Calculates the ratio of grass to manure on a VS basis. Used for my regression equations.

# CHECK OVER THESE NUMBERS
chemicals = bst.Chemicals(
   [bst.Chemical('manure', #Dairy Cow Manure
              Cp=2.75, #heat capacity (kJ/kg)
              rho=1000, #density (kg/m3)
              default=True,
              search_db=False,
              phase='s',
              MW=1.),
   bst.Chemical('solids', #Volatile Solids
              Cp=1.100,
              rho=1000,
              default=True,
              search_db=False,
              phase='s',
              MW=1.),
   bst.Chemical('ash', #Non-Volatile Solids
              rho=1540,
              Cp=0.37656,
              default=True,
              search_db=False,
              phase='s',
              MW=1.),
   bst.Chemical('CH4',
```

```
            rho=0.634,
            Cp=2.2,
            default=True,
            search_db=True,
            phase='g'),
    bst.Chemical('CO2',
            rho=1.7067,
            Cp=0.84,
            default=True,
            search_db=True,
            phase='g'),
    bst.Chemical('O2',
             search_db=True),
    bst.Chemical('Water',
             search_db=True)]
)
bst.settings.set_thermo(chemicals) #sets the thermodynamic properties listed above

'''This is biogas and methane conversions '''
#Calculates Biogas yield in mL/gVS
biogas_volume = 380*(((manure_flowrate*1000)*(manure_vsolids)))+460*(((grass_flowrate*1000)*(grass_vsolids)))
#Calculates Methane yield in mL/gVS
CH4_volume = 215*(((manure_flowrate*1000)*(manure_vsolids)))+300*(((grass_flowrate*1000)*(grass_vsolids)))

#Calculates CO2 yield in mL/gVS based on the difference between biogas and methane.
CO2_volume = biogas_volume - CH4_volume

#Methane yield as a proportion of the mass of total VS fed
CH4_mass_yield = CH4_volume/biogas_volume

#CO2 yield as a proportion of the mass of total VS left after methane
CO2_mass_yield = CO2_volume/biogas_volume

# Utility Costs
CEPCI = 750 # CEPCI: 2021
bst.settings.CEPCI = CEPCI
```

```
bst.settings.electricity_price = 0.1105 # setting the electricity price per kWh, coming from the grid

# Heating and Cooling Parameters
steam_utility = bst.settings.get_agent('low_pressure_steam') #Using low pressure steam because our heating needs
are not that high.
bst.settings.heating_agents = [steam_utility]
steam_utility.heat_transfer_efficiency = 1.0   ##This is heat transfer efficiency. The boiler already takes this into
account, so it can be ignored.

cooling_utility = bst.settings.get_agent('chilled_brine')   #Chilled brine was used for the gas cooling stage in order to
get the gas cool enough to remove all the moisture.
bst.settings.cooling_agents = [cooling_utility]
cooling_utility.heat_transfer_efficiency = 0.8

# Raw material price (USD/kg)
price  = {'manure': 0, #The liquid digestate serves as a fertilizer, we are assuming we are not losing money by not
selling manure as is
         'Water': .00011, #Cost of water per kg from the US department of energy
         'grass': .066} #Cost of switchgrass per kg from billion ton report

## Importing packages needed
from biosteam import units, Stream
import numpy as np
import pandas as pd
from biosteam.units.decorators import cost
from biosteam.units import Pump
import thermosteam as tso
import flexsolve as flx
from math import ceil
from biosteam import Splitter

## Begin Unit Operations

# Setting streams
manure = bst.Stream(
    'manure',
```

```
    total_flow=manure_flowrate,
    solids=manure_vsolids,
    water=manure_water,
    ash=manure_ash,
    units='kg/hr',
    price=price['manure'],
    T=30 +273.5 #need temp?
)

grass = bst.Stream(
    'grass',
    total_flow=grass_flowrate,
    solids=grass_vsolids,
    water=grass_water,
    ash=grass_ash,
    units='kg/hr',
    price=price['grass'],
    T = 20 + 273.15
)
RNG = bst.Stream('RNG', price=1.64) #Stream for natural gas (USD/kg)

## Setting up unit operations

# Manure Pit Storage
class MixTank(bst.Unit): pass
Manure_Pit = units.MixTank('Manure_Pit', manure, outs='manure')

# Pump for manure into digester
Manure_Pump = units.Pump('Manure', Manure_Pit-0, outs='manure')

# Switchgrass Storage
class MixTank(bst.Unit): pass
Switchgrass_Pit = units.MixTank('Switchgrass_Pit', grass, outs='grass')

# Shredder unit: Shreds large biomass particles to smaller sizes. Assuming the base Biosteam system.
Shredder = units.Shredder('Switchgrass', Switchgrass_Pit-0, outs='shredded_grass')
```

```
# Anaerobic Digestion. This AD unit is build off of a continuous CSTR. Because of that, we have to adjust: 1)cost data, 2)heating needs, and 3)mixing energy needs.

