REVERSE OSMOSIS DESALINATION WITH INTEGRATED COMPRESSED AIR ENERGY STORAGE: CONCEPTUAL DESIGN AND MODELING

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Environmental Engineering

---

by
Pooja Kishor Mahajan
December 2012

Accepted by:
Dr. David A. Ladner, Committee Chair
Dr. Tanju Karanfil
Dr. Thomas Overcamp

ABSTRACT

Growing fresh water needs have led to an interest in water desalination using reverse osmosis (RO) membranes. Energy consumption has a large share in the expenses incurred to drive a reverse osmosis desalination system. With depleting fossil fuel reserves, tapping renewable sources of energy is a promising alternative to meet the energy requirements. Wind power coupled to an RO desalination system is a potential means of delivering clean water using sustainable energy. This work investigates modeling of a conceptual wind-energy-driven RO desalination system that is made possible because of an air-pressure energy storage mechanism. Bench-scale experiments were performed on an RO membrane connected to a pressure vessel to validate the models, which were developed based on film theory and solution-diffusion concepts. The models predict water flux and salt concentration through an RO membrane, and were further expanded to predict the performance of a few conceptual full-scale system designs consisting of conventional and air-pressure energy storage wind-RO systems. In the latter design, the energy storage tank serves as a buffer to dampen the variability caused in the discharges and pressures due to the stochastic nature of the wind. The performance of the two systems were compared by varying several input parameters such as the wind patterns, tank volumes, number of RO elements, initial air pressures inside the tank, and the lower pressure limit. The air-pressure energy storage wind-RO system was found to deliver higher water production and better water quality than the conventional wind-RO system demonstrating the usefulness of an energy storage tank for a wind-driven

desalination system. Parameters that were important in the design were the initial air pressure and the lower pressure limit (the lowest pressure at which the air tank was allowed to operate). When both the initial air pressure and the lower pressure limit were high, the greatest water productivity was achieved and salt rejection was kept high (98.5 %). These and other parameters are explained in this thesis, giving a framework for thinking about how an air-pressure energy storage system can be integrated with RO to provide renewable-energy desalination.

DEDICATION

I would like to dedicate this thesis to my parents, Mr. Kishor Mahajan and Mrs. Pratibha Mahajan, my sister Ms. Rucha Mahajan and a friend Mr. Sameer Samant who stood by me and helped me complete it successfully.

ACKNOWLEDGMENTS

I would like to thank my research advisor, Dr. David Ladner, for the guidance he has lent to my work. I would also like to thank Dr. Tanju Karanfil and Dr. Thomas Overcamp for serving as my research committee.

I would like to thank Mr. Mark Sheikhrezai, CEO of Windation Energy Systems, who first gave my advisor the challenge of desalinating seawater using wind energy. The College of Engineering and Science at Clemson University is also acknowledged for funding in the form of Dr. Ladner's startup package. I would also like to thank my co-worker, Ying Sun for performing the experiments.

Lastly, I would like to thank my family and a special friend Mr. Sameer Samant for providing me with all the support and understanding.

TABLE OF CONTENTS

Table of Contents (Continued)

LIST OF TABLES

LIST OF FIGURES

List of Figures (Continued)

List of Figures (Continued)

Page

List of Figures (Continued)

List of Figures (Continued)

Page

NOMENCLATURE

RO	Reverse osmosis
NF	Nanofiltration
ED	Electro dialysis
CA	Cellulose acetate
COE	Cost of energy
PV	Pressure vessel
GHG	Green house gas
ERD	Energy Recovery Device
WECs	Wind energy converter system
lmh	liters/m^2/h
C_b	Bulk feed concentration
C_p	Permeate concentration
σ_b	Bulk feed conductivity
σ_p	Permeate conductivity
π_b	Bulk osmotic pressure
J_v	Permeate flux
A	Water permeability coefficient
Δp	Applied Pressure
$\Delta\pi_m$	Osmotic Pressure
K_s	Salt permeability

C_w	Membrane wall concentration
C_p	Permeate salt concentration
J_s	Solute flux
C_b	Bulk feed concentration
k	Mass transfer coefficient
D	Mass transfer coefficient of NaCl
γ	Wall shear rate across membrane
u	Crossflow velocity
Q_f	Feed flow rate across RO membrane
w	Width of flow channel
L	Length of channel
h	Height of membrane channel
A_{eff}	Effective cross sectional area of membrane
ε_{spacer}	Porosity of the feed spacer
Q_c	Cross flow rate
f_{cp}	Concentration polarization factor
SR	Salt Rejection
Q_p	Permeate flow rate
W_{pump}	Pump Power
W_T	Power from wind turbine
ρ_{air}	Density of air
A_{rotor}	Area swept by the blades of a wind turbine

V_{wind}	Wind speed
Q	Discharge rate
N	Pump speed
m	Mass of permeate water
t	Time
ρ	Permeate water density
a_{memb}	Surface area of the membrane coupon

1. BACKGROUND AND LITERATURE REVIEW

1.1 Introduction

With rapid depletion in fresh water reserves, seawater desalination has become a viable option to meet the increasing demand for drinking water across the world. Fresh water resources have been exploited causing water shortage and a need to find an alternative water source. According to the World health Organization (WHO), at least 1.1 billion people in the world today have no access to clean drinking water, and this number is expected to rise to 3 billion by the year 2025 (Greenlee et al., 2009; Forstmeier et al., 2007).

With a water demand estimated to double in every 20 years (Greenlee et al., 2009), production of potable water from seawater is considered as a potential water source to curb the water shortage problems in the world. Typical salt concentrations in seawater vary from 35,000-45,000 mg/L and the permissible limit for potable water is 500 mg/L as recommended by the US Environmental Protection Agency (Greenlee et al., 2009). Thus, desalination is a process of removing or reducing the salt content in the water to produce fresh water. Seawater desalination has been used for over 100 years in the Middle East (Greenlee et al., 2009) and several other countries, including the United States, have adopted this technique in an attempt to meet their growing water demand. It can be achieved by two methods – 1. Thermal processes which involve phase change of the feed solution and, 2. Membrane processes which do not involve a change in the phase. In the 1950s, seawater desalination was accomplished using thermal processes (Greenlee et al., 2009), however, they are now replaced with membrane systems due to the increasing energy costs and fuel consumption incurred by the former process. Membrane desalination is driven by an electric pump and therefore consumes relatively less energy (Sourirajan and Agrawal, 1969; US EPA, 1996; Greenlee et al., 2009; Li and Wang 2010). Most commonly used membranes for desalination include reverse osmosis (RO), nanofiltration (NF) and electrodialysis (ED) of which RO membranes are used most widely in the United States accounting for 69 % of the plants (Greenlee et al., 2009).

Reverse osmosis is a process used for separation of a solute from a solvent in aqueous or gaseous phase. In general, it involves allowing the mixture to flow through a porous membrane from one side and collecting the separated product from the other side (Agrawal and Sourirajan, 1969). It is used for desalination purposes as RO membranes are capable of excluding mono-valent ions from aqueous solutions, while allowing water to pass through it (Sourirajan and Agrawal, 1969; US EPA, 1996; Garcia-Rodriguez, 2003; Greenlee et al., 2009; Li and Wang 2010). For separation to occur, an external pressure is applied to overcome the membrane osmotic pressure. Thus, the fluid mixture is forced through the membrane at high pressure and the product water is collected at atmospheric pressure (Agrawal and Sourirajan, 1969). The process of separation occurs by diffusion through the membrane. There are two models proposed for the transport mechanism through a membrane- pore flow model and solution-diffusion model. The most widely accepted explanation is the solution-diffusion theory according to which water transport occurs as a result of absorption and diffusion across the membrane (Paul, 2004; Greenlee et al., 2009).

An efficient RO membrane delivers a high water flux and a high salt rejection, up to 99.8% when operated under standard conditions. However, the feed water concentrations and the operating conditions greatly affect the performance of an RO membrane. RO membranes are made from cellulose acetate (CA) or aromatic polyamide (PA) of which the latter polymer exhibits superior performance as compared to the former (Sourirajan and Agrawal, 1969; Kiranoudis et al., 1997). RO is used in 50 % of the desalination plants across the world and is finding ample interest in the water treatment industry (Li and Wang 2010). Besides its high salt rejection and relatively low energy costs, its compactness provides an advantage of leaving a small spatial footprint. An RO system consists of multiple closed vessels that house the RO membrane which is spirally wound around a central tube. Pressurized feed water is fed through one end and the concentrate or the brine is collected at the other end. The central tube collects the permeate water (NREL, 2006)

Despite offering several benefits, a substantial amount of energy is consumed in membrane processes due to the low recovery ratio and high pressures required for overcoming the trans-membrane osmotic pressures. The energy consumption and membrane replacement cost account for almost 45 - 50% of the total water production cost (El-Ghonemy, 2012; Avlonitis et al., 2003). For a RO system, the energy required to desalinate 1000 L of water ranges between to be 3 – 8 kWh depending on the feed concentration, membrane properties and operating conditions (Avlonitis et al., 2003;

Forstmeier et al., 2007; Charcosset, 2009). Most of the desalination plants today, use energy derived from fossil fuels either directly or in the form of electricity (Forstmeier et al., 2007). With rising fuel costs and a necessity to protect our environment from CO_2 emissions, there is a need to harvest renewable energy sources to drive desalination systems (Tzen and Morris, 2003). Another advantage of using renewable energy desalination system is that the cost of energy (COE) associated with transmission and distribution of conventional energy is saved (NREL, 2006). Thus, an RO desalination system powered by renewable energy is an attractive alternative to solve the drinking water issues and also help with the energy crisis we are facing today.

1.2 Renewable Energy for Desalination

Renewable energy resources are inexhaustible and clean, and these advantages can be used for an energy demanding RO process (Charcosset, 2009). Several studies have been conducted to investigate the reliability and feasibility of employing alternate renewable energy sources (Liu et al., 2002; Miranda and Infield, 2002; Garcia-Rodriguez, 2003; Kalogirou, 2005; Gilau and Small 2008). Using natural energy sources such as wind, tidal, geothermal and solar to supplement and/or substitute the conventional sources for driving electrically operated RO systems has been widely discussed (Liu et al., 2002; Miranda and Infield, 2002; Garcia-Rodriguez, 2003). Solar energy and wind energy have been found to be the most promising sources to drive RO desalination plants (Sourirajan and Agrawal, 1969; Madireddi et al., 1999; Kershman et al., 2002). There are two technologies involved in a renewable energy driven desalination system: first, conversion of the energy and second, desalination of the water. It is important to integrate the systems such that an optimum design can be obtained (Mathioulakis et al., 2007).

The availability of renewable energy is discontinuous and has a fairly low intensity. An RO process, on the other hand, is a continuous and energy intensive process. Most often, a renewable energy desalination plant is designed to operate by converting to the energy to electricity to compensate for the discontinuous nature of renewable energy. In such a system, the RO plant operates as a conventional system and the renewable energy is simply a replacement for the traditional fossil-fuel source (Eltawil et al., 2009)

Solar photovoltaic (PV) cells connected to an RO system are successfully used for both seawater and brackish water desalination on a commercial basis in countries like Spain, Italy and Saudi Arabia (Madireddi et al., 1999; Miranda and Infield, 2002; Liu, 2002; Garcia-Rodriguez, 2003; Ma et al., 2009). Although solar energy desalination systems are commercially available, they exhibit a high capital cost as they require large PV arrays and specialized electronic invertors and charge regulators (Thomson and Infield, 2002; Garcia-Rodriguez, 2003; Forstmeier et al., 2007).

As compared to solar PV, wind power incurs lower energy costs and has the potential of replacing the conventional sources. (Garcia-Rodriguez, 2003). Wind energy proves beneficial especially for plants located near coastal areas or areas with wind speeds suitable to power desalination systems. In a study conducted by Kesherman et al. (2002), they compared the performances of four configurations of a desalination system based on its power supply source namely, grid supply, combined wind and grid supply, combined PV and grid supply and combined wind, PV and grid supply. Their results showed that the performance of the system driven purely by electricity was poorest in terms of energy availability, fuel consumption and greenhouse gas (GHG) emissions. It was found that for the Grid + PV system the energy supply required from the grid was 89% whereas, the Grid + wind required 57% of the energy from the grid. In terms of environmental benefits, the emissions of GHGs were lower in a grid connected to a wind turbine than PV. Also, the cost analysis suggested that the additional costs incurred due to integration of renewable energy sources were higher for PV than wind energy conversion. Another study by Kiranoudos et al. (1997) showed that the cost of production of freshwater can be reduced by 20% by a wind-RO system receiving average wind speeds greater than 5 m/s. Charcosset (2009) suggested that, with the advancements in the technology, there is a decrease in the RO plant costs and wind turbine costs making it more economical. Further, Mathioulakis et al., (2007) recommended in their study that a seawater wind-RO system is most suitable for small-medium scale plants.

A stand-alone wind-RO system is more economical in areas where electricity grids are not available (Tzen and Morris, 2003). Furthermore, if methods are devised that can supply renewable energy directly to drive the desalination system, using a grid connection becomes redundant. However, the intermittent and unpredictable nature of wind poses a big challenge in utilizing it as a reliable source of energy. Most commonly, wind energy has been used as an auxiliary source to the conventional sources (Swift et al., 2009) In a conventional wind energy converter system (WECs), the mechanical energy is first converted to electrical energy which is converted back to mechanical energy to drive the pumps connected to the RO modules. These conversions increase the project cost and losses in the system (Witte et al., 2003). However, attempts have been made to design a stand-alone wind energy desalination system to avoid the use of grid system. Witte et al. (2003) showed that a wind turbine can be directly connected to a RO system by using a pressure accumulator and a control valve that can regulate the pressure and flow through the RO membrane. Miranda and Infield (2003) developed a variable-flow RO desalination plant powered by a 2.2 kW wind turbine. Their modeled system consisted of the wind turbine supplying feed water directly to the RO system with the use of a variable-speed, positive displacement pump. A Clark pump was used to recover energy from the brine stream. Their modeling results indicated that such a system is feasible and projected a permeate flow rate of 8.5 m^3/d and an average product water concentration of 300 mg/L. Further, the model also estimated a specific energy of 3.4 kWh/m^3 as compared to the specific energy of desalination systems without energy recovery devices of 10 kWh/m^3 (Thomson et al., 2002).

In a stand-alone system, the challenge lies in coupling the renewable energy system with the desalination system such that there is an uninterrupted energy supply available to drive the RO system from an intermittent energy source. Most commonly the two processes are connected together by using an energy storage mechanism such as batteries, diesel engines and flow/pressure stabilizer (Kesherman et al., 2002; Miranda and Infield, 2003; Mathioulakis et al., 2007; Gilau and Small, 2008). Batteries can cause problems in hot climate and result in an energy loss up to 25 % (Thomson et al., 2002). The flow/pressure stabilizer acts as a buffer to dampen the fluctuations in the energy supply to the pump that increases the pressure of the feed water for desalination (Garcia-Rodriguez, 2003).

Liu et al., 2002 developed a prototype of a stand-alone wind-RO system with an energy storage tank on the Pacific Islands. The system used wind energy to pump brackish water through the ultra-low pressure RO membrane units. It also consisted of a pre-treatment unit that was incorporated into the RO system. The system made use of a pressure stabilizer to dampen the fluctuations in the pressure and flows caused by the intermittency of wind. Since the feed water was brackish in nature, with a salinity of 2,500 mg/L the working pressures of this system were in the range of 70-105 psi (483-724 kPa). On the contrary, seawater desalination requires pressures of the order of 800-1000 psi owing to the higher salt concentrations of the order of 32,000 – 35,000 mg/L and designing a system that uses wind energy to generate such high pressures has not been found in the literature. This study is intended towards modeling a wind-RO desalination system with integrated energy storage tank for sweater desalination.

The design of a wind driven RO system requires consideration of optimum membrane configuration, wind turbine size and the operational characteristics. A typical wind powered desalination plant consists of a network of membranes connected to a high pressure reciprocating pump which is in turn connected to a wind turbine (Kiranoudos et al., 1997). Most commonly such systems require a feedback control system for their optimal operation. As RO membranes work at pressures of the order of 7,000 kPa, substantial pump power is required to bring the feed water to this pressure.

1.3 Research Approach

Bench-scale, proof-of concept experiments were designed and performed by a co-worker, Ying Sun. The system was subjected to variable regime consisting of successive stops and starts to simulate the operation of a wind turbine connected to the RO pump directly. For details about the experimental setup and procedures refer to Sun (2013).

The focus of this thesis is to develop mathematical models to predict the performance of RO systems with integrated energy storage tank. The modeling is conducted in two parts. The first part consists of modeling the bench-scale setup with different operating characteristics and parameters. Film theory is used to model the bench-scale setup and obtain the mass transport parameters. The second part of the modeling consists of designing a full-scale system using these transport parameters.

The present work describes the use of a pressure vessel (or energy storage) tank as an energy storage device to operate the RO system. The design of an energy storage device is crucial in determining the system performance and the total project costs (Miranda and Infield, 2003). Further, to compare the performances of different system configuration and the important parameters, the model will predict concentration polarization, water quality, permeate water flux and salt rejection. Of these, the most critical parameter to a RO membrane is the concentration polarization. Concentration polarization is the reversible buildup of a salt layer on the membrane surface. It is a phenomenon in which the solute concentration on the membrane surface is higher than the bulk concentration (Zydney, 1997; Wiley and Fletcher, 2002; Zhou et al., 2006). Thus, the solute concentration profile at the membrane surface varies from the bulk solute concentration (Zhou et al., 2006). Concentration polarization has detrimental effects on the membrane performance. It causes solute passage and trans-membrane osmotic pressure resulting in lower salt rejection and permeate flux rates (Elimelech and Bhattacharjee, 1998; Wiley and Fletcher, 2002). It also degenerates the membrane quality reducing its useful life. One way to reduce concentration polarization is to place spacers on the feed side of the membrane and run the system with tangential or crossflow.