# Agitation electrical needs. Assuming 5.65 kWh/MT dry solids
kW_MT = (5.65*((grass_flowrate * (grass_ash + grass_vsolids)) + manure_flowrate * (manure_ash + manure_vsolids)))/1000

@cost('Reactor volume', 'Reactors', CE=556.8, S=2200, cost = 347000, n=0.6, BM=.76) #Cost data from USDA and EPA and DOE
class AD(bst.CSTR): # most parameters are not yet set
    _N_ins = 2
    _N_outs = 2
    T_default = 32. + 273.15
    P_default = 101325.
    tau_default = hrt_hr
    V_max_default = digester_volume # generates the max digester size in m3
    V_wf = 0.7 #Working volume of your anaerobic digester
    kW_per_m3_default = kW_MT #This is electricity needs for your digester

    def _design(self):
        Design = self.design_results
        ins_F_vol = sum([i.F_vol for i in self.ins if i.phase != 'g'])
        V_total = ins_F_vol * self.tau / self.V_wf
        N = ceil(V_total/self.V_max)
        if N == 0:
                V_reactor = 0
        else:
                V_reactor = V_total / N
        Design['Reactor volume'] = V_reactor
        self.parallel['Reactors']=N
        self.parallel['agitator']=N

        #Heating needs for feedstocks.
        #Heating needs for 9°C -> 37°C (MJ/hr). 9°C was chosen because average well water
        #temperature is 9°C in Pennsylvania.
```

```
temp_difference = 37-9
Cp_manure = 2.75 #heat capacity (kJ/kg)
Cp_water = 4.18 #not set
Cp_crop = 1.35 #not set

Q_man = manure_flowrate * Cp_manure * (temp_difference)
Q_crop = grass_flowrate * Cp_crop * (temp_difference)
Q_water = water_flowrate * Cp_water * (temp_difference)
Q_inputs = (Q_man + Q_water + Q_crop) / 1000   #This is the total heat needed to heat substrates (MJ/hr).

#Heat loss through the walls, cover, and floor.
#Reactor Dimensions (above ground CSTR with impermeable cover and 30m diameter)
R_Radius = (digester_volume/(3.14*7))**(1/3) # Based off volume and 3.5:1 l to w ratio
R_Height = 7*(R_Radius)
R_Diameter = 2*R_Radius # meters
R_wall_area = 3.14159 * R_Diameter * R_Height
R_floor_area = 3.14159 * R_Radius**2

#Constants and dimensions for materials. L stands for Length. k is the thermal conductivity.
#Assuming 20 cm of concrete along base and walls. 10cm of polyurethane insulation.
L_BG = 4
k_BG = .0332

L_mem = 0.002
k_mem = 0.06

L_air = L_mem/2
k_air = 0.024

L_con = 0.2
k_con = 2.3

L_ins = 0.1
k_ins = 0.04

L_soil = 1
```

```
        k_soil = 1


        Q_BG = ((temp_difference) * (k_BG) * (((3.14159 * R_Diameter**2)/2) + (L_BG * 3.14159 * R_Radius**2))) *
0.0036   #Convective heat loss through biogas.
        Q_mem = (((temp_difference) * (3.14159 * R_Diameter * L_mem * 2)) / (L_mem / k_mem)) * 0.0036
#Conductive heat loss through membrane.
        Q_air = (((temp_difference) * (3.14159 * R_Diameter * L_air)) / (L_air / k_air)) * 0.0036
#Convective heat loss through air gap.
        Q_cover = Q_BG + Q_mem + Q_air                                                #Total heat loss from biogas
and cover.
        Q_wall = (((temp_difference) * R_wall_area) / ((L_con/k_con)  + (L_ins/k_ins))) * 0.0036
#Conductive heat loss through the wall.
        Q_floor = (((temp_difference) * R_floor_area) / ((L_con/k_con) + (L_soil/k_soil))) * 0.0036
#Conductive heat loss through the floor.
        Q_loss = Q_cover + Q_wall + Q_floor                                            #Total heat loss