The magnitude of concentration polarization is governed by solute and membrane properties and flow hydrodynamics (Kim and Hoek, 2005; Ma et al., 2009; Ladner et al., 2010). An accurate prediction of concentration polarization and methods to reduce it are critical in optimizing the RO process. Several theories have been used to model concentration polarization in order to understand and predict the performance of the system. The models used for simulating concentration polarization are either analytical models such as film theory and finite element models, or numerical models such as computational fluid dynamics (Song and Eilmelech, 1995; Murthy and Gupta, 1997; Kangwondo, 1999; Hoek and Elimelech 2003; Zhou et al., 2006; Ladner et al., 2010). In the present work, the parameters listed above are determined by using the analytical film theory model in order to predict steady-state RO performance.

Water quality is the most important aspect of any water treatment systems. In a RO desalination system, water quality depends on the ability of the membranes to reject the salts from passing through it. In order to obtain water with minimum salt, the salt rejection of the membrane must be very high. Typically, an RO membrane has salt rejections greater than 99 % (Greenlee et al., 2009). There are two theories proposed to explain the passage of salt through an RO membrane: pore flow and the diffusion theory (Sourirajan and Agrawal, 1969). The most commonly accepted theory is the solution-diffusion theory (Wijmans and Baker, 1995).

After water quality, the next important parameter that governs the feasibility and economics of any treatment system is the amount of treated water produced per unit untreated water and per unit energy consumed. It is well known that RO systems require a high energy for it to operate at high pressures. Further, the recovery ratio which is defined as the ratio of the product water flow rate and the feed flow rate, is also low for a RO system due to the high osmotic pressure. To overcome this, energy recovery devices are used to make use of energy stored in the pressurized feed water.

A conventional RO system is modified to incorporate a long cylindrical vessel working on the principle of compressing gas inside it. The novel idea in this study is that it is proposed that the wind-RO system be operated in batches. In other words, desalination will occur in two stages. In the first stage, feed water is pumped into the energy storage device and in the second stage, that feed water is discharged into the membrane elements for desalination to occur. The energy storage tank will be pre-charged with air and as feed water is pumped into the tank, it will be pressurized. The pressurized water is then discharged into the membrane units attached to the pressure vessel, so that the pressure inside the tank declines.

For the conceptual full-scale system, real wind data is used to model the system. Four wind patterns are used to compare different system configurations. Further, commercially available spiral wound RO membrane specifications are used for the design purposes. A base design, a conventional wind-RO system is modeled to compare with the energy storage tank setup. The conventional system was also modeled for a hypothetical steady wind pattern the performance data provided by the membrane manufacturers is often over-estimated and hence the conventional wind-RO with steady wind pattern served as the standard base design.

The energy storage design was first modeled for a single tank. This design, although not most efficient served as a basis for studying the performance of such a system. Several input parameters such as the tank volume, initial air pressure, number of membrane elements etc. were varied to forecast the water production and water quality and to understand the working of the energy storage tank. The system was then modeled to incorporate three energy storage tanks to make the system efficient and practical. The input parameters were varied to verify its operation and the design was optimized.

2. MATERIALS AND METHODS

2.1 Modeling Bench-Scale RO Desalination System

The bench-scale RO desalination system was modeled for two purposes: first, to validate that the transport models could accurately fit real data and, second, to extract membrane transport parameters that were used to model a conceptual full-scale system design. The bench-scale system was modeled using the film theory approach. Following is a brief description of the experimental setup and methods that form the basis of the bench-scale modeling.

2.1.1 Bench-Scale Experimental Procedure

The components of the bench-scale RO setup included a crossflow membrane test cell, a positive displacement pump, motor, pressure vessel tank, pressure gauges, cooling system, valves, balance, conductivity probes and data acquisition equipment. The tubing was made from stainless steel. Permeate conductivity was measured by allowing the permeate water to flow into a small tube with the conductivity probe inserted into it. The permeate water flowed out of the tube into a flask placed on a balance that measures the mass of the product water. An automatic needle valve was used to control the pressure in the membrane cell. The data acquisition unit consisted of a personal computer with a data acquisition card capable of analog input and output. The software used for programming and signal interpretation was LabVIEW, which features feed and permeate conductivities, permeate mass, applied pressure, salt rejection, tank volume, temperature and the automatic needle valve voltage. These data were collected by LabVIEW program called "RO control" at an interval of 10 milliseconds and the values were averaged over a period of 10 seconds and saved to the hard drive. Each set of experimental data was stored in a computer with a unique filename and in a single folder so that the data of any experiment can be extracted and analyzed using Matlab software.

For the present study, the setup was built in such a way that it could be operated in two modes: the conventional mode and air-pressure energy storage mode. The conventional setup for the system is as shown in Figure 1. The energy storage tank setup consisted of a pressure vessel connected before the membrane cell (Figure 2) as that worked as an energy storage device for a wind-driven RO desalination system.

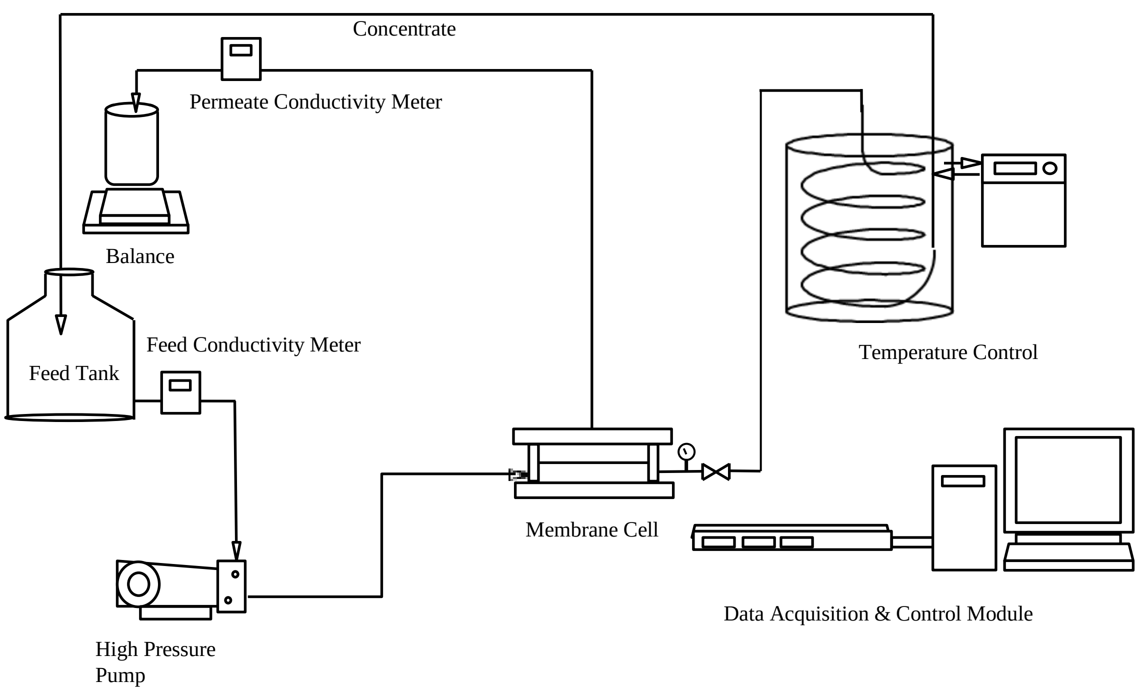


Figure 1. Convetnional bench-scale setup for the RO desalination unit.

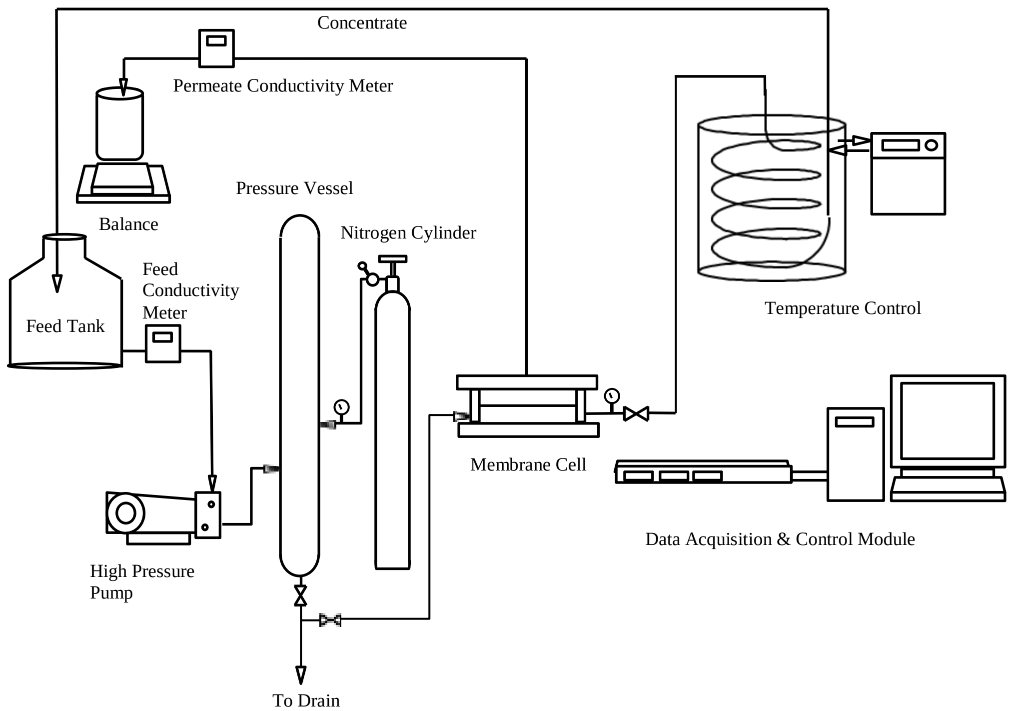


Figure 2. Modified bench-scale energy storage setup for RO desaliantion unit.

The main hypothesis of this work was that when an RO system is connected to a variable energy input, such as a wind turbine or a windmill, a technique, to store the energy and dampen the fluctuations in the flow rates will make such a system feasible to operate. An air-pressure energy storage method was proposed here and was built in the laboratory. The existing RO apparatus was modified to incorporate a pressure vessel that was intended to serve as a buffer to dampen the fluctuations caused in the pressure and flow rates due to varying speeds of the pump attached to a wind turbine (Figure 2). A nitrogen cylinder was used to pressurize the vessel with nitrogen gas. The pressure vessel tank was made from stainless steel and has four openings. Three openings were attached each to the pump, membrane and the nitrogen cylinder. The fourth opening served as an outlet to drain excess water from the vessel. Each opening had ball valves to control the flow of water. The volume of the pressure vessel tank measured in the lab was 10.7 L.

Salt Water

Salt water was prepared in the laboratory by mixing sodium chloride in deionized water. The concentrations used for testing were 0 g/L, 5 g/L, 15 g/L, 25 g/L and 35 g/L. A 14 L fresh salt water solution was prepared for each experiment.

Membranes

RO membranes were obtained from two manufacturers: SW30HR from Filmtec, a wholly owned subsidiary of the Dow Chemical Company (Midland, Michigan). Coupons of size 14.6 cm by 9.5 cm were cut and placed in DI water and 0.02% sodium azide, then stored at 4 °C at least overnight and up to several weeks with DI water replaced regularly.

Experimental Methods

In the conventional setup, salt water of known volume and concentration was pumped into the membrane test cell. The system was run until steady-state conditions were reached and allowed to maintain the steady state for sufficient time. The steady-state condition is achieved when the pressure applied to the system remains fairly constant over time resulting in a constant salt rejection and water flux. In this setup, a constant pressure of 1,000 psi is applied to the membrane during the steady-state.

While running the system in the pressure vessel setup, the pressure vessel tank was first pre-charged with nitrogen gas at a chosen pressure. Feed water of known concentration was then pumped continuously into it. As feed water filled the tank, the nitrogen gas was compressed and the pressure inside the pressure vessel increased. The high pressure inside the pressure vessel drove the feed water into the membrane cell unit for desalination to occur. Thereafter the system worked exactly like the conventional system.

In the laboratory, different protocols were developed for simulating a fluctuating wind energy supply to the RO system. One method to operate such a system was by conducting desalination in batches i.e., by allowing the pressure vessel to fill so that high pressure is developed inside the pressure vessel and then discharging the water into the membrane cell unit. Thus the pressure energy stored by pumping water into a pre-charged tank was used to drive the system. As the water flowed out of the pressure vessel, the pressure decreased gradually. In the laboratory, the system was operated at a constant pressure of 1,000 psi (6,900 kPa) to achieve a steady-state and then gradually declining the pressure by operating the needle valve to simulate batch operation method. The pressure was decreased at calculated intervals and until the pressure decreased to 700 psi. In this method, the pump was operated continuously and at a steady speed.

Before and after each experiment the system was rinsed by pumping DI water to minimize salt-induced corrosion in the pump, pipes, vessel and the membrane test unit. Clean water flux experiments were also conducted using DI water to calculate the water permeability coefficient of the membrane. The bench-scale experiments were performed by a co-worker Ying Sun and the experimental data were used for modeling purposes in this study. For a detailed explanation on the experimental setup and the experimental protocol please refer to Sun (2013).

The eight parameters measured and stored in the computer during the bench-scale experiments were: time, pump speed, pressure, feed and permeate conductivities, feed temperature, permeate mass, actuator voltage and feed tank volume. The pump speed governed the cross flow velocity across the membrane which in turn affected the concentration polarization across the membrane. The pressure applied externally worked against the osmotic pressure to drive water through the membrane. In the laboratory setting, the applied pressure was measured and recorded in psi. Feed conductivity and permeate conductivities were measured using conductivity probes immersed into the feed tank and the permeate water collection vessel during the experiments. The conductivities were used to determine the permeate salt concentrations. The mass of the permeate water was measured using a balance, and these data were used to calculate the permeate flux through the RO membrane. The temperature of the feed water has a effect on the permeate flux as it results in variation of the viscosity of the water (Goosen and Sablani et. al., 2002). The feed temperature was monitored and maintained constant throughout the experiment by using a cooling system.

The actuator voltage was used to control the operation of the automatic needle valve. This valve is used to regulate the pressure by controlling the crossflow. The automatic valve consists of an actuator attached to a needle valve. An actuator is an electronic device that takes input signal in voltage to controls the rotations of the needle valve. The operating voltage of the actuator was from 2 to 10 volts with 1 turn per volt.

The valve offers a resistance to the flow and thus helps in controlling the pressure inside the membrane unit.

2.1.2 Modeling the Bench-Scale System

The programming tool used for modeling the bench-scale RO system was Matlab. Program codes were developed to predict the performance of the system under different experimental setups. The bench-scale modeling program extracted the parameters from the bench-scale experiments to calculate the permeate flux and salt rejection. The program then predicted the permeate flux and salt rejection to validate the experimental results. The following flowchart describes the general steps involved in the modeling codes.

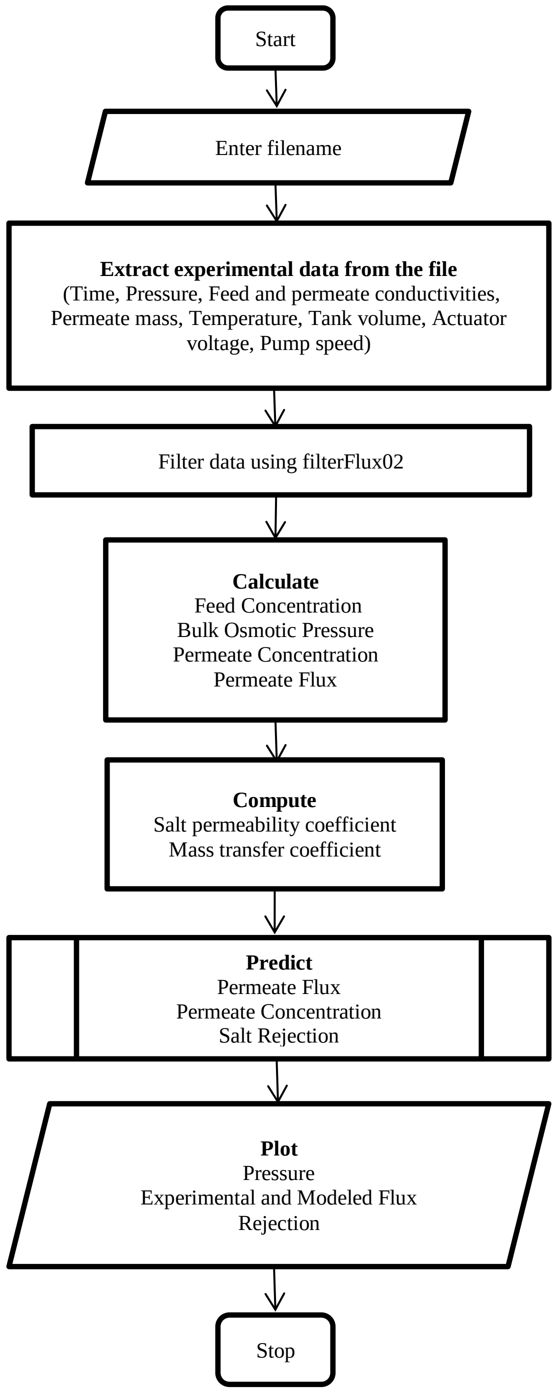


Figure 3. Flow chart depicting steps for modeling the bench-scale RO system.

Extracting and filtering experimental data

The program for bench-scale modeling followed the flow chart in Figure 3. It began by taking the filename from the user and extracting the data from that file. Each data file contained the eight parameters mentioned above. Each parameter was stored as a variable in the Matlab memory.

The experimental data set required adjustment before applying the modeling equations. Spurious data points were caused by changing the permeate container held on the balance when it was full, which affects balance readings and flux calculations. Other bumps or upsets also caused flux calculation error. To remove the false data a subroutine called filterFlux02 cuts off the data points where the change in mass over time fell outside of the expected range (0.5 to 4 g/min).

Computing indirectly measured parameters

As mentioned previously, some of the input parameters required for computing the water quality and quantity were measured indirectly in the laboratory. These included the concentrations of the feed and permeate measured in terms of the conductivities of the feed and permeate water and permeate water flux measured in terms of the permeate water mass. The bulk osmotic pressure, which was the osmotic pressure exerted by the presence of salt in the feed, was computed using the feed concentration.

The relationship between feed conductivity and the corresponding salt concentration is given by Ladner et al., 2010

$$C_b = 613.11 \,.\, \sigma_b - 664.62 \quad (1)$$

The relationship for permeate conductivity and permeate concentration is,

$$C_p = 512.1 \,.\, \sigma_p - 1.30 \qquad (2)$$

Where C_b is the bulk feed concentration in mg/L, σ_b is the bulk feed conductivity in mS/cm, C_p is the permeate concentration in mg/L and σ_p is the permeate conductivity in μS/cm. The bulk osmotic pressure (π_b) is required to predict the water flux through the membrane. It was computed in kilo Pascal using the bulk feed concentration as follows,

$$\pi_b = C_b.\ 8.505 \text{ x } 10^{-2} - \ 86.61 \qquad (3)$$

In a reverse osmosis desalination process the water quantity is measured in terms of the permeate water flux which is defined as the volume of water obtained per unit time per unit membrane area. The most commonly used unit for expressing the water flux is lmh or liters/m^2/hour. In the laboratory, the permeate water quantity was measured in terms of the mass of the permeate water collected per unit time (g/sec). In an RO process, the permeate water has a very low salt concentrations (> 99 % salt rejection) and hence the density of the permeate water was assumed to be equal to the density of pure water, i.e. 1000 g/L. The volume of the permeate water was calculated from the permeate water mass and its density. The permeate flux was obtained by dividing the permeate mass (g) over time (s) and the surface area of the membrane coupon (which is 0.01387 m^2).

$$J_v = \frac{m}{\rho\,.\,t\,.\ a_{memb}} \qquad (4)$$

Where J_v is the permeate flux (lmh), m is the mass of the permeate water (g), ρ is the water density (g/L), t is the time (h), a_{memb} is the surface area of the membrane coupon (m^2). The permeate water mass measured in the laboratory was a cumulative mass collected over the entire duration of the experiment. Hence, the value of permeate mass was subtracted from its former value to calculate the permeate water flux attained at each time point. This above equation of water flux is the actual flux obtained in the bench-scale experiments. This permeate flux was compared with the predicted flux values to validate the data.

Computing water permeability and salt permeability

Water and salt transport properties are intrinsic characteristics of a polymer RO membrane (Voros et al., 1996). To understand the operation of reverse osmosis process it is important to estimate these transport paramaeters (Taniguchi and Kimaru, 2004). The permeate flux depends on the water permeability coefficient of that membrane and its salt rejecting capacity is dependent on the salt permeability coefficient (Giese et al., 2010). Ideally, a reverse osmosis membrane should have a high water permeability coefficient and a zero salt permeability coefficient (Huisman, 1993).

For this study, the water permeability coefficient was determined by performing a clean water flux experiment in the laboratory. During a clean water run, deionized water was pumped into the membrane test cell at a pressure of 1,000 psi. The system was run for at least 2 h to achieve membrane compaction. The permeability coefficient was determined using the permeate flux equation as defined in the solution-diffusion theory. The flux equation in its simplest form can be expressed as (Kim and Hoek, 2005; Sassi and Mujtaba, 2010),

$$J_v = A\,.(\Delta p - \Delta\pi_m) \tag{5}$$

Where, J_v is the permeate water flux measured in lmh, Δp is the applied pressure in kPa and $\Delta\pi_m$ is the osmotic pressure in kPa. As there is no salt content in the water, there is no osmotic pressure exerted across the membrane. Thus, the flux equation can be written as,

$$J_v = A \ . \Delta p \tag{6}$$

or

$$A = \frac{J_v}{\Delta p} \tag{7}$$

A Matlab program was developed that extracted the data from the clean water experiment, filtered it and calculated pure water flux values as described by equation (4). Using the pressure and calculated flux values, the water permeability coefficient was determined using equation (7). The average value of the water permeability coefficient from the experiments was found to be 0.0634 lmh/psi or 0.0092 lmh/kPa. This value was used to model the conceptual full-scale system designs. The values found in the literature for SW30HR membrane are in the range of 0.0075-0.008 lmh/kPa (Ladner et al., 2010).

To determine the salt permeability of the membrane, the equation of solute flux through the membrane was used. The solute flux is a function of the salt permeability and the solute concentrations at the membrane surface and in the permeate water. It can be expressed as (Huehmer and Voutchkov, 2007),

$$J_s = K_s \ . \ (C_w - C_p) \tag{8}$$

Where, J_s is the solute flux through the membrane expressed in mg/m^2/s, K_s is the salt permeability through the membrane in m/s, and C_w and C_p are the membrane wall concentration and the permeate salt concentration respectively expressed in mg/L. Since the RO membrane has a high salt rejection, the value of the permeate concentration is very low as compared to the concentration at the membrane wall. Neglecting the permeate concentration the above equation can be rewritten as,

$$K_s = \frac{J_s}{C_w} \tag{9}$$

The permeate concentration can also be expressed as the ratio of solute flux and the water flux through the membrane,

$$C_p = \frac{J_s}{J_v} \tag{10}$$

And the concentration of salt at the membrane surface is higher than the bulk feed concentration (C_b) by a factor equal to the concentration polarization factor. Thus,

$$C_w = f_{cp} \ . C_b \tag{11}$$

Thus, combining equations (9), (10) and (11), the salt permeability can be computed as,

$$K_s = \frac{C_p \ . \ J_v}{f_{cp} \ . \ C_b} \tag{12}$$

A model was developed that used the experimental data for permeate and feed concentrations, and permeate flux to calculate the salt permeability of the membrane. The concentration polarization factor was assumed to be 1.5 (Ladner et al., 2010).

Computing mass transfer coefficient

Another important membrane transport parameter that governs the performance of the RO membrane is the mass transfer coefficient. The equation for the average mass transfer coefficient in film theory as given by Kim and Hoek (2005) is,

$$k = 0.807\left\{\frac{\gamma . D^2}{L}\right\}^{1/3} \qquad (13)$$

Where k is the mass transfer coefficient (m/s), D is the diffusion coefficient of NaCl which is 1.7 x 10^{-9} m^2/s (Kim and Hoek, 2005) and L is the length of the channel (m). γ is called the wall shear rate across the membrane. It is dependent on the average crossflow velocity across the channel. It is expressed in per second and calculated as,

$$\gamma = \frac{3 . u}{h} \qquad (14)$$

Where u is the average crossflow velocity in m/s and h is the half-channel height (m). The average crossflow velocity is computed from the feed flow rate and the cross sectional area of the membrane channel as,

$$u = \frac{Q_f}{2 . h . w} \qquad (15)$$

Where Q_f is the feed flow rate across the RO membrane in m^3/s and w is the width of the flow channel in m. The above equations are from the film theory are applicable to an open flow channel i.e., a membrane flow channel that does not use a feed spacer (Kim and Hoek, 2005). However, the commercially available RO membranes are always accompanied by the feed spacers to improve the performance of the system. They are used for introducing turbulence at the membrane surface to reduce the buildup of concentration polarization on the membrane surface. This introduces complexities in the hydrodynamics at the membrane surface as the channel area does not remain constant due to the porous feed spacer. This affects the flux prediction and hence must be taken into account.

In the current study, the bench-scale RO unit uses a feed spacer to simulate real RO working conditions. Hence, the models developed in this study account for the effect of this feed spacer by multiplying the channel area by the porosity of the feed spacer to obtain effective channel area. The porosity of the spacer was assumed to be 88 % (Schock and Miquel, 1987). Thus, the effective cross sectional membrane channel area,

$$A_{eff} = 2 \, . \, h \, . \, w \, . \, \varepsilon_{spacer} \tag{16}$$

Where, A_{eff} is the effective cross sectional area of the membrane channel in m^2 and ε_{spacer} is the porosity of the feed spacer. The effective cross sectional area for of the bench-scale membrane cell was 122.05 cm^2.

Further, the crossflow rates across the membrane were measured during the bench-scale experiments by measuring the volume of the water leaving the membrane cell as crossflow per unit time. The measurements were taken at different pump speeds and pressures. The crossflow velocity can be computed by dividing the crossflow rate by the effective membrane area as,

$$u = \frac{Q_c}{A_{eff}} \tag{17}$$

The equation for the wall shear rate is also modified to,

$$\gamma = \frac{3 \, . \, u}{h \, . \, \varepsilon_{spacer}} \tag{18}$$

The values of the crossflow velocities at corresponding pressures are shown in Figure 4 for three different pump speeds (data collected by Ying Sun). The linear relationship is used to calculate the crossflow velocity at any pressure. Once the crossflow velocity is known, the mass transfer coefficient is calculated using equation (13).

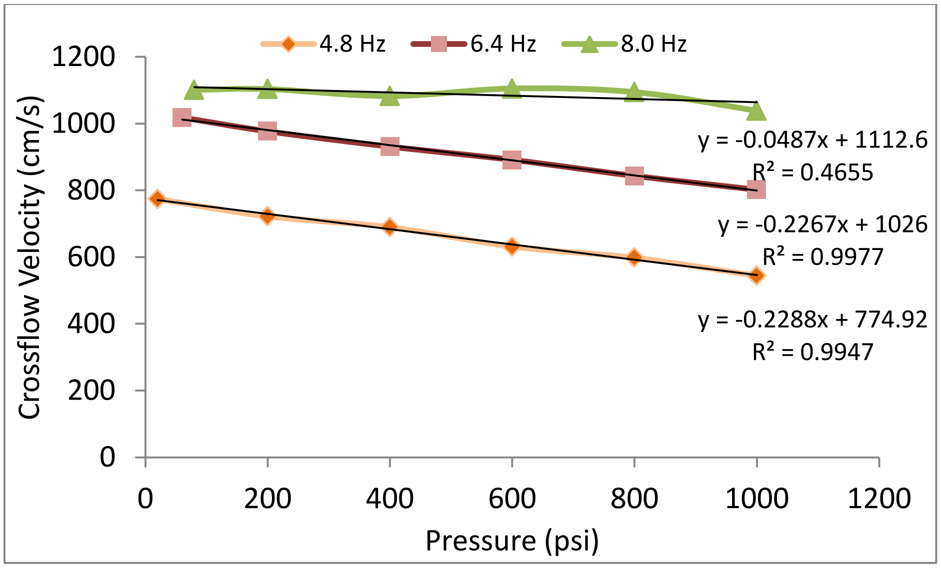


Figure 4. Plot of the crossflow velocity and the pressure at different pump speeds.

Predicting the permeate flux

Once the membrane transport parameters are computed, the performance of the system can be modeled. The important parameters that were predicted are the permeate water flux and concentration polarization; permeate concentration and the salt rejection. The flowchart for these predictions is shown in Figure 5.

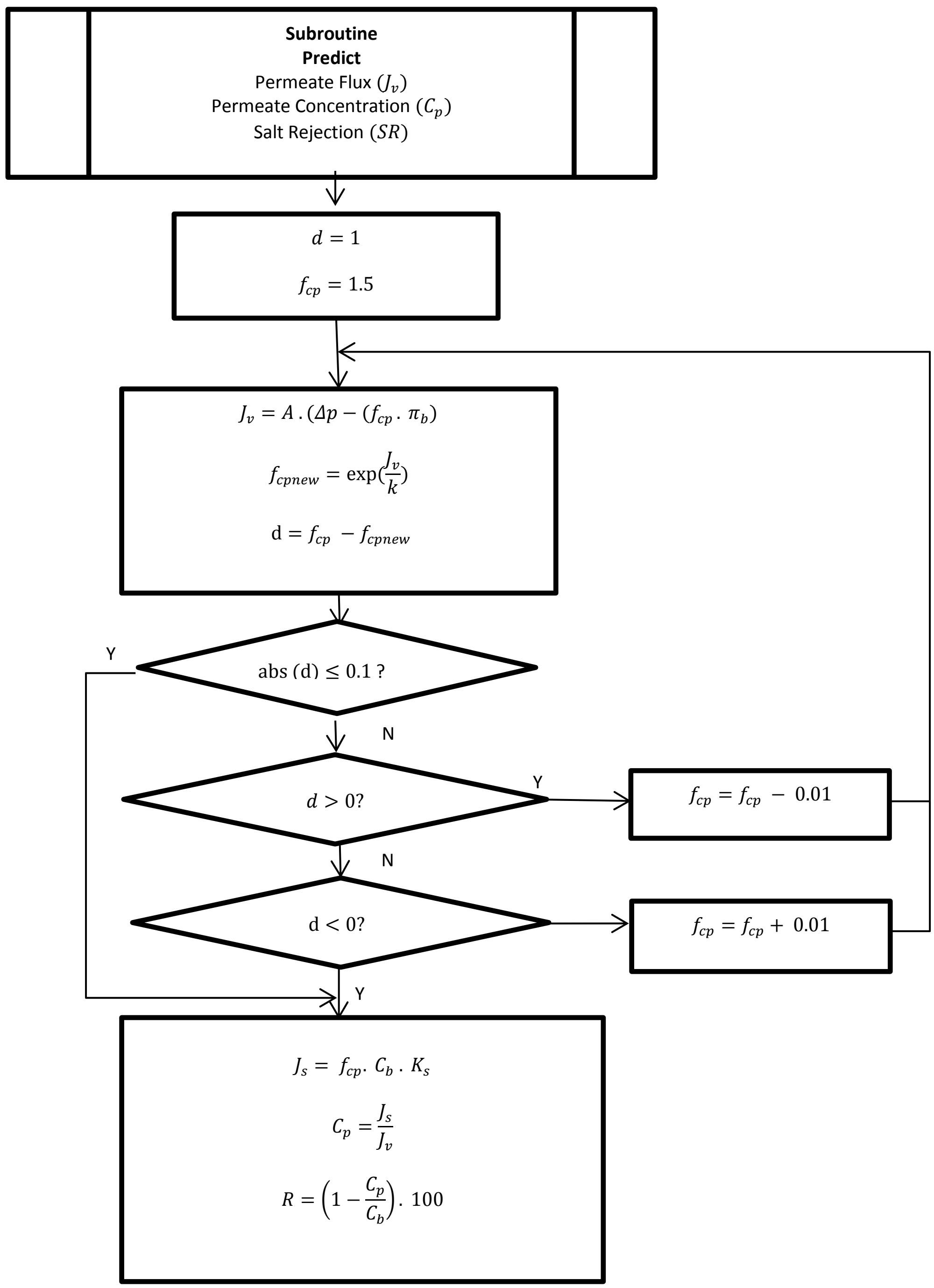


Figure 5. Flowchart depicting the prediction of flux, concentration polarization and salt rejection.

A key factor that makes the permeate flux lower than would be expected from simple flux equations is concentration polarization. According to the film theory, concentration polarization is given by the equation (Kim and Hoek, 2005; Taniguchi and Kimaru, 2004),

$$\frac{C_w - C_p}{C_b - C_p} = \exp\left(\frac{J_v}{k}\right) \quad (19)$$

or

$$f_{cp} = \exp\left(\frac{J_v}{k}\right) \quad (20)$$

Where, C_w is the salt concentration at the membrane wall, C_p is the permeate salt concentration, C_b is the bulk feed water concentration, f_{cp} is the concentration polarization factor, J_v is the permeate flux (m/s) and k is the mass transfer coefficient (m/s). Alternately, the permeate flux can be expressed as a function of pressure and the water permeability as,

$$J_v = A\,.\,(\Delta p - \Delta \pi_m) \quad (21)$$

Where J_v is the flux in m/s, Δp is the applied differential pressure in kPa, $\Delta \pi_m$ is the trans-membrane-osmotic pressure in kPa and A is the water permeability coefficient of the membrane.

The trans-membrane osmotic pressure is the difference between the osmotic pressures at the membrane wall (π_w) and the permeate side pressure (π_w),

$$\Delta \pi_m = \pi_w - \pi_p \quad (22)$$

However, the permeate side pressure is negligible and can be assumed to be equal to the atmospheric pressure. The osmotic pressure on the membrane wall is greater than the bulk osmotic pressure (π_b) by a factor equivalent to the concentration polarization. Thus, π_w can be calculated by multiplying the concentration polarization factor to the bulk osmotic pressure.

$$\pi_w = \pi_b \, . f_{cp} \tag{23}$$

Therefore equation (21) for flux can be rewritten as

$$J_v = A \, . (\Delta p - (f_{cp} \, . \pi_b)\,) \tag{24}$$

Equations (20) and (24) were solved using an iterative method in Matlab to predict the flux and the corresponding concentration polarization factor as shown in Figure 5. The predicted flux was plotted with the actual flux as calculated from the experimental data to validate the prediction. The predicted flux was adjusted to match the actual flux by varying the water permeability coefficient. In other words, the discrepancy in the actual and predicted values is due to the membrane resistance and this discrepancy was accounted for by adjusting the value of water permeability coefficient.