        duty = Q_loss*N*1000 + Q_inputs*1000        #This is the heating duty for the digester system (kJ/hr).
        Design['Reactor duty'] = duty
        self.add_heat_utility(duty, T_in=15+273.15, T_out=37+273.15)  #The temperatures in here don't do anything
since the heat demand is set, but the line is still needed to run the code.
        kW = self.kW_per_m3_default / N
        if kW > 0: self.agitator = bst.Agitator(kW)


    def _setup(self):
        super()._setup()
        chemicals=bst.chemicals
        self.fermentation_reaction1 = bst.Reaction('solids -> CH4', 'solids', CH4_mass_yield, basis = 'wt')
        self.fermentation_reaction2 = bst.Reaction('solids -> CO2', 'solids', CO2_mass_yield, basis='wt')

    def _run(self): # Mainly source code
        vent, effluent = self.outs
        effluent.mix_from(self.ins, energy_balance=False)
        self.fermentation_reaction1(effluent)
        self.fermentation_reaction2(effluent)
```

```
        effluent.T = vent.T = self.T
        effluent.P = vent.P = self.P
        vent.phase = 'g'
        vent.empty()
        vent.receive_vent(effluent, energy_balance=False)


Digester = AD('Digest', [Manure_Pump-0, Shredder-0], outs=('CO2', 'digestate'))

## Gas Upgrading

#Gas cooling: This cools the gas to remove moisture.
#Standard gas cooling units seem to be cooling to ~5°C. I'm assuming 2°C for this.
Moisture_Removal = units.HXutility(ID='Moisture_Removal', ins = Digester-0, outs=('Dry_Biogas'), T=2+273.15,
rigorous=True, cool_only=True)

#Separating the remaining water because the gas cooling unit doesn't do it for us.
Gas_Scrubbing = units.Splitter('Gas_Scrubbing', ins=Moisture_Removal-0, outs=('Water','Dry_Gas'), split={'Water':
1.0, 'CH4': 0, 'CO2': 0})

#Gas Purification using membrane filtration. This filtration unit was build off of a splitter unit in biosteam.
__all__ = ('Membrane_Filtration',)
@cost('Flow rate', 'm3/hr', cost= 311218, CE=382, S=1000, n=0.6,kW=250, BM = 3)
class Membrane_Filtration(Splitter):
      _units = {'Flow rate': 'm3/hr'}
      def __init__(self, ID='', ins=None, outs=(), *, order=None, split,
                 P=None, approx_duty=True):
        Splitter.__init__(self, ID, ins, outs, order=order, split=split)
        self.P = None
        self.approx_duty = approx_duty

      def _run(self):
        Splitter._run(self)
        P = self.P
        if P is None: P = self.ins[0].P
        for i in self.outs: i.P = P
```

```
    def _design(self):
      self.design_results['Flow rate'] = flow = self._outs[1].F_mass
      if self.approx_duty:
          T = self.outs[0].T
          self.add_heat_utility(0 * flow, T)
          self.add_heat_utility(0 * flow, T)

Gas_Upgrading = units.Membrane_Filtration('Gas_Upgrading', Gas_Scrubbing-1,
                                  outs = ('RNG', 'CO2'),
                                 split=dict(CH4=0.9653,    #assuming 2% loss from digester and 1.5% loss from upgrading
                                        Water=0,
                                        CO2=0))


#Gas compressor for CH4 to pipeline. This unit was chosen because the standard one has a bug with the cost being
0
RNG_Compressor = units.IsentropicCompressor(ID='RNG_Compressor', ins=Gas_Upgrading-0, outs=RNG,
P=1.1e6)

@cost('Flow rate', units='lb/hr', CE = 750, S=8000,cost = 1780, n=1, BM=2.03)
class Screwpress(bst.SolidsSeparator):
    kWh_per_bmt = 20
    def _cost(self):
        self._decorated_cost()
        flow=self.ins[0]
        Solid_flow = flow.F_mass * 0.001 * (1-influent_mc)
        self.add_power_utility(self.kWh_per_bmt*Solid_flow)
  #The separation level below was determined based on experimental results.
Dewater = units.Screwpress('Dewater', Digester-1, outs=('solid_digestate', 'liquid_effluent'),
                     split=dict(ash=0.5,
                            solids=(0.7)*manure_flowrate+grass_flowrate))
                     # moisture_content=.92))



cost = bst.decorators.cost
```

```
__all__ = ('BoilerTurbogenerator',)

#these costs are from BioSTEAM
@cost('Work', 'Turbogenerator',
      CE=551, S=42200, kW=0, cost=9500e3, n=0.60, BM=1.8)
@cost('Flow rate', 'Hot process water softener system',
      CE=551, cost=78e3, S=235803, n=0.6, BM=1.8)
@cost('Flow rate', 'Amine addition pkg',
      CE=551, cost=40e3, S=235803, n=0.6, BM=1.8)
@cost('Flow rate', 'Deaerator',
      CE=551, cost=305e3, S=235803, n=0.6, BM=3.0)
@cost('Flow rate', 'Boiler',
      CE=1000, cost=8000000, kW=1371, S=22*3600, n=0.59, BM=1.5)
@cost('Ash disposal', 'Baghouse bags',
      CE=551, cost=466183. / 4363., n=1.0, lifetime=5)
class BoilerTurbogenerator(bst.Facility):
    ticket_name = 'BT'
    network_priority = 0
    boiler_blowdown = 0.03
    RO_rejection = 0
    _N_ins = 6
    _N_outs = 3
    _units = {'Flow rate': 'kg/hr',
              'Work': 'kW',
              'Ash disposal': 'kg/hr'}
    def __init__(self, ID='', ins=None,
                 outs=('emissions',
                       'rejected_water_and_blowdown',
                       'ash_disposal'),
                 thermo=None, *,
                 boiler_efficiency=None,
                 turbogenerator_efficiency=None,
                 side_steam=None,
                 agent=None,
                 other_agents=None,
                 natural_gas_price=None,
```

```
            ash_disposal_price=None,
            T_emissions=None,
            satisfy_system_electricity_demand=None,
            boiler_efficiency_basis=None,
    ):
    if boiler_efficiency_basis is None: boiler_efficiency_basis = 'LHV'
    if boiler_efficiency is None: boiler_efficiency = 0.80
    if turbogenerator_efficiency is None: turbogenerator_efficiency = 0.85
    if satisfy_system_electricity_demand is None: satisfy_system_electricity_demand = True
    bst.Facility.__init__(self, ID, ins, outs, thermo)
    settings = bst.settings
    self.boiler_efficiency_basis = boiler_efficiency_basis
    self.agent = agent = agent or settings.get_heating_agent('low_pressure_steam')
    self.define_utility('Natural gas', self.natural_gas)
    self.define_utility('Ash disposal', self.ash_disposal)
    self.boiler_efficiency = boiler_efficiency
    self.turbogenerator_efficiency = turbogenerator_efficiency
    self.steam_utilities = set()
    self.power_utilities = set()
    self.steam_demand = agent.to_stream()
    self.side_steam = side_steam
    self.other_agents = [i for i in settings.heating_agents if i is not agent] if other_agents is None else other_agents
    self.T_emissions = self.agent.T if T_emissions is None else T_emissions # Assume no heat integration
    if natural_gas_price is not None: self.natural_gas_price = natural_gas_price
    if ash_disposal_price is not None: self.ash_disposal_price = ash_disposal_price
    self.satisfy_system_electricity_demand = satisfy_system_electricity_demand
    self._load_components()

def _load_components(self):
    chemicals = self.chemicals
    if 'SO2' in chemicals:
        CAS_lime = '1305-62-0'
        if CAS_lime in chemicals or 'Ca(OH)2' in chemicals:
            if 'Ca(OH)2' not in chemicals:
                chemicals.set_synonym(CAS_lime, 'Ca(OH)2')
            self.desulfurization_reaction =  tmo.Reaction(
```

```
                'SO2 + Ca(OH)2 + 0.5 O2 -> CaSO4 + H2O', 'SO2', 0.92, chemicals
            )
            self._ID_lime = 'Ca(OH)2'
            return
        CAS_lime = '1305-78-8'
        if CAS_lime in chemicals or 'CaO' in chemicals:
            if 'CaO' not in chemicals:
                chemicals.set_synonym(CAS_lime, 'CaO')
            self.desulfurization_reaction =  tmo.Reaction(
                'SO2 + CaO + 0.5 O2 -> CaSO4', 'SO2', 0.92, chemicals
            )
            self._ID_lime = 'CaO'
            return

@property
def blowdown_water(self):
    return self.outs[1]

@property
def makeup_water(self):
    """[Stream] Makeup water due to boiler blowdown."""
    return self.ins[2]

@property
def natural_gas(self):
    """[Stream] Natural gas to satisfy steam and electricity requirements."""
    return self.ins[3]