Predicting permeate concentration and salt rejection

The concentration of the permeate determines the quality of water. As the RO membrane has a high salt rejection, the permeate salt concentration must be below the standard of 500 mg/L (El-Ghonemy, 2012). The permeate concentration is determined by computing the solute flux through the membrane as given in equation (8) used for calculating the salt permeability. The equation can be rewritten as,

$$J_s = f_{cp} \, . C_b \, . \, K_s \tag{25}$$

In which, the value of K_s was determined as described previously, the concentration polarization factor was obtained during the permeate water flux prediction and the feed concentration was obtained from the experimental measurements. The permeate concentration was then computed by taking the ratio of the solute flux and the permeate water flux as described in equation (10).

The quality of the water produced is usually expressed in terms of the observed salt rejection of the membrane which depends on the permeate and feed concentrations. The observed salt rejection was calculated using the following equation,

$$SR = \left(1 - \frac{C_p}{C_b}\right).100 \tag{26}$$

The above steps are presented in the flow chart presented in Figure 5.

2.2 Modeling Conceptual Full-Scale Wind-RO System

The film theory modeling approach was validated with bench-scale experimental results and the same theory was used to model a conceptual full-scale system. The conceptual full-scale RO desalination system was modeled such that it can be solely operated by using wind energy. The general system design comprises of a wind turbine, high pressure pump, pressure vessel tank and spiral wound RO elements arranged with different design configurations for comparison.

The configurations used to compare the performance of a wind driven RO desalination system were: 1. a wind-RO system without an energy storage tank (a conventional system), 2. a wind-RO system with a single air-pressure energy storage tank, 3. a wind-RO system with three air-pressure energy storage tanks. The conventional wind-RO desalination system was used as a basis for comparing its performance with the air-pressure energy storage setup. As conventional RO systems are not designed to work under variable pressure and discharge conditions, it was expected that the conventional wind-RO setup would perform poorly as compared to the energy storage setup. The inability of a wind energy driven conventional RO system to work efficiently provided the motivation to develop a design for the desalination system that can be powered by a wind turbine.

Since the RO desalination process requires a continuous and high energy supply and wind energy is intermittent and of low intensity, integration of the two systems requires an energy storage mechanism that can provide energy to develop the necessary pressure and dampen the fluctuations caused in the flows. A simple storage tank pre-charged with pressurized air can provide the necessary dampening effect and also develop pressure energy required for desalination. Thus, the desalination system and the wind turbine are coupled together by inserting an air-pressure energy storage tank between the feed pumping system and the RO membranes. Feed water will be pumped into the energy storage tank where it will get pressurized, and then it will be discharged into the membrane units in a controlled manner.

Some of the simplifying assumptions made in modeling the full-scale wind-RO configurations are listed below.

1. The flow development was instantaneous in the RO elements under varying pressures and flow rate conditions. The performance of the system can be completely predicted with film theory equations.
2. The membrane transport parameters calculated from the bench-scale experiments are valid for spiral wound membrane elements and for the energy storage tank setup. In other words, the geometry of the spiral wound membranes or the pressure variation due to the fluctuating wind conditions does not affect the membrane transport parameters.
3. The effects of temperature variation due to fluctuating gas pressures inside the air-pressure energy storage tank are neglected. It was assumed that the system was operated at a temperature of 21± 1°C.
4. The frictional losses and leakages through the pipes and pump were assumed to be negligible and the pressure loss across each spiral wound membrane element was constant.

5. The variable speed drive pump works with a constant efficiency at highly variable pressures and feed water discharge conditions. Also, it was assumed that the wind turbine can provide the required high torque to the pump during startup.
6. Membrane deterioration was neglected.

Wind regime

Actual wind data recorded at 30-second intervals on a daily basis were obtained from an online resource tool developed by Kansas State University. (wind.ece.ksu.edu/dataselect.php and sustain.ece.ksu.edu/daq/data.php). From the database of wind speeds available on these websites, four different wind patterns were selected such that a wide range of wind speeds was covered for testing the system. The wind turbine used to record the wind speeds is located on Kansas State University campus. The total time for which wind data was available for each pattern differed from 9.48 h to 23.13 h. Hence, to maintain consistency, the longer data sets were shortened and the wind speed duration was assumed to be 9.48 h.

Further, as the wind speeds were recorded at every 30 seconds, the wind data were converted into a step input by assuming a constant wind speed over the period of 30 seconds. The four wind regimes, namely wind pattern A, wind pattern B, wind pattern C and wind pattern D are plotted in Figure 6. The average wind speed of the 4 patterns was 6.75 m/s, 4.86 m/s, 4.30 m/s, and 1.89 m/s respectively.

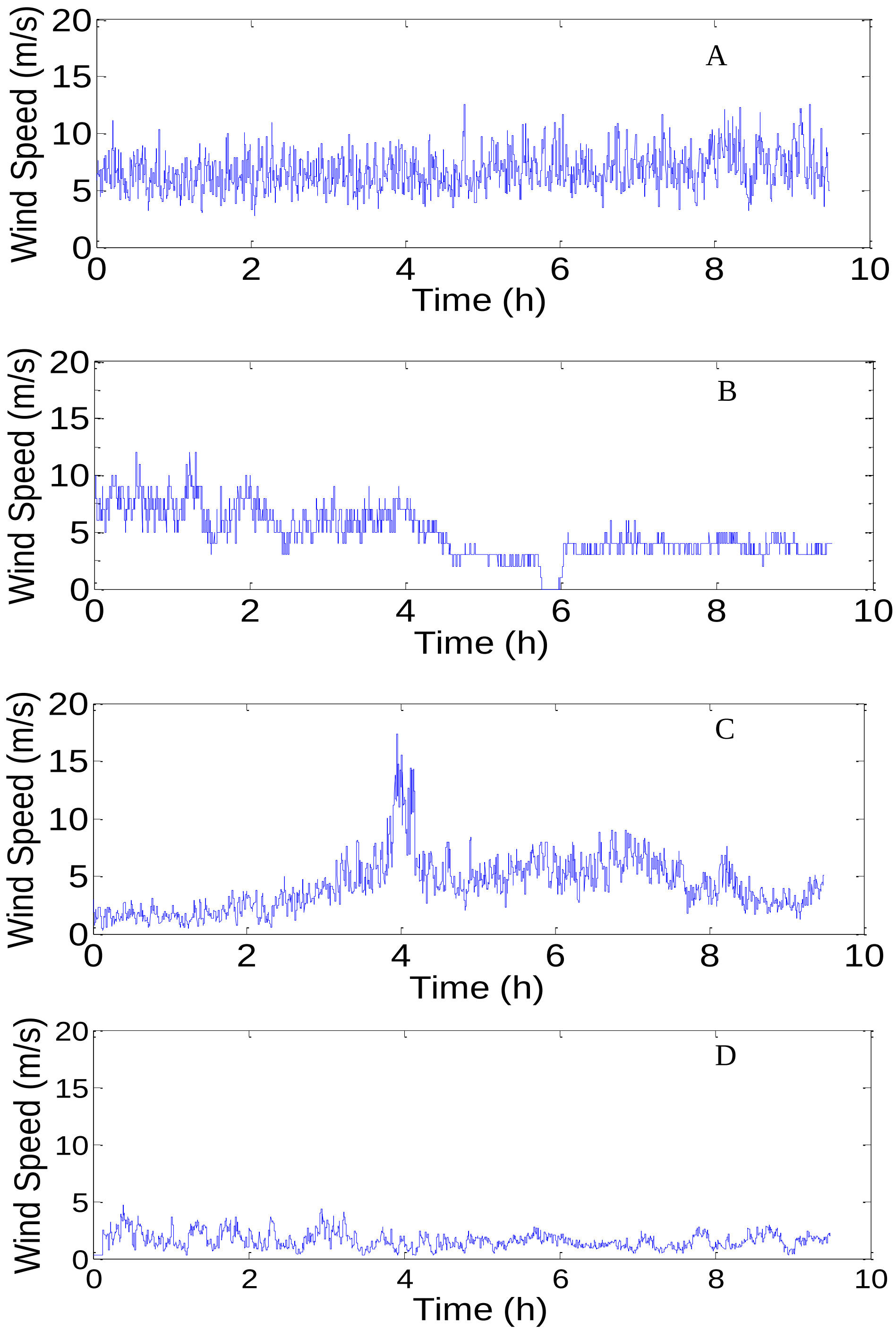


Figure 6. Wind patterns A, B, C and D used as inputs to modeling runs.

Spiral wound RO membrane element

Specifications of a commercially available standard spiral wound RO membrane were used as modeling parameters. Specifications were obtained from Dow Filmtec 8" element and used as the basis of design (Huisman, 1993). They are presented in Table 1

Table 1. SW30HR-320 Membrane Specifications.

Parameter	Value	Unit
Type	Filmtec SW30HR-320 Seawater Reverse Osmosis Element	-
Feed Spacer	34	mil
Element diameter (I.D.)	0.201	m
Length	1.016	m
Permeate tube diameter (I.D.)	0.029	m
Active Area	320	m^2
Maximum operating pressure	6,900 (1,200)	kPa (psig)
Permeate flow rate	23	m^3/d
Minimum salt rejection	99.6	%
Stabilized salt rejection	99.75	%

The above values of permeate flow rate and salt rejection are obtained by testing the RO membrane element under standard test conditions of feed concentration of 32,000 mg/L, applied pressure of 800 psi and the recovery of 8%. These values are therefore used as the basis for calculating the flow rates and concentrations under varying pressures.

From the known parameters, some other useful data was calculated, such as the membrane feed flow and cross flow rates. The recovery of a membrane element can be defined as the ratio of the permeate flow rate and the feed flow rate. Therefore the required feed flow rate for the membrane element was calculated using the equation,

$$Q_f = \frac{R}{Q_p} \tag{27}$$

The standard feed flow rate to the SW30HR RO membrane was 287.5 m^3/d for a recovery of 8 % and permeate flow rate of 23 m^3/d.

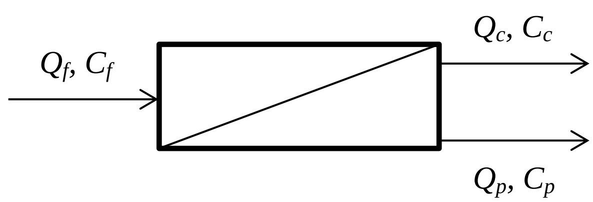


Figure 7. Flow through an RO membrane.

The crossflow rate was calculated using the flow balance equation (Figure 7) through the RO membrane element as,

$$Q_f = Q_c + Q_p \tag{28}$$

The standard crossflow rate was found to be 264.5 m^3/d. The values of flow rates and the recovery were used as standard values and to compare and calculate flow rates through the RO membrane in the designed system.

Another important parameter required in the design was the energy required to pump feed water. The power required to pump a feed at the standard feed flow rate of

287.5 m^3/d at the a standard pressure (Δp) of 800 psi (5,520 kPa) was calculated as 18.36 KW, using the equation

$$W_{pump} = Q_f \, . \, \Delta p \tag{28}$$

Wind turbine

For the wind-RO system, the wind turbine was the source of power supply and therefore it was important to select a turbine that can provide energy to pump water at a high pressure over a wide range of wind speeds. The maximum power required was calculated by considering the efficiencies of the turbine as well as the pump it was connected to.

Assuming a low speed roto-dynamic pump with 1:4 step up coupled to the wind turbine with an efficiency of 60 %, the power transmitted to the pump from the wind turbine was calculated as,

$$W_{Pump} = 0.6 \, W_T \tag{29}$$

Theoretically, the power required by a pump under standard pressure and flow rate was approximately 18 kW. Therefore, the wind turbine requires generating a power of approximately 38 kW. Thus, a mid-sized, thin blade wind turbine of diameter 15 m and rated power of 50 kW was selected (www. engineeredwindsystems.com). The turbine power at a given wind speed was calculated using the following equation.

$$W_T = \frac{1}{2} \, . \, \rho_{air} \, . \, C_{pt} \, . A_{rotor} \, . \, {V_{wind}}^3 \tag{30}$$

where W_T is the power generated in KW, ρ_{air} is the density of air which is assumed to be 1.1839 kg/m^3, A_{rotor} is the area swept by the blades of the turbine in m^2 and V_{wind} is the wind speed in m/s and C_{pt} is the power coefficient which is an term equivalent to the efficiency of the turbine. The swept area of the turbine was 176.7 m^2 and C_{pt} was assumed to be 0.4 for this wind turbine.

In a wind-RO system, due to the instantaneous variation in the wind speeds, the power available was also varying. Hence there was variation in the pump flow rates and the pressures also. The instantaneous pump flow rates were calculated by using the pump affinity laws. The pump affinity laws are used to predict the pump performance at different pump speeds. The dimensionless relationship for the discharge and power for a pump at two different pump speeds having a fixed impeller diameter is given by,

$$\frac{Q_1}{Q_2} = \frac{N_1}{N_2} \quad and \quad \frac{W_1}{W_2} = \frac{{N_1}^3}{{N_2}^3} \tag{31}$$

Where Q is the discharge rate, N is the pump speed and W is the pump power (The Pump Handbook, McGraw-Hill). The above equation can be rewritten to obtain a relation between the pump power and discharge rates as,

$$\frac{Q_1}{Q_2} = \left(\frac{W_1}{W_2}\right)^{1/3} \tag{32}$$

For the wind-RO system this comparison was used to find the discharge rate of the pump for the available power as,

$$Q_{pump} = Q_{Std} \cdot \left(\frac{W_{Pump}}{W_{Std}}\right)^{1/3} \tag{33}$$

Q_{std} and W_{std} are the standard feed flow rate and power as calculated above and Q_{pump} and W_{pump} is the instantaneous flow rate and power. Once the power and flow rate are calculated the instantaneous pressure delivered by the pump can be computed using equation (33).

2.2.1 Base Case Design: Conventional RO Setup with Wind Turbine

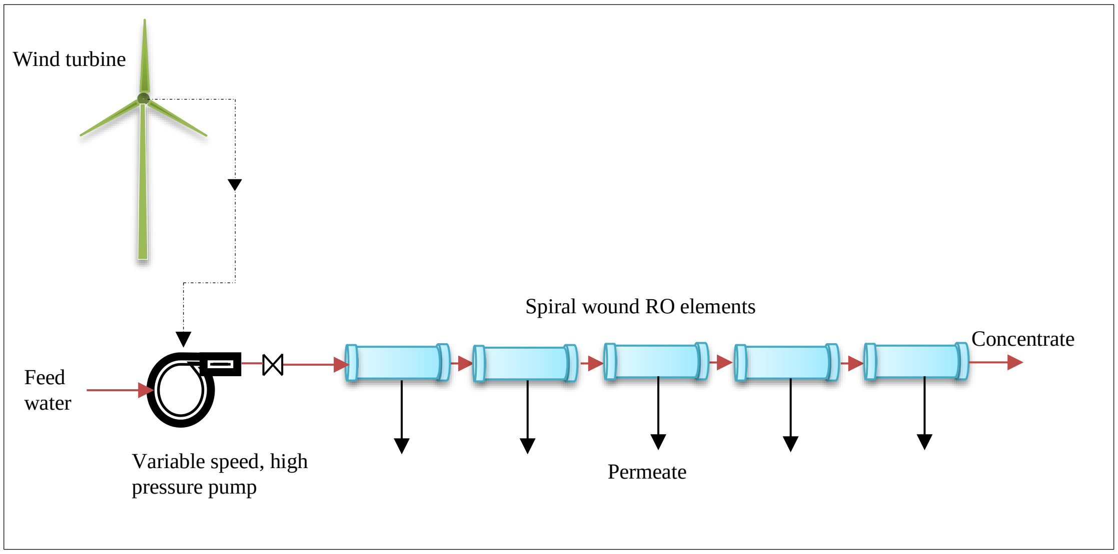


Figure 8. Conceptual design of a conventional wind-RO desaliantion system.

The base case, full-scale model follows a simple design comprised of a conventional RO desalination system connected to a wind turbine that supplies energy required to operate the system. The membrane units receive the feed water directly from a roto-dynamic pump driven by a wind turbine as shown in Figure 8. In this design, there was no mechanism to dampen the fluctuations in pressures and discharges caused by the varying wind speeds and the membranes are subjected to pulsating feed flow rates.

Since there was no energy storage in this system, this system cannot be operated in a batch mode. The system was modeled to run continuously, as do most conventional RO desalination systems. Clearly, during a dead wind condition, there will be no flow and the pressure would subsequently drop to atmospheric pressure. The system will operate intermittently depending on the availability of the wind power. It was assumed that the system will start and shutdown instantaneously. An array of five spiral wound membrane elements connected to each other in series was modeled. The flowchart describing the steps involved in modeling the system is shown in Figure 9. The results for the conventional system are presented in chapter three.

The model for the conventional system design begins with defining the membrane, pump and turbine specifications. Since the wind speeds were recorded at every 30 seconds, the program extracts the values of the wind speeds and converts them into step inputs of corresponding magnitudes at each second.

The program then computes the instantaneous turbine power, pump power, pump flow rate and pressure using the equations as described above. In the conventional system, the water discharged by the pump directly enters the first membrane element. As the membrane elements are connected in series, the subsequent membrane elements receive crossflow from the previous element. The permeate flux, permeate concentration and the salt rejection was calculated for each membrane using the film theory equations (Kim and Hoek, 2005). The system was tested for four different wind regimes. For comparison the system was also modeled for a hypothetical steady wind regime. In this, a constant wind speed was assumed to clearly demonstrate the inefficient performance of a conventional system under varying energy input.