@property
def ash_disposal(self):
    """[Stream] Ash disposal."""
    return self.outs[2]

@property
def natural_gas_price(self):
    """[Float] Price of natural gas, same as `bst.stream_utility_prices['Natural gas']`."""
```

```
        return bst.stream_utility_prices['Natural gas']

    @natural_gas_price.setter
    def natural_gas_price(self, new_price):
        bst.stream_utility_prices['Natural gas'] = new_price

    @property
    def ash_disposal_price(self):
        """[Float] Price of ash disposal, same as `bst.stream_utility_prices['Ash disposal']`."""
        return bst.stream_utility_prices['Ash disposal']

    @ash_disposal_price.setter
    def ash_disposal_price(self, ash_disposal_price):
        bst.stream_utility_prices['Ash disposal'] = ash_disposal_price

    def _run(self): pass

    def _load_utility_agents(self):
        steam_utilities = self.steam_utilities
        steam_utilities.clear()
        agent = self.agent
        units = self.other_units
        for agent in (*self.other_agents, agent):
            ID = agent.ID
            for u in units:
                for hu in u.heat_utilities:
                    agent = hu.agent
                    if agent and agent.ID == ID:
                        steam_utilities.add(hu)
        self.electricity_demand = sum([u.power_utility.consumption for u in units])

    def _design(self):
        B_eff = self.boiler_efficiency
        TG_eff = self.turbogenerator_efficiency
        steam_demand = self.steam_demand
        Design = self.design_results
```

```
chemicals = self.chemicals
self._load_utility_agents()
mol_steam = sum([i.flow for i in self.steam_utilities])
feed_solids, feed_gas, makeup_water, feed_CH4, lime, chems = self.ins
feed_CH4.phase = 'g'
feed_CH4.set_property('T', 60, 'degF')
feed_CH4.set_property('P', 14.73, 'psi')
emissions, blowdown_water, ash_disposal = self.outs
if not lime.price:
   lime.price = 0.19937504680689402
if not chems.price:
   chems.price = 4.995862254032183
H_steam =  sum([i.duty for i in self.steam_utilities])
side_steam = self.side_steam
if side_steam:
   H_steam += side_steam.H
   mol_steam += side_steam.F_mol
steam_demand.imol['7732-18-5'] = mol_steam
duty_over_mol = 39000 # kJ / mol-superheated steam
emissions_mol = emissions.mol
emissions.T = self.T_emissions
emissions.P = 101325
emissions.phase = 'g'
self.combustion_reactions = combustion_rxns = chemicals.get_combustion_reactions()
non_empty_feeds = [i for i in (feed_solids, feed_gas) if not i.isempty()]
boiler_efficiency_basis = self.boiler_efficiency_basis
def calculate_excess_electricity_at_natual_gas_flow(natural_gas_flow):
   if natural_gas_flow:
      natural_gas_flow = abs(natural_gas_flow)
      feed_CH4.imol['CH4'] = natural_gas_flow
   else:
      feed_CH4.empty()
   emissions_mol[:] = feed_CH4.mol
   for feed in non_empty_feeds: emissions_mol[:] += feed.mol
   combustion_rxns.force_reaction(emissions_mol)
   emissions.imol['O2'] = 0
```

```
        if boiler_efficiency_basis == 'LHV':
            H_combustion = feed_CH4.LHV
            for feed in non_empty_feeds: H_combustion += feed.LHV
        elif boiler_efficiency_basis == 'HHV':
            H_combustion = feed_CH4.HHV
            for feed in non_empty_feeds: H_combustion += feed.HHV
        else:
            raise ValueError(
                f"invalid boiler efficiency basis {boiler_efficiency_basis}; "
                f"valid values include 'LHV', or 'HHV'"
            )
        H_content = B_eff * H_combustion
        #: [float] Total steam produced by the boiler (kmol/hr)
        self.total_steam = H_content / duty_over_mol
        Design['Flow rate'] = flow_rate = self.total_steam * 18.01528

        # Heat available for the turbogenerator
        H_electricity = H_content - H_steam

        electricity = H_electricity * TG_eff  # Electricity produced
        self.cooling_duty = electricity - H_electricity
        Design['Work'] = work = electricity/3600
        if self.satisfy_system_electricity_demand:
            boiler = self.cost_items['Boiler']
            rate_boiler = boiler.kW * flow_rate / boiler.S
            return work - self.electricity_demand - rate_boiler
        else:
            return work

    self._excess_electricity_without_natural_gas = excess_electricity =
calculate_excess_electricity_at_natual_gas_flow(0)
    if excess_electricity < 0:
        f = calculate_excess_electricity_at_natual_gas_flow
        lb = 0.
        ub = - excess_electricity * 3600 / feed_CH4.chemicals.CH4.LHV
        while f(ub) < 0.:
```