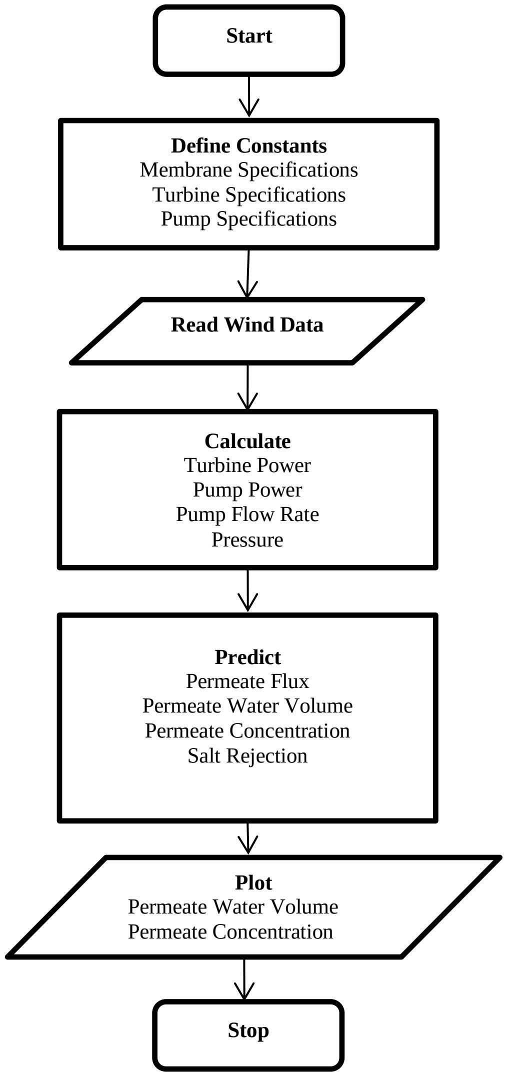


Figure 9. Flow chart depicting the modeling steps for the conventional wind-RO system.

2.2.2 Proposed Design: Wind-RO with Integrated Energy Storage

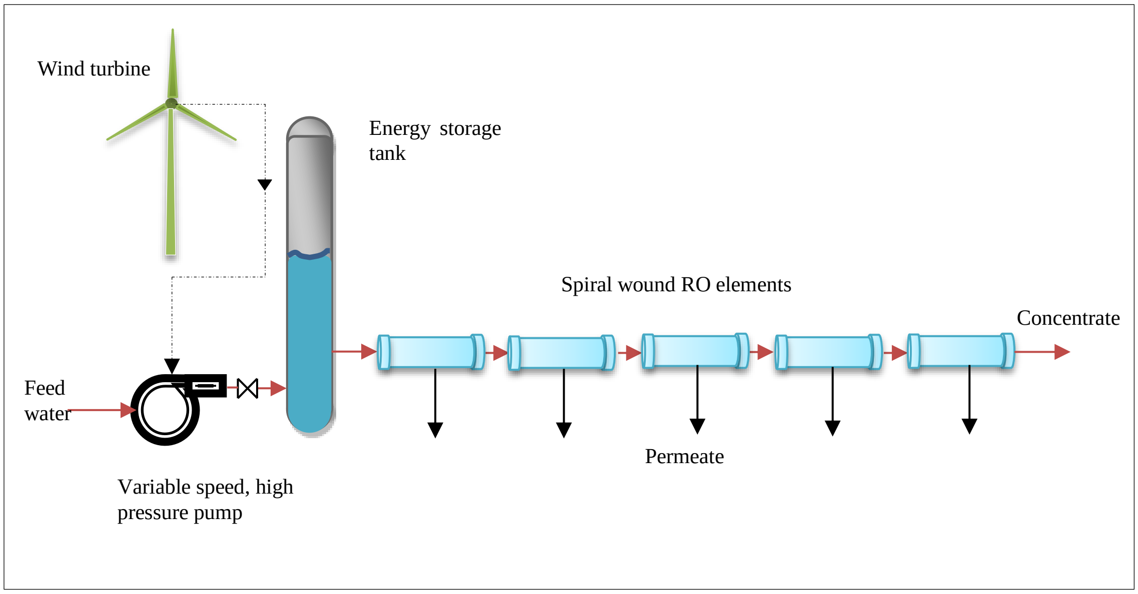


Figure 10. Conceptual design of the proposed wind-RO system with a single energy storage tank.

The wind-RO desalination system with integrated energy storage mechanism is shown in Figure 10. The system was modeled to operate in batches. In other words the system will operate in a dual stage process: (1) fill and (2) desalinate. In the first stage, feed water is pumped inside the pre-charged pressure vessel so that the gas inside the pressure vessel is compressed, which increases the pressure inside the vessel. The water is pumped until the pressure reaches up to 1,000 psi. In the second stage, the pumping of feed water into the pressure vessel was stopped and the pressurized feed water is discharged into the RO membrane unit.

The advantage of designing the system to operate in a batch mode was that the desalination process was isolated from fluctuating wind energy characteristics. In the first stage, while the energy storage tank was filled with the feed water that was pumped at varying pressures and discharges, the RO membranes are prevented from being subjected to these fluctuating conditions. Then only in the second stage, the pressurized feed water was allowed to be discharged in a controlled manner and till the pressure drops to a certain set lower limit, into the membrane units. Such a system would be simple to operate and design and also offers a better control over the system operation. Further, this method of operation would ensure that the RO membranes do not deteriorate as a result of pulsating discharges. Also, since this system operates solely by wind energy, it will be more economical in areas where electricity was provided through long transmission lines. Furthermore, a pressure vessel is simple to construct, economical and maintenance free. The only disadvantage of this system would be the membrane downtime experienced during the fill period as compared to a continuously operated system. However, the system provides many advantages in terms of operability and designing.

In this study, the system was designed with a single energy storage device connected to an array of membrane elements. The performance of this simple design was evaluated and compared with the conventional wind-RO system. The performance of this system was also analyzed by varying the following input parameters to determine the optimum design conditions.

Wind Regime

The most important input parameter for a wind driven desalination system was the type of wind pattern available at the location. Expectedly, the system will perform efficiently at high wind speeds. However, the performance variation for different wind patterns was analyzed to have a better understanding of the feasibility of the system.

Initial air pressure

The energy storage tank was pre-charged with air at a certain pressure so that when the water was pumped inside it, the pressure rises as the air compresses. In general, an RO desalination process requires a pressure between 800 psi – 1,200 psi for desalination to occur. The system was therefore modeled to pump water in the energy storage tank until the pressure inside it reached 1,000 psi (6,900 kPa).

Volumes of the air inside the tank were calculated using the ideal gas law given as,

$$PV = nRT \tag{34}$$

Where, P is the initial pressure of the air filled inside the empty energy storage tank, V is the volume of the energy storage tank, which was also the volume of the air when it was empty. Thus, by setting the initial air pressure and tank volume, the value of nRT can be determined. For the purpose of modeling, it was assumed that the ideal gas law was valid at high pressures and the temperature was fairly constant so that the value of nRT was a constant for a given initial air pressure and energy storage tank volume.

By knowing the volume of feed water pumped inside the energy storage tank, the new pressure inside the tank can be computed. Table 2 shows the water and air volumes at 1,000 psi inside a 15 m^3 energy storage tank with different initial air pressures. At an initial air pressure of 600 psi, 6,000 L (6 m^3) of feed water was required to attain a pressure of 1,000 psi; whereas, if the tank was not pre-charged with pressurized air, 14,780 L (14.78 m^3) water was required to reach 1,000 psi inside the pressure vessel.

Table 2. Water and air volumes inside a 15 m^3 energy storage tank with different initial air pressures.

Tank volume	**Initial air pressure**		***nRT***	**Air Volume at 1,000 psi**	**Feed water volume at 1,000 psi**
m^3	**psi**	**kPa**	**kPa m^3**	**m^3**	**m^3**
15	600	4,134	62,010	9	6
15	400	2,756	41,340	6	9
15	200	1,378	20,670	3	12
15	14.7	101.28	1,519.25	0.22	14.78

Clearly, a high initial air pressure would require less time and energy to achieve a pressure of 1,000 psi during the first filling operation. However, a smaller volume of feed water will be filled inside the tank. Therefore, the initial air pressure must be selected such that during the desalination phase, the tank must not run out of feed water. The modeling experiments included determination of permeate flow rate and permeate concentration at different initial air pressures.

Lower pressure limit

During the desalination stage the pressure inside the pressure vessel was allowed to decline gradually as desalination occurs. Hence, the value to which the pressure was allowed to drop was an important factor that needs to be considered. From equations (5) it can be observed that the permeate flux decreased with a decrease in the applied pressure. If the permeate flux reduces, then from equation (10), the permeate salt concentration increases.

Furthermore, the lower pressure limit was set to be very high then most of the pressure energy available for desalination will remain unused and the tank would require frequent filling. On the other hand, if the limit was set too low, there was a chance of the energy storage tank running out of feed water. The initial air pressure governs the value to which the lower limit can be set. For example, if the initial air pressure was 800 psi, then the lower pressure limit cannot be set below 800 psi since the tank would run out of feed water when the pressure declines below 800 psi. Thus, the lower pressure limit must be selected such that it optimizes the operation of the system. In this study, a range of lower pressure limit was selected and modeling experiments were performed to evaluate the performance of the system.

Energy storage tank volume

The volume of the energy storage tank governs the wind energy required to fill the tank and consequently the time required for filling and desalination. The effect of variation in the tank volume on the output of the system was analyzed.

Number of RO membrane elements

The permeate discharge rate and the water quality are primarily dependent on the performance of the RO membrane. In this study, the number of membrane elements selected to examine the performance of the system were 3, 5, and 8. The models were tested for RO elements arranged in series. A parallel configuration with multiple stages were not selected simply because in this design, if the membranes connected to a single pressure vessel are arranged in a parallel configuration, it would result in a pressure loss at a faster rate though the energy storage tank.

Clearly, the above mentioned parameters are interdependent and have a combined effect on the system operation. The results and analysis of the wind-driven of a single energy storage tank were used to then model a full-scale system comprising of multiple energy storage tanks each attached to an array of spiral wound membrane elements and driven by a single wind turbine. Further, the water production and the permeate salt concentrations were the main basis for comparing the systems. Since a single wind turbine was used for modeling, the amount of energy available is the same for all systems for a particular wind pattern. Hence, this study focused on designing systems that can "capture" the most energy and turn it into highest water productivity.

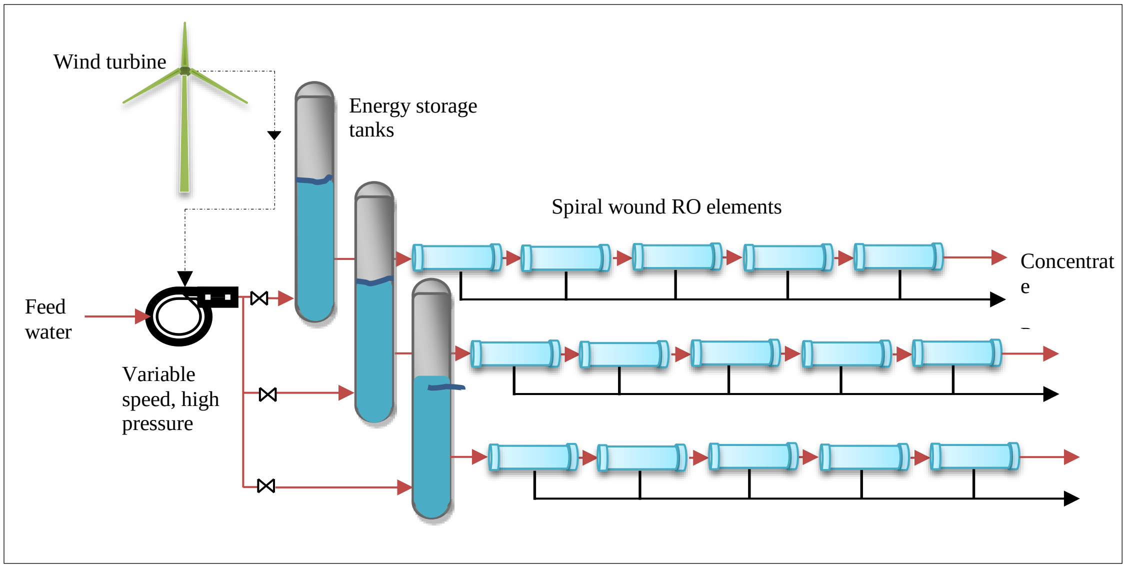


Figure 11. Conceptual design of the proposed wind-RO system with three energy storage tank.

In the single energy storage setup, during the fill period, the membranes remain unproductive as there was no desalination taking place. Alternately, during the desalination period, the wind energy available was not utilized as feed water from the energy storage tank was being discharged into the membrane elements during this phase. Thus, neither the wind power nor the membrane area was used to its maximum capacity in this design, which is not economical. Thus, in order to make the system feasible, the system must be designed to consist of more than one energy storage tank and operated in such a way that filling and desalination can occur alternately in each tank.

Therefore the wind-RO desalination system was designed with three energy storage tanks connected to a single pump and wind turbine. The design is shown in Figure 11. The working of this system is the same as the single energy storage tank setup. However, this system was modeled to operate such that when one energy storage tank is filling, the other two tanks are desalinating. In other words, the wind energy is used to pump feed water into the energy storage tanks continuously one after the other. Figure 12 shows the model developed for operation of the three energy storage tanks wind-RO system.

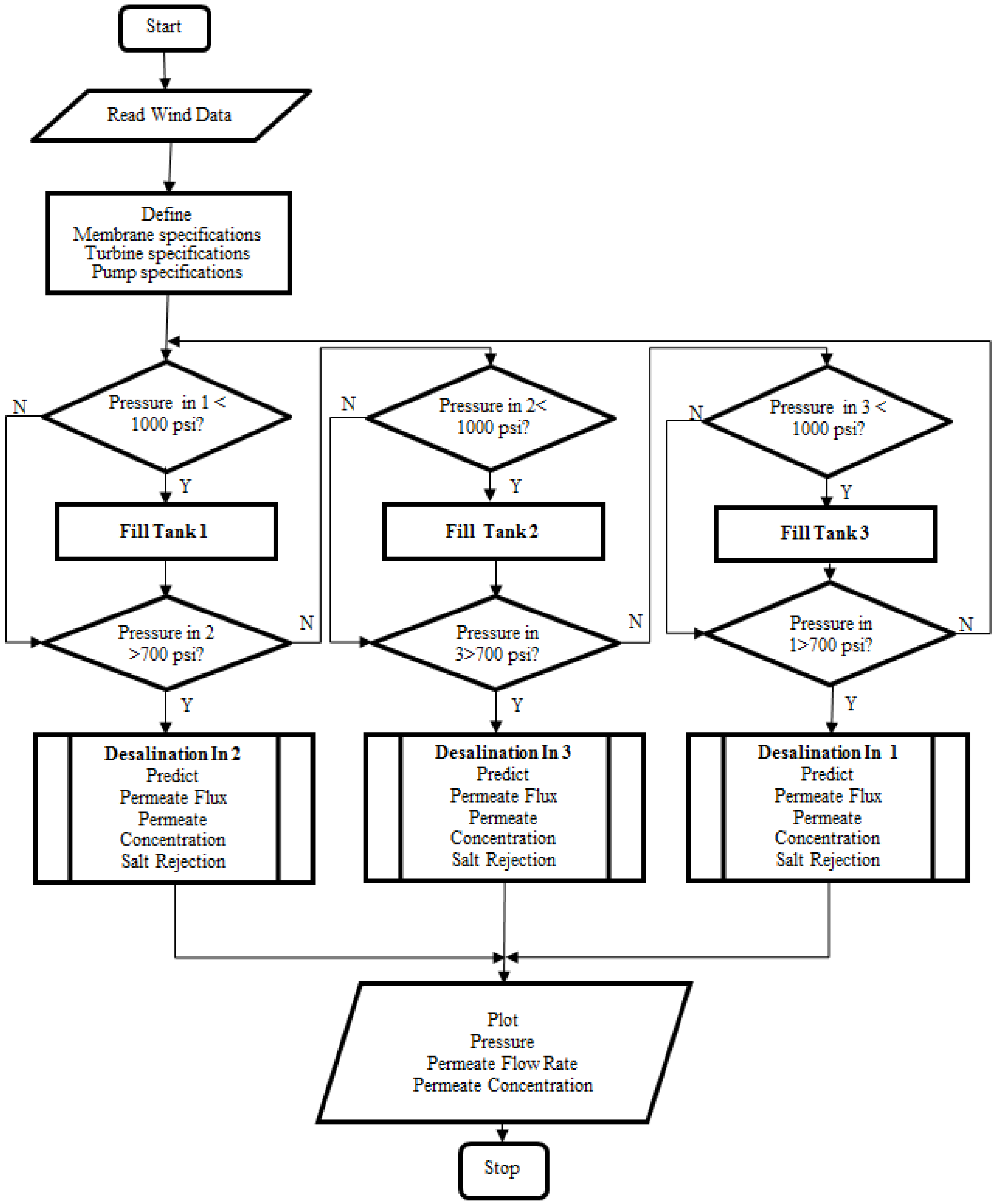


Figure 12. Flow chart depicting the operation of three energy storage tanks wind-RO system.

3. RESULTS AND DISCUSSIONS

3.1 Bench-Scale Setup Modeling

The bench-scale experiments were modeled using the film theory equations as described in the previous chapter. Figure 13 shows the experimental and modeling results simulating the batch operation in the laboratory. In these experiments, the system was operated at a pressure of 1,000 psi for approximately 2 hours to achieve a steady-state and then the pressure was allowed to gradually decrease to approximately 800 psi by controlling the crossflow. The experiment was stopped after the pressures decreased further as it would yield a negligible amount of permeate flux. For details of the experimental results please refer to Sun (2013).

Figure 13 shows the results for the experimental and modeled flux and permeate concentrations at different initial air pressures inside the air-pressure tank. The experiments were performed with initial feed concentration of 35,000 mg/L and the initial air pressures were 14.7 psi (atmospheric), 200 psi, 400 psi, 600 psi. The permeate flux was found to vary directly with the pressure and the permeate concentration increased as the flux decreased.

In the bench-scale experiments, the crossflow across the membrane was recirculated into the feed tank which increased the feed concentration. This resulted in an increase in the bulk osmotic pressure. Hence, although the pressure was maintained at 1,000 psi during the first two hours of the experiments, in Figure 13, a small decline in the flux was observed due to the increase in feed concentration. During the gradual decline in the applied pressures, a decrement in the flux. In general, for each initial air pressure, the predicted flux was found to fit the trend of the experimental results. The water permeability coefficient is usually found to vary for RO membranes, and therefore in this model the permeability coefficient was varied in the range of 0.055-0.08 lmh/psi to make the predicted values fit the experimental data.

The total permeate water volume and the overall salt concentration are shown in Figure 14. The values for the modeled permeate volume were in good agreement with the experimental values. From Figure 14 (a) shows that with an increase in the initial air pressures from 200 to 600 psi, the permeate volumes increased. Although, a permeate volume of 1.5 L at an initial pressure of 14.7 psi, this value has to be disregarded because a full-scale system working at an initial air pressure of 14.7 psi would be uncontrollable. This is explained in detail in the subsequent sections

The permeate concentration is inversely proportional to the permeate flux. This relation was clearly observed in the permeate concentration plots in Figure 13. Another factor that caused an increase in the permeate concentrations is the increase in the feed concentrations due to recirculation of the crossflow. However, for each experiment, the permeate concentration was found to be less than 500 mg/L, thus, maintaining a high salt rejection.

At the beginning, between 0-0.1 h, it can be seen that the experimental values for the permeate concentrations drops sharply. During this period, the membrane is compacting and there is a high solute flux through the membrane. This transitional state could not be modeled as the film theory equations are valid only for steady-state conditions.

Figure 14 (b) shows the experimental and modeled values for the overall salt concentrations in the permeate water at the four different initial air pressures. The values show some discrepancy in the modeled and experimental values and this may be due to the high concentrations during the transition state at the beginning of the experiments. However, the deviation in the values is negligible. Although the salt rejection in each experiment was 99 %, a slightly higher permeate concentration was obtained at an initial pressure of 400 psi as compared to the others. Also, the lowest concentration of 250 mg/L was obtained at 200 psi. It is unclear if the initial air pressure would cause the variability in the permeate concentrations. The comparison of the permeate concentrations with their respective average feed concentrations indicated that the variability was due to the difference in the feed concentrations as shown in Figure 14 (b).

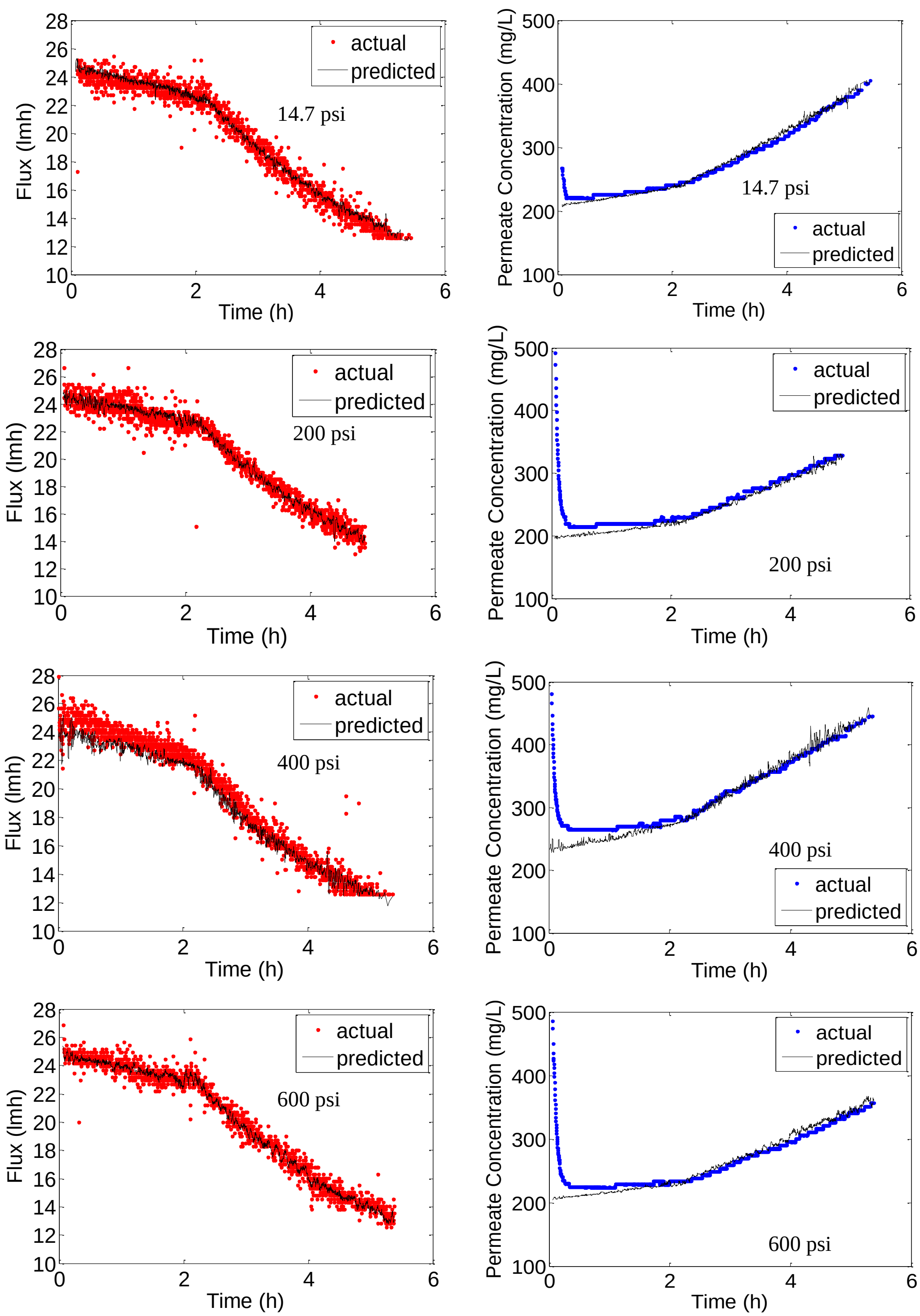
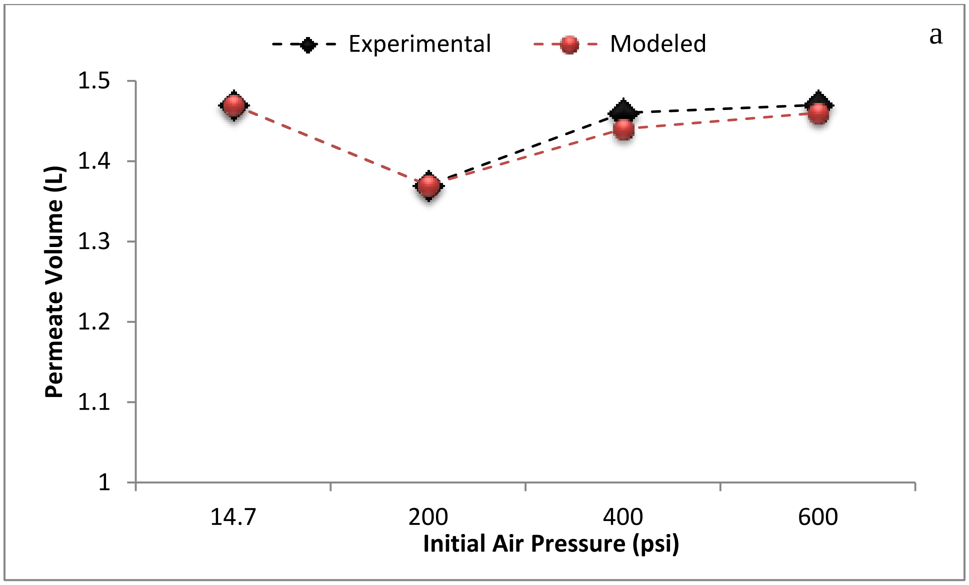


Figure 13. Experimental and modeled permeate flux and salt concentrations for different initial air pressures of 14.7 psi, 200 psi, 400 psi and 600 psi.



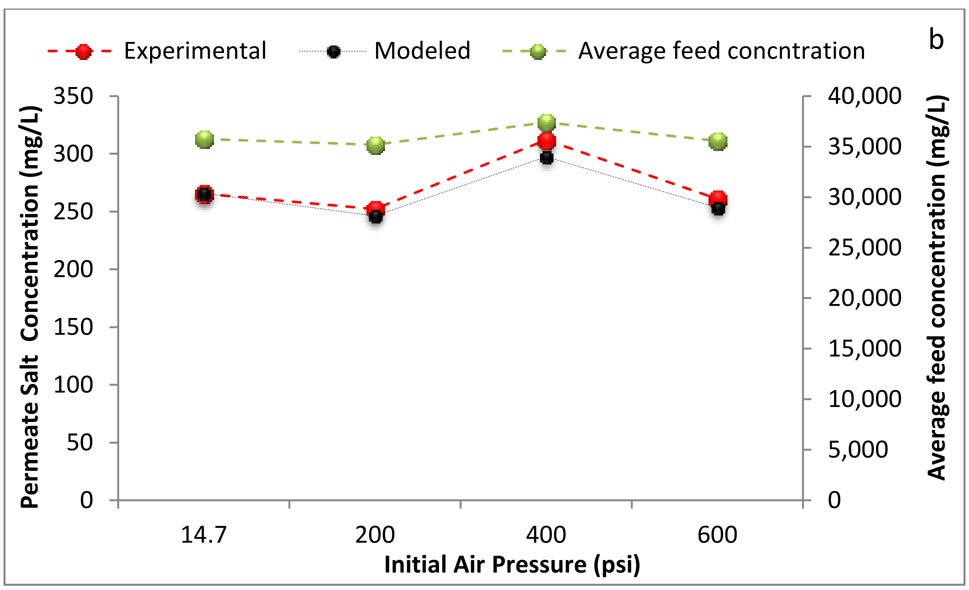


Figure 14. (a) Experimental and modeled permeate volumes at different initial air pressures. (b) Experimental and modeled pemreate salt concentrations and the average feed water concentration at different initial air pressures.

3.2 Conceptual Full-Scale Design Modeling

3.2.1 Conventional Full-Scale System

For a basis of comparison with the wind-driven energy storage RO system, a conventional system driven by a wind turbine was modeled. The design of this system is shown in Figure 8 and its performance is presented in Figures 15 and 16. The plots show the permeate flow rates and permeate concentration of the system for the 4 different wind patterns and for variable number of membrane elements. These results are also compared with steady wind conditions.

As shown in Figure 15 (a) and Figure 15(b), for the steady wind conditions and for different wind patterns, the permeate flow rates increased with the increase in the number of membrane elements. However, the flow rates for the variable wind-driven RO systems are much lower than the steady wind-conventional systems. At a steady wind speed of 10 m/s, the pump delivers feed water at a constant pressure of 985 psi which results in a high water production. But, at a steady wind speeds of 6 m/s or below, the pump assumed in this design is unable to develop the pressures required for desalination to occur and hence the system fails. Thus it can be seen that, a conventional RO system driven by wind would require high wind speeds for desalination to occur. However, since the wind speeds are highly variable, a pump connected to a wind turbine cannot supply a constant feed flow resulting in lower permeate flows. This is seen in Figure 15 (b) where the permeate volume at different wind patterns is nearly 75 % lower than at steady wind conditions. Among the four wind patterns, wind pattern A has the highest average wind speed of 6.75 m/s and hence delivers the highest water production. On the other hand, for areas with low average wind speeds such as in case of wind pattern D, the conventional system fails altogether. Since the overall water production is much lower than in case of a steady wind pattern, the results suggest a necessity of an energy storage mechanism.

Figure 15 (a) and (b) show that the permeate concentrations for the steady and variable wind conditions. It can be observed that the permeate salt concentration increased with the increase in number of membrane elements. This was because the membrane elements were arranged in series in which the crossflow from one element became the feed for next element. The crossflow is always high in salt concentration and therefore as the number of elements are increased, the feed concentration for the subsequent elements is also higher which causes an overall increase in the permeate concentration. Further, it can be seen that the concentration is higher for lower wind speeds in both the steady and variable wind patterns. As salt concentration in the permeate water is inversely proportional to the water flux through the membrane, lower wind speeds would produce lower water flux thus increasing the permeate concentrations.

Amongst the four wind patterns, the system performs best for wind pattern A which has a higher average wind speed than the others. And although the system modeled with 8 membrane elements, delivers the highest a flow rate of 20 m^3/d, its permeate salt concentration is higher than 500 mg/L. In other words, there is a tradeoff between the water quantity and water quality.

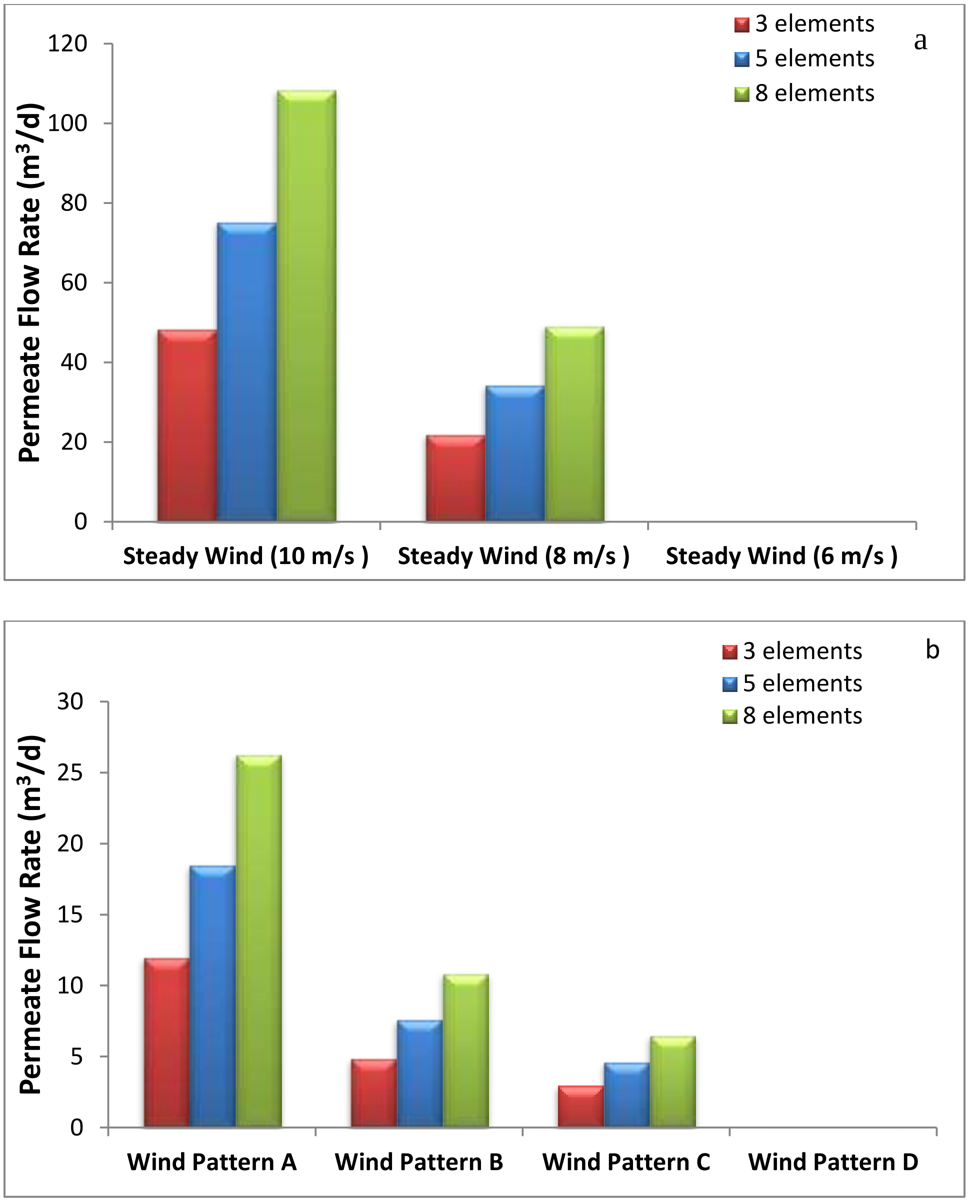


Figure 15. (a) Permeate flow rates for steady wind patterns for three different membrane elements. (b) permeate flow rates for wind patterns A, B, C, and D for three different membrane elements.