```
        lb = ub
        ub *= 2
    flx.IQ_interpolation(f, lb, ub, xtol=1, ytol=1)
    if self.cooling_duty > 0.:
        # In the event that no electricity is produced and the solver
        # solution for natural gas is slightly below the requirement for steam
        # (this would lead to a positive duty).
        self.cooling_duty = 0.
        Design['Work'] = 0.

hu_cooling = bst.HeatUtility()
hu_cooling(self.cooling_duty, steam_demand.T)
hus_heating = bst.HeatUtility.sum_by_agent(tuple(self.steam_utilities))
for hu in hus_heating: hu.reverse()
self.heat_utilities = [*hus_heating, hu_cooling]
water_index = chemicals.index('7732-18-5')
makeup_water.mol[water_index] = blowdown_water.mol[water_index] = (
        self.total_steam * self.boiler_blowdown * 1 / (1 - self.RO_rejection)
)
ash_IDs = [i.ID for i in self.chemicals if not i.formula]
emissions_mol = emissions.mol
if 'SO2' in chemicals:
    ash_IDs.append('CaSO4')
    lime_index = emissions.chemicals.index(self._ID_lime)
    sulfur_index = emissions.chemicals.index('CaSO4')
    self.desulfurization_reaction.force_reaction(emissions)
    # FGD lime scaled based on SO2 generated,
    # 20% stoichiometric excess based on P52 of ref [1]

    lime.mol[lime_index] = lime_mol = max(0, emissions_mol[sulfur_index] * 1.2)
    emissions_mol.remove_negatives()
else:
    lime.empty()
# About 0.4536 kg/hr of boiler chemicals are needed per 234484 kg/hr steam produced
chems.imol['ash'] = boiler_chems = 1.9345e-06 * Design['Flow rate']
ash_disposal.empty()
```

```
        ash_disposal.copy_flow(emissions, IDs=tuple(ash_IDs), remove=True)
        ash_disposal.imol['ash'] += boiler_chems
        dry_ash = ash_disposal.F_mass
        ash_disposal.imass['water'] = moisture = dry_ash * 0.3 # ~20% moisture
        Design['Ash disposal'] = dry_ash + moisture
        if 'SO2' in chemicals:
            if self._ID_lime == '1305-62-0': # Ca(OH)2
                lime.imol['water'] = 4 * lime_mol # Its a slurry
            else: # CaO
                lime.imol['water'] = 5 * lime_mol

    def _cost(self):
        self._decorated_cost()
        self.power_utility.production = self.design_results['Work']

Boiler = BoilerTurbogenerator('Boiler', ('','','','natural_gas','',''), boiler_efficiency=0.8, natural_gas_price=0.1734,
satisfy_system_electricity_demand=False)



system_sys = bst.main_flowsheet.create_system('AD_Plant')
# Puts the units together to construct final diagram
system_sys.simulate()
system_sys.diagram()

#TEA
class AD_TEA(bst.TEA):

    def __init__(self, system, IRR, duration, depreciation, income_tax,
                 operating_days, lang_factor, construction_schedule, WC_over_FCI,
                 labor_cost, fringe_benefits, property_tax,
                 property_insurance, supplies, maintenance, administration):
        # Huang et. al. does not take into account financing or startup
        # so these parameters are 0 by default
        super().__init__(system, IRR,duration, depreciation, income_tax,
                         operating_days, lang_factor, construction_schedule,
```

```
                    startup_months=0, startup_FOCfrac=0, startup_VOCfrac=0,
                    startup_salesfrac=0, finance_interest=0, finance_years=0,
                    finance_fraction=0, WC_over_FCI=WC_over_FCI)
        self.labor_cost = labor_cost
        self.fringe_benefits = fringe_benefits
        self.property_tax = property_tax
        self.property_insurance = property_insurance
        self.supplies= supplies
        self.maintenance = maintenance
        self.administration = administration

    # The abstract _DPI method should take installed equipment cost
    # and return the direct permanent investment. Huang et. al. assume
    # these values are equal
    def _DPI(self, installed_equipment_cost):
        return installed_equipment_cost