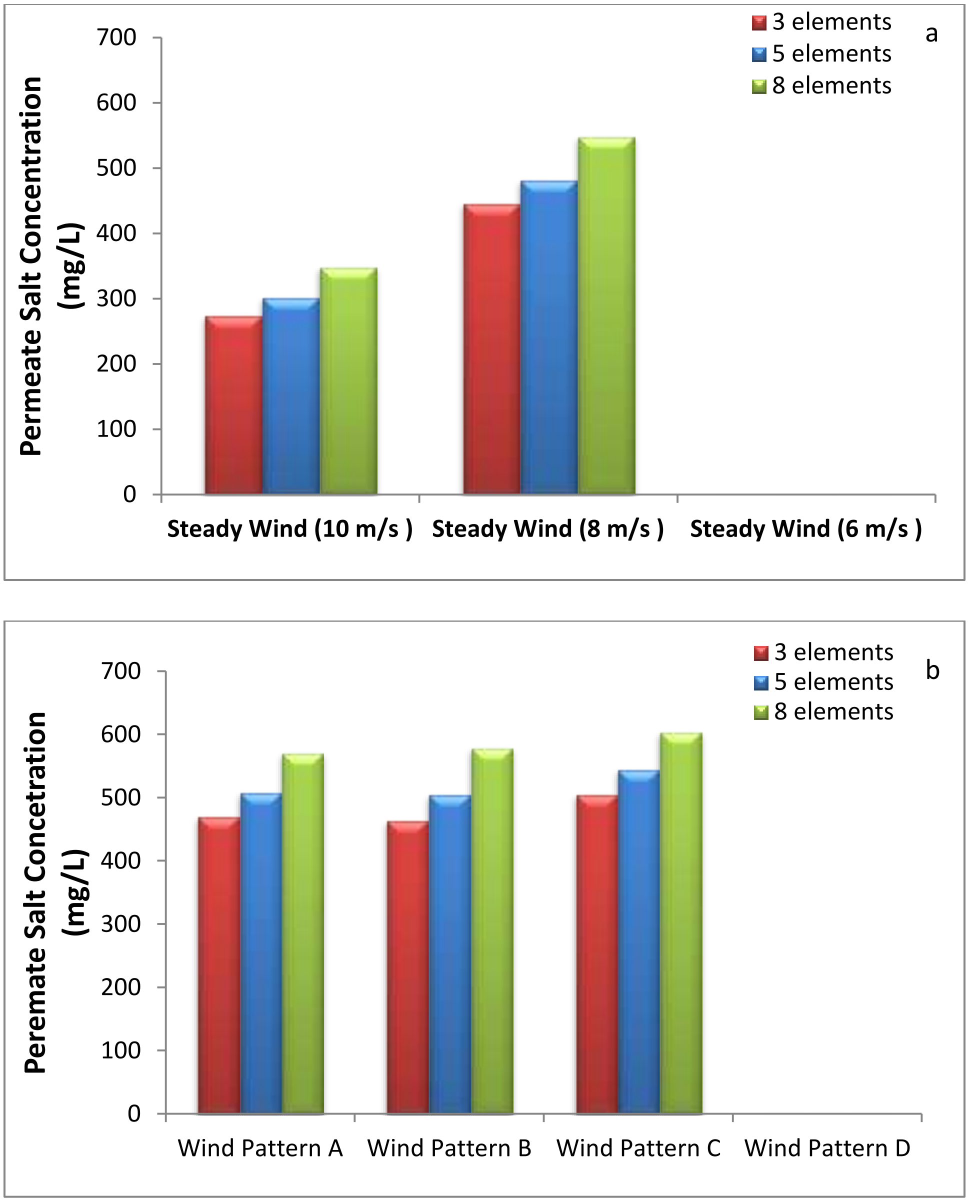


Figure 16. (a) Permeate salt concentrations for steady wind patterns for three different membrane elements. (b) permeate salt concentrations for wind patterns A, B, C, and D for three different membrane elements.

3.2.2 Single Energy Storage Tank System

Basic system operation

The performance graphs for a system with a single pressure vessel working in batch mode are shown in Figure 17 .

Figure 17 (a) shows the pressure variation inside the energy storage tank. During the fill period, the pressure inside the tank increased up to 1,000 psi (6,900 kPa) whereas during the desalination stage, the pressure declined as water discharges into the membrane elements. In the first cycle, the energy storage tank is empty and the air inside the tank is at an initial pressure 600 psi (4,134 kPa). The pressure thus increases from 600 psi to 1,000 psi and then begins to desalinate. Desalination occurs until the pressure drops to 680 psi. The cycle repeats until the wind speed is available to fill the energy storage tank. In the last cycle, the fill cycle terminates before the pressure reaches 1,000 psi because no wind energy is available and the desalination cycle begins with the available pressure. The number of cycles the system undergoes depends on the initial air pressure, tank volume, lower pressure limit and the wind energy

Figure 17 (b) and (c) shows the permeate flow rate and concentration. In the single pressure vessel setup, the permeate water is obtained only during the desalination stage. During the fill period, the membranes remain unproductive and this is indicated by the discontinuity in the permeate flow rate and permeate concentration graphs. A high permeate flux is obtained at high pressure and decreases as the pressure decreases. This can be seen from the graph of permeate flow rate, it decreased gradually as the energy storage tank depressurizes. The permeate concentration on the other hand, was low (about

350 mg/L) at the beginning of the desalination process and increased as the pressure and consequently the permeate flux decreased.

The system was tested by changing different parameters in the model and the results are presented in the following sections.

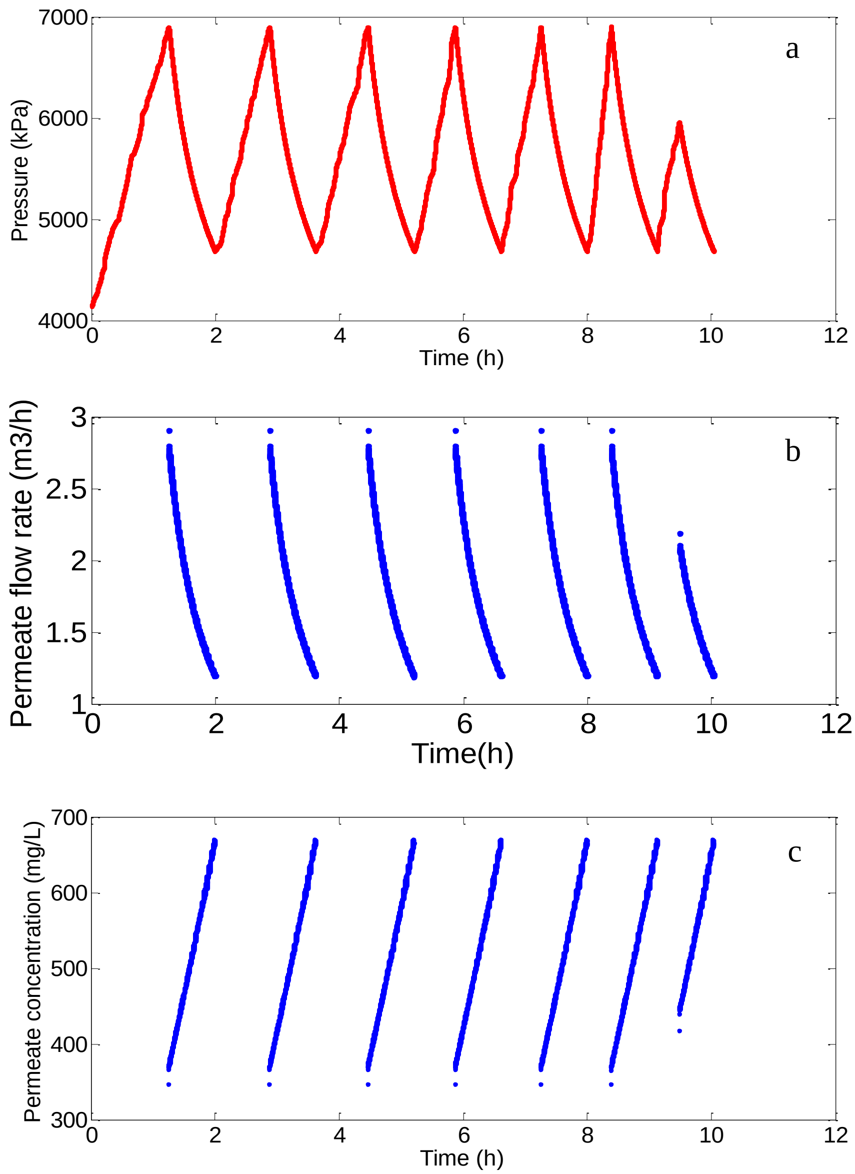


Figure 17. Example of operation of a wind-RO system with integrated single energy storage tank. (a) pressure variation in energy storage tank (b) permeate flow rate (c) permeate salt concentration for wind pattern A.The tank volume was 15 m^3, initial air pressure was 600 psi lower pressure limit was 680 psi and the number of membrnae elements was 5.

Effects of varying the initial air pressure

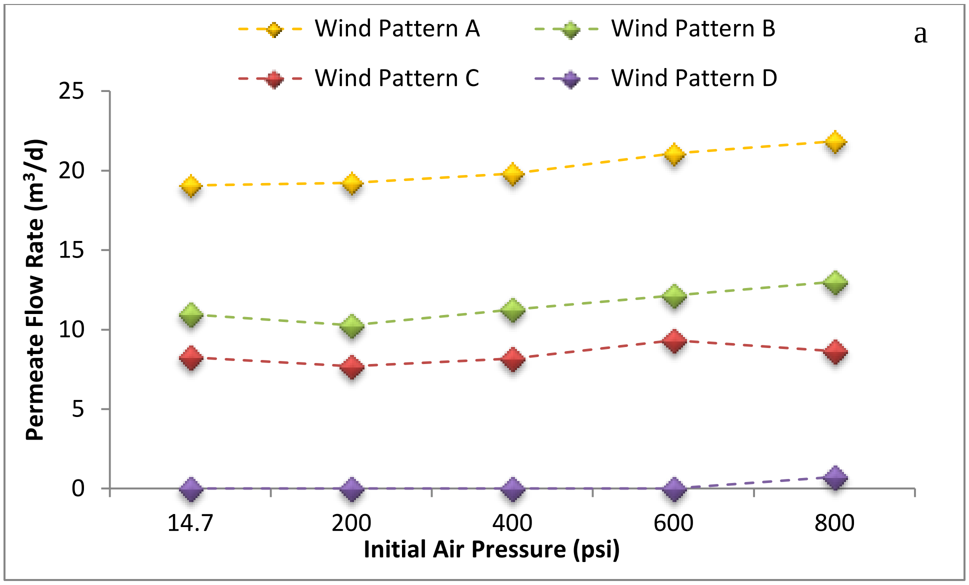
The performance of the full-scale system was tested by varying the initial air pressures. Since the energy storage tank is pressurized to a maximum of 1,000 psi, the initial air pressure can be varied between atmospheric pressure to a pressure close to 1,000 psi. If the initial air pressure is high, the volume of the feed water required to attain a pressure of 1,000 psi is low. However, the pump requires more energy to pump feed water at high initial air pressures. The system was modeled at initial air pressures of 14.7 (atmospheric), 200 psi, 400 psi and 600 psi and 800 psi.

Figure 18 (a) shows the permeate flow rates and permeate concentration obtained for the four wind patterns at different initial air pressures. In this model, an energy storage tank of 15 m^3 was assumed to be connected to 5 membrane elements.

In Figure 18 (a), high permeate flow rates are obtained for wind pattern A while the system failed to deliver water for wind pattern D. The average wind energy produced for wind pattern A is higher than for the other wind patterns. Therefore, it takes less time to fill the energy storage tank when the system is operating under wind pattern A as compared to the other wind patterns. Thus, the number cycles are highest for wind pattern A which results in a high water production rates. Wind patterns B and C produce low power as compared to wind pattern A and therefore the flow rates are lower for these wind patterns.

No water production for wind pattern D indicates that the filling process was not complete in the time for which the wind energy was available. However, a small flow rate of 0.73 m^3 was obtained for the initial pressure of 800 psi. Further, comparing the permeate discharges at the different initial air pressures, it can be observed that there is an increasing trend for the permeate flow rates with increase in the initial air pressures. At high initial air pressure, the pressure drop in the energy storage tank during desalination is slow and in a controlled manner. Thus, high pressures are maintained inside the energy storage tank at high initial air pressure. Therefore the flow rate at 600 psi is about 15 % higher than at 14.7 psi.

From Figure 18 (b), it can be observed that the permeate concentrations for wind patterns A, B and C are nearly equal to 500 mg/L. The concentration for wind pattern D is 0 for all the initial air pressures except at 800 psi because there was no desalination occurring for this configuration. With an exception of initial air pressure of 800 psi, these plots signify that the initial air pressure does not affect the permeate concentration of the water. At 800 psi, lower permeate concentrations of about 450 mg/L were obtained for all the wind patterns. However, the factor that resulted in a lower salt concentration is the lower pressure limit which is explained in the subsequent paragraphs. Thus, the initial air pressure inside the energy storage tank governs the water production rate but has no effect on the permeate salt concentration.



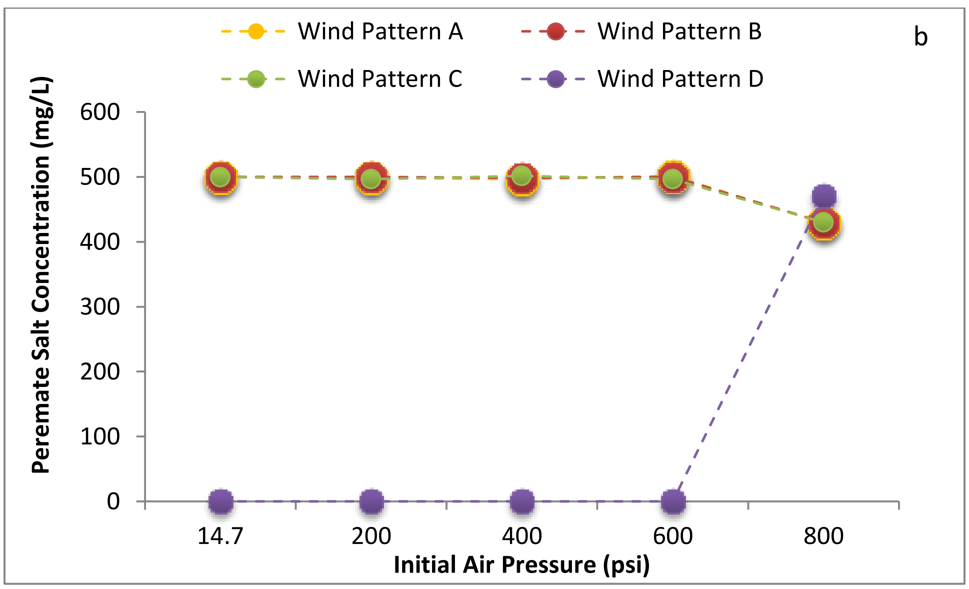


Figure 18. (a) Permeate flow rates and (b) permeate salt concentrations at different initial air pressures for wind patterns A, B, C and D. The tank volume was 15 m^3, number of membrane elements was 5, and lower pressure limit was 680 psi.

Effects of varying the lower pressure limit

An important factor that affects the initial air pressure is the lower pressure limit. The lower pressure limit is the minimum pressure to which the pressure inside the energy storage tank is allowed to decline during the desalination stage. This lower pressure limit affects the water quality obtained from the system. For the modeling experiments, the lower pressure limits were varied from 980 psi to 670 psi. When the lower pressure is set at a high value (close to 1,000 psi), the system will undergo more number of cycles of filling and desalination. When it is set to a lower value, the number of cycles will be also be lower. For example, when the lower pressure limit is set to 980 psi, the pressure drops only by 20 psi during desalination, whereas, if the lower pressure limit is set to 680 psi, then the pressure will drop by 320 psi before the tank starts filling gain. Therefore, a pressure drop of 20 psi will require frequent filling as compared pressure drop of 320 psi.

In Figure 19 (a), as the lower pressure limit is reduced from 900 psi to 670 psi, the permeate flow rates do not change significantly indicating that change in lower pressure limit does not affect the permeate flow rate. This is because at high pressures, the permeate flux is high, but as the pressure declines, the permeate flux is low. Further, at any lower pressure limit, the permeate flow rate does not vary significantly for each of the initial air pressures. Figure 19 shows the effect of lower pressure limit on the permeate flow rates and the permeate concentrations at four different initial air pressures namely 600 psi, 400 psi, 200 psi, and 14.7 psi.

Figure 19 (b) shows the permeate concentrations. As the lower pressure limit is reduced, the permeate concentration increases for all initial air pressures. Also, the concentrations are nearly similar for each initial air pressure at a specific lower pressure limit. Therefore, the permeate concentration is governed by the lower pressure limit and not by initial air pressure. The lowest salt concentrations are obtained at lower pressure limit of 980 psi while at 670 psi, the salt concentration is nearly 500 mg/L. If the lower pressure limit is reduced further, the salt concentrations will be higher than 500 mg/L. Therefore, for each initial air pressure, the lower pressure limit cannot be set to a value below 670 psi.

Figure 19 shows the air and water volumes inside the energy storage tank at 1,000 psi and 670 psi for the 4 initial air pressures. In a tank volume of 15 m^3, when the initial air pressure is 600 psi, at 1,000 psi the volume of feed water is 6 m^3 and at 670 psi the volume is 2.14 m^3. Therefore, 3.86 m^3 of water leaves the pressure vessel. Whereas, when the initial air pressure is at atmospheric pressure (14.7 psi), the energy storage tank requires 14.78 m^3 of water to reach a 1,000 psi. At 670 psi, the feed water volume inside the energy storage tank is 14.69 m^3. In other words, only 0.09 m^3 of water leaves the energy storage tank. Thus, it can be seen that more feed water remains unused inside a tank that has a lower initial air pressure. Clearly, if the pressure limits are set to a higher value, more amount of feed water that is pumped inside the energy storage tank will remain unused. Thus, with increase in lower pressure limit and decrease in the initial air pressure, the “dead” water volume increases.

Table 3. Volume of air and feed water inside energy storage tank at different initial air pressures.

Initial Air Pressure (psi)	Tank Volume (m^3)	*nRT* (kPa. m^3)	At 1000 psi		At 670 psi		Water discharged into membrane units (m^3)
			Air Volume (m^3)	Water Volume (m^3)	Air Volume (m^3)	Water Volume (m^3)	
600	15	62,010	9	6	12.86	2.14	3.86
400	15	41,340	6	9	8.57	6.43	2.57
200	15	20,670	3	12	4.29	10.71	1.29
14.7	15	1,519.25	0.22	14.78	0.32	14.69	0.09

Further investigation on the number of times the system undergoes the fill and desalination cycles was done.