    # The abstract _TDC method should take direct permanent investment
    # and return the total depreciable capital. Huang et. al. assume
    # these values are equal
    def _TDC(self, DPI):
        return DPI

    # The abstract _FCI method should take total depreciable capital
    # and return the fixed capital investment. Again, Huang et. al.
    # assume these values are equal.
    def _FCI(self, TDC):
        return TDC

    # The abstract _FOC method should take fixed capital investment
    # and return the fixed operating cost.
    def _FOC(self, FCI):
        return (FCI*(self.property_tax + self.property_insurance
                     + self.maintenance + self.administration)
                + self.labor_cost*(1+self.fringe_benefits+self.supplies))
```

```
tea = AD_TEA(system=system_sys,
            IRR=0.1, # Internal rate of return
            duration=(2023, 2043), # Start and end year
            depreciation='MACRS7', # Number of years
            income_tax=0.0307, # PA income tax
            operating_days=365*cap_factor, # Operating days per year
            lang_factor=5, # Lang factor: sum of capital costs beside purchased equipment
            construction_schedule=(0.4, 0.6),
            WC_over_FCI=0.05, #5% of fixed capital cost
            labor_cost=80000,#Assume one professional is hired
            fringe_benefits=0.4, #Cost of fringe benefits as a fraction of labor cost.
            property_tax=0.001, #Fee as a fraction of fixed capital investment
            property_insurance=0.005, #Fee as a fraction of fixed capital investment
            supplies=0.20, #Yearly fee as a fraction of labor cost.
            maintenance=0.01, #Yearly fee as a fraction of fixed capital investment.
            administration=0.005) #Yearly fee as a fraction of fixed capital investment.


tea.show() # Print TEA summary at current options
tea.get_cashflow_table()
```

BIBLIOGRAPHY

Abdallah, M., Shanableh, A., Adghim, M., Ghenai, C., & Saad, S. (2018). Biogas production from different types of cow manure. *2018 Advances in Science and Engineering Technology International Conferences, ASET 2018*, 1–4. https://doi.org/10.1109/ICASET.2018.8376791

Agostini, F., Gregory, A. S., & Richter, G. M. (2015). Carbon Sequestration by Perennial Energy Crops: Is the Jury Still Out? *BioEnergy Research 2015 8:3*, *8*(3), 1057–1080. https://doi.org/10.1007/S12155-014-9571-0

Anaerobic Digestion on Dairy Farms | US EPA. (n.d.). Retrieved March 28, 2024, from https://www.epa.gov/agstar/anaerobic-digestion-dairy-farms

Chen, Y., Cheng, J. J., & Creamer, K. S. (2008). Inhibition of anaerobic digestion process: A review. *Bioresource Technology*, *99*(10), 4044–4064. https://doi.org/10.1016/J.BIORTECH.2007.01.057

Cortes-Peña. (n.d.). *GitHub - BioSTEAMDevelopmentGroup/Bioindustrial-Park: BioSTEAM's Premier Repository for Biorefinery Models and Results*. Retrieved March 27, 2024, from https://github.com/BioSTEAMDevelopmentGroup/Bioindustrial-Park?tab=readme-ov-file

Cortes-Peña, Y., Kumar, D., Singh, V., & Guest, J. S. (2020). BioSTEAM: A Fast and Flexible Platform for the Design, Simulation, and Techno-Economic Analysis of Biorefineries under Uncertainty. *ACS Sustainable Chemistry and Engineering*, *8*(8), 3302–3310. https://doi.org/10.1021/ACSSUSCHEMENG.9B07040/ASSET/IMAGES/LARGE/SC9B07040_0007.JPEG

Dairy Farm Waste Manure Dewatering Machine/screw Press Cow Dung Slurry Separator/cow Dung Cleaner Animal Manure Dryer. (2024). https://www.alibaba.com/product-detail/dairy-farm-waste-manure-dewatering-machine_1600466113079.html

Deng, L., & Hägg, M. B. (2010). Techno-economic evaluation of biogas upgrading process using CO2 facilitated transport membrane. *International Journal of Greenhouse Gas Control*, *4*(4), 638–646. https://doi.org/10.1016/J.IJGGC.2009.12.013

Foxen, R., & Fitzgerald, J. (2019). Environmental and Agricultural Gains From Renewable Natural Gas. *Natural Gas & Electricity*, *35*(11), 12–18. https://doi.org/10.1002/GAS.22123

Holly, M. A., Gunn, K. M., Rotz, C. A., & Kleinman, P. J. A. (2019a). Management characteristics of Pennsylvania dairy farms. *Applied Animal Science*, *35*(3), 325–338. https://doi.org/10.15232/AAS.2018-01833

Holly, M. A., Gunn, K. M., Rotz, C. A., & Kleinman, P. J. A. (2019b). Management characteristics of Pennsylvania dairy farms. *Applied Animal Science*, *35*(3), 325–338. https://doi.org/10.15232/AAS.2018-01833

Jin, G., Bierma, T., & Walker, P. (2012). Biogas production from switchgrass under experimental conditions simulating U.S. digester operations. *Journal of Environmental Science and Health, Part A*, *47*(3), 470–478. https://doi.org/10.1080/10934529.2012.646150

Lam, P. S., Sokhansanj, S., Bi, X., Lim, C. J., Naimi, L. J., Hoque, M., Mani, S., Womac, A. R., Narayan, S., & Ye, X. P. (2008). Bulk Density of Wet and Dry Wheat Straw and Switchgrass Particles. *Applied Engineering in Agriculture*, *24*(3), 351–358. https://doi.org/10.13031/2013.24490

Mcvoitte, W. P. A., & Clark, O. G. (2017). The effects of temperature and duration of thermal pretreatment on the solid-state anaerobic digestion of dairy cow manure. *Heliyon*, e02140. https://doi.org/10.1016/j.heliyon.2019.e02140

Niles, M. T., & Wiltshire, S. (2019). Tradeoffs in US dairy manure greenhouse gas emissions, productivity, climate, and manure management strategies. *Environmental Research Communications*, *1*(7), 075003. https://doi.org/10.1088/2515-7620/AB2DEC

Nleya, Y., Young, B., Nooraee, E., & Baroutian, S. (2023). Opportunities and Challenges for Anaerobic Digestion of Farm Dairy Effluent. *ChemBioEng Reviews*, *10*(6), 924–940. https://doi.org/10.1002/CBEN.202300021

Parker, N., Williams, R., Dominguez-Faus, R., & Scheitrum, D. (2017). Renewable natural gas in California: An assessment of the technical and economic potential. *Energy Policy, 111*, 235–245. https://doi.org/10.1016/J.ENPOL.2017.09.034

Seider, W., Lewin, D., Seader, J., Gani, R., & Ng, K. (2017). *Product and process design principles: synthesis, analysis, and evaluation* (4th ed.). Wiley.

Tan, W. E., Liew, P. Y., Tan, L. S., Woon, K. S., Mohammad Rozali, N. E., Ho, W. S., & NorRuwaida, J. (2022). Life Cycle Assessment and Techno-Economic Analysis for Anaerobic Digestion as Cow Manure Management System. *Energies*, *15*(24), 9586. https://doi.org/10.3390/EN15249586/S1

US Electricity Profile 2022 - U.S. Energy Information Administration (EIA). (n.d.). Retrieved March 25, 2024, from https://www.eia.gov/electricity/state/

USDA, EPA, & DOE. (2015). *BIOGAS OPPORTUNITIES ROADMAP PROGRESS REPORT*.

USDA ERS - Dairy Data. (n.d.). Retrieved March 5, 2024, from https://www.ers.usda.gov/data-products/dairy-data.aspx

von Keyserlingk, M. A. G., Martin, N. P., Kebreab, E., Knowlton, K. F., Grant, R. J., Stephenson, M., Sniffen, C. J., Harner, J. P., Wright, A. D., & Smith, S. I. (2013). Invited review: Sustainability of the US dairy industry. *Journal of Dairy Science*, *96*(9), 5405–5425. https://doi.org/10.3168/JDS.2012-6354

Wang, H., Aguirre-Villegas, H. A., Larson, R. A., & Alkan-Ozkaynak, A. (2019). Physical Properties of Dairy Manure Pre- and Post-Anaerobic Digestion. *Applied Sciences 2019, Vol. 9, Page 2703*, *9*(13), 2703. https://doi.org/10.3390/APP9132703

Wendt, C., Ellis, C., Guillen, D. P., Feris, K., Coats, E. R., & McDonald, A. (2016). Reduction of GHG emissions through the conversion of dairy waste to value-added materials and products. *Energy Technology 2016: Carbon Dioxide Management and Other Technologies*, 109–116. https://doi.org/10.1007/978-3-319-48182-1_13/COVER

Wilkie, A. C. (2005). Anaerobic Digestion of Dairy Manure: Design and Process Considerations. *Natural Resource, Agriculture, and Engineering Service*, 301–312.