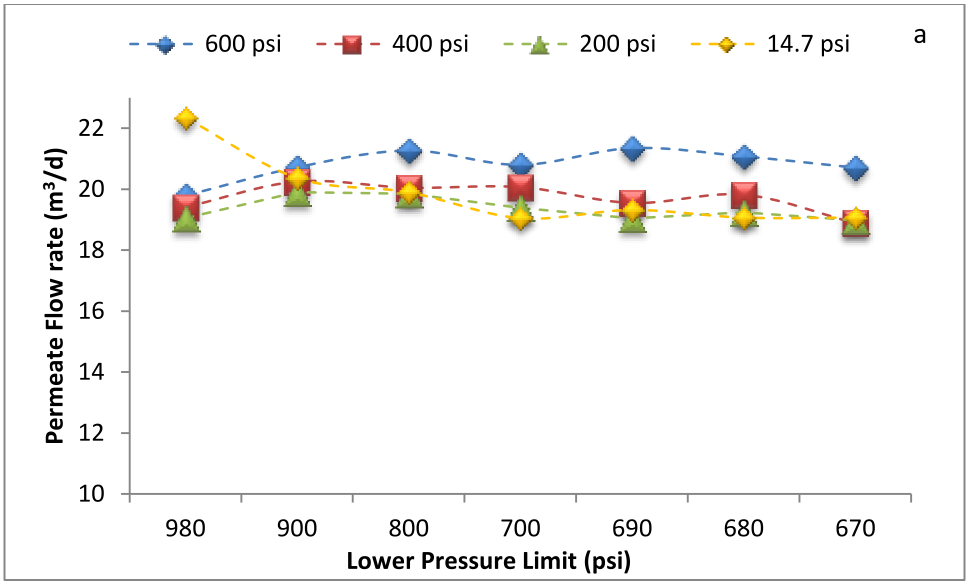
Table 4 shows the cycles at the four initial air pressures and for the lower pressure limits.

Table 4. Total number of cycles (filling and desaliantion) at different lower pressure limits and initial air pressures.

Lower Pressure Limit (psi)	980	900	800	700	690	680	670
Initial Air Pressure (psi)							
600	122	25	12	7	7	7	6
400	179	37	17	10	10	9	9
200	353	73	33	19	18	18	17
14.7	5,519	1036	455	259	251	237	228

It can be seen that when the lower pressure limit is high, the system has to undergo more cycles of filling sand desalination as compared to lower pressure limits. This is clear because when the lower pressure limit is set to a high value, the system will require frequent filling. For instance, the time required for the pressure inside the energy storage tank to drop from 1,000 psi to 980 psi is smaller than the time required to drop from 1,000 psi to 670 psi. Thus, for the duration for which the wind speed is available, the system will undergo more filling and desalination cycles when the pressure limit is set to a higher value as compared to a lower value. Although, the energy storage tank requires frequent filling but it also produces the best quality water. The rejection obtained when the lower pressure limit is set to 980 psi was 98.8 %. The drawback of setting the lower pressure limit to a high value is that only a small portion of the saline water is actually desalinated. A large amount of feed water remains pressurized inside the tank during each cycle.

Further, interestingly, the number cycles increase as the initial air pressure is reduced with a dramatic increase in the number when the initial air pressure is at atmospheric pressure. This can be explained in terms of the resistance offered to the pump due to a high initial air pressure during the fill stage. When feed water is being pumped inside the energy storage tank, the air is getting compressed. When the air is at a higher initial air pressure, the pump has to overcome a higher pressure fill the tank. As the pressure continues to rise, the resistance increases and therefore, it takes longer to fill the tank when the initial air pressure is high. On the other hand, when the air inside the energy storage tank is at atmospheric pressure, water can be pumped inside the tank much faster. This phenomenon also applies for the desalination stage. When the system is at higher initial air pressure, the pressure drop occurs slower and in a more controlled manner. Thus, as the initial air pressure is increased, the filling and desalination take longer time to desalinate. Thus, the number of cycles decrease as the initial air pressure is increased.



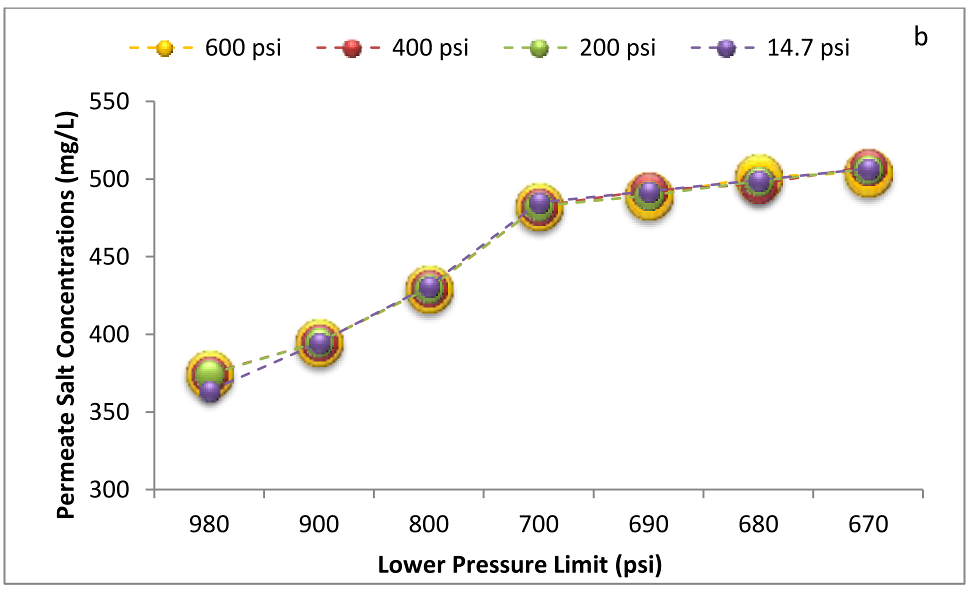


Figure 19. (a) Permeate flow rates and (b) permeate salt concentrations at different lower pressure limits for wind patterns A, B, C and D at different initial air pressures. The tank volume was 15 m^3, number of membrane elements was 5, and initial air pressure was 600 psi.

Effects of varying the tank volume

Figure 20 (a) shows the relation between the permeate flow rates and permeate concentrations as the tank volumes are varied from 100 L to 45,000 L. The initial air pressure was assumed to be 600 psi, the lower pressure limit was 680 psi and the number of membrane elements was 5. It can be seen that, for wind patterns A, B and C, the water production increases with increase in the tank volume with the exception at 30 m^3 and 45 m^3 for wind pattern A. The decrease in the water production at the high volumes is because at the system undergoes only one cycle of filling and desalination. For wind pattern D, at small tank volumes, there is some production of water indicating that this wind pattern is best suitable when a small water quantity is desired and requires smaller tank volume.

Figure 20 (b) shows the permeate salt concentrations for the four wind patterns. The concentrations are nearly same for wind patterns A, B and C and the variation is negligible with the increase in the tank volume. For wind pattern D, the permeate concentration is nearly 600 mg/L for tank volume of 5 m^3. This is primarily due to incomplete filling of the tank. In other words, the wind energy was insufficient to fill the pressure vessel so that a pressure of 1000 psi was attained and desalination occurred at lower pressures. From the figure, it can be concluded that the permeate quality remains unaffected if the volume of the energy storage tank is changed and wind pattern D which has an average wind speed of 1.89 m/s, is suitable for small tank volumes.

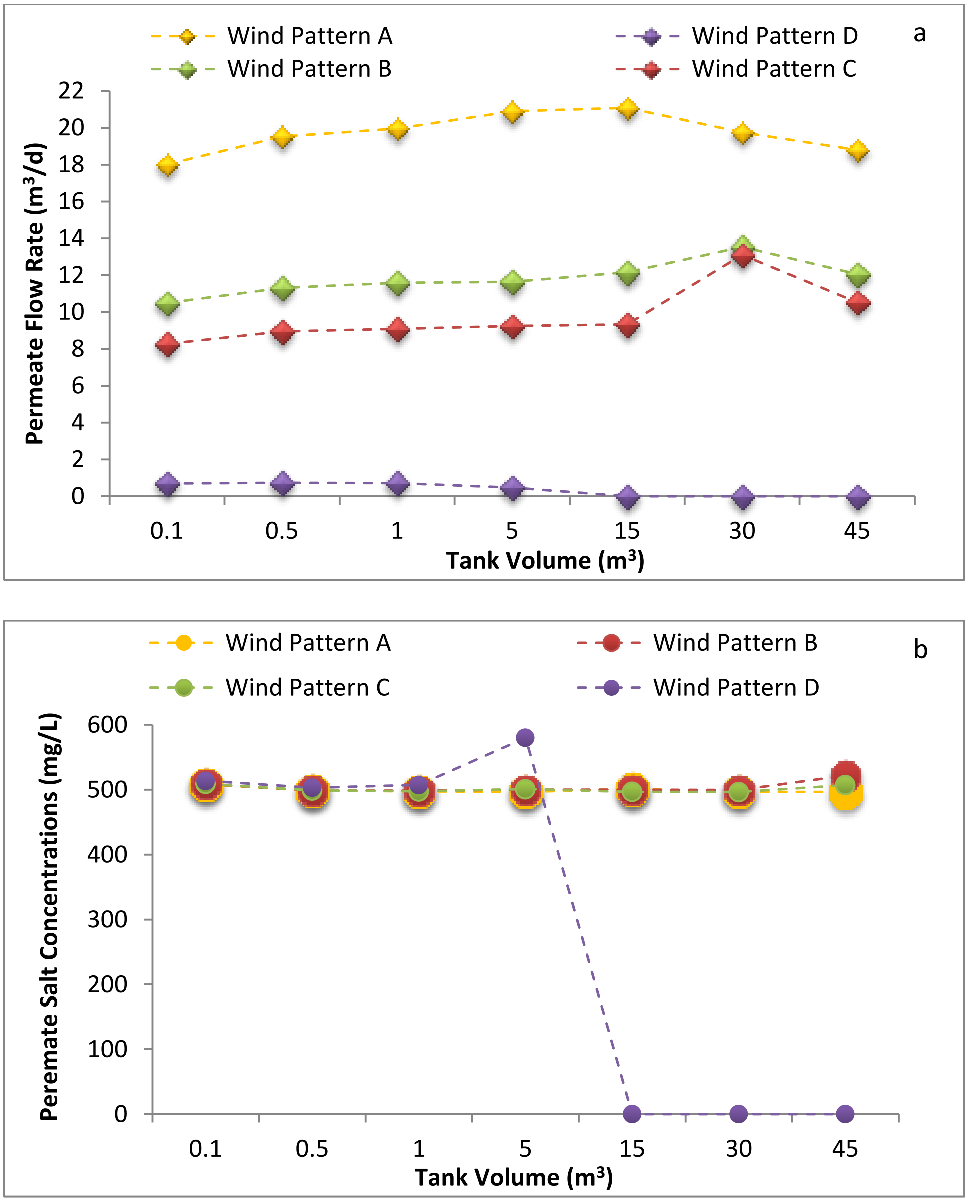


Figure 20. (a) Permeate flow rates and (b) permeate salt concentrations at varying tank volumes for wind patterns A, B, C, and D. The initial air pressure was 600 psi, number of membrane elements wase 5 and lower pressure limit was 680 psi.

Effects of varying the number of RO elements

Figure 21 (a) shows the effect of the number membrane elements on the permeate water production and concentration for each wind pattern. For this simulation an energy storage tank volume of 15 m^3, initial air pressure of 600 psi and lower pressure limit of 680 psi was assumed.

From Figure 21 (b) as the number of membranes elements are increased, the membrane area increases and hence the permeate flow rate also rises. With the increase in number of membrane elements the water production increases. Again, for wind pattern D, with the selected design configuration, there is no water production as the wind energy is not sufficient to fill the energy storage tank.

However, when there are more membrane elements arranged in series, the concentration of the permeate water also increases as the crossflow with high salt concentration from the previous membrane becomes the feed flow for the subsequent membranes. Therefore, if it is desired to add more membrane elements to achieve a higher water production, it is necessary to increase the lower pressure limit so that the permeate water quality is not compensated.

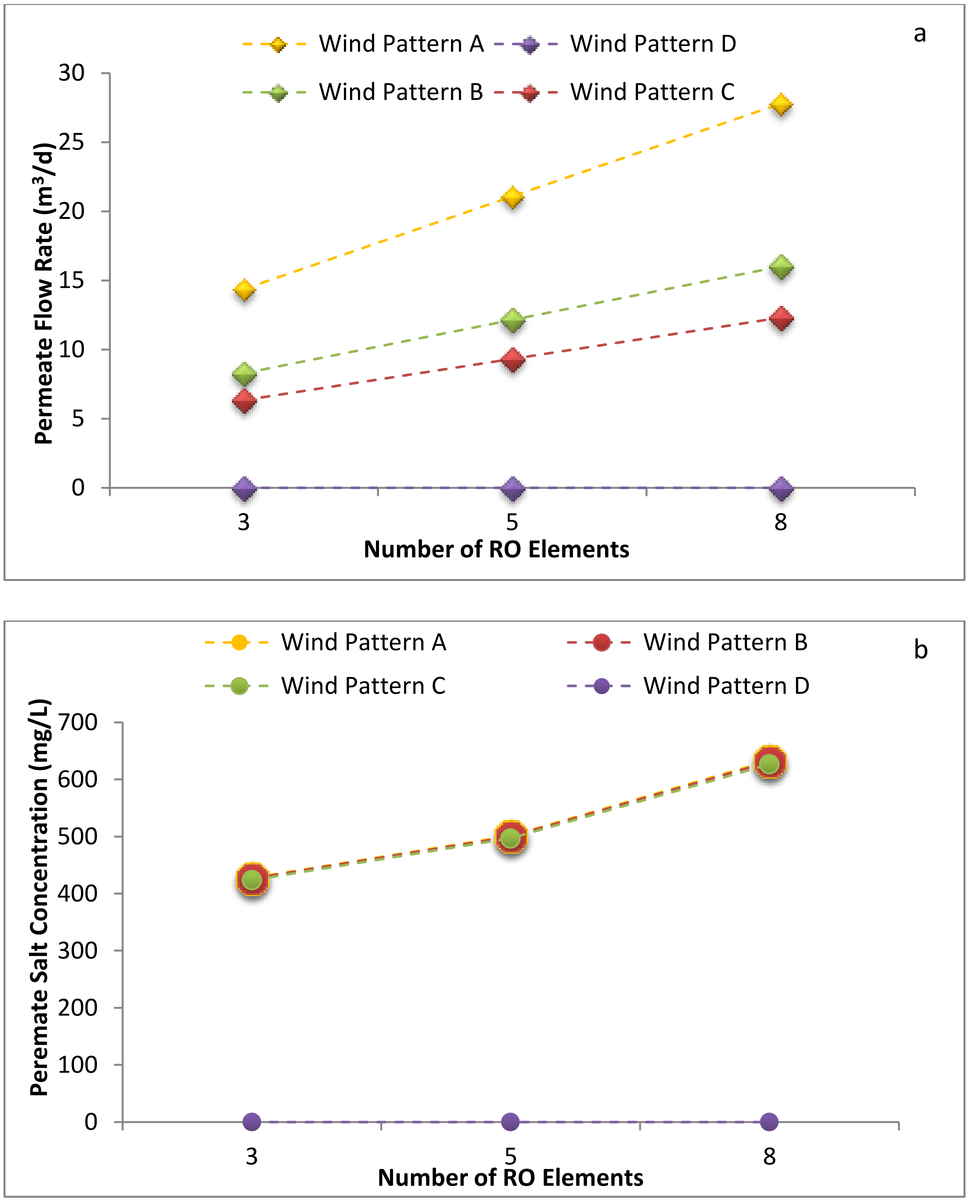


Figure 21 (a) Permeate flow rates and (b) permeate salt concentrations at different number of RO elements for wind patterns A, B, C and D. The tank volume was 15 m^3, initial air pressure was 600 psi, and lower pressure limit was 680 psi.

3.2.3 Three Energy Storage Tanks System

Basic system operation

A system with three energy storage tanks was designed, where each tank is connected to an RO skid and operates in batch mode. This is essentially the connection of three one-tank systems, where the second tank begins operating after the first tank is filled. Such a design allows for continued use of the wind turbine capacity after the first pressure vessel fills and desalination is occurring.

Figure 22 (a) shows the pressure variation inside each energy storage tank. Each energy storage tank had a volume of 15 m^3, initial air pressure was set at 600 psi, the number of membrane elements was 5 and the lower pressure limit was 700 psi (4,823 kPa). As the wind blows, the first energy storage tank begins filling and the pressure builds until it 6,900 kPa. The pump then switches to the second energy storage tank and fills it with feed water until the pressure in this tank reaches 6,900 kPa. At the same time when the second energy storage tank begins to fill, the first tank begins to depressurize as desalination proceeds. Feed water is released from the first tank for desalination and the pressure begins to drop until it reaches 4,823 kPa or 700 psi. Similarly, once the pressure inside the second energy storage tank reaches 6,900 kPa, the third tank begins to fill with feed water and the second one starts depressurizing. This cycle continues until the wind pattern ends.

Figure 22 (b) shows the pressure variation inside the tanks when the lower pressure limit was set to 900 psi. The other parameters were kept same as the first configuration. When the lower pressure limit is increased, the number of cycles also increase. Comparing Figure 22 (a) and (b), it can be seen that the membranes remain idle for shorter periods of time after the initial cycles.

Figure 22 (c) shows the variation in the pressures when the tank volume was changed to 5 m^3. The initial air pressure was 600 psi and the lower pressure limit was 700 psi. As the tank volume is lower, the system takes less time to fill the tanks and hence the system undergoes highest number of cycles compared to the other configurations shown in the figure. Here too, the idle time of the membranes is therefore lowest for this configuration.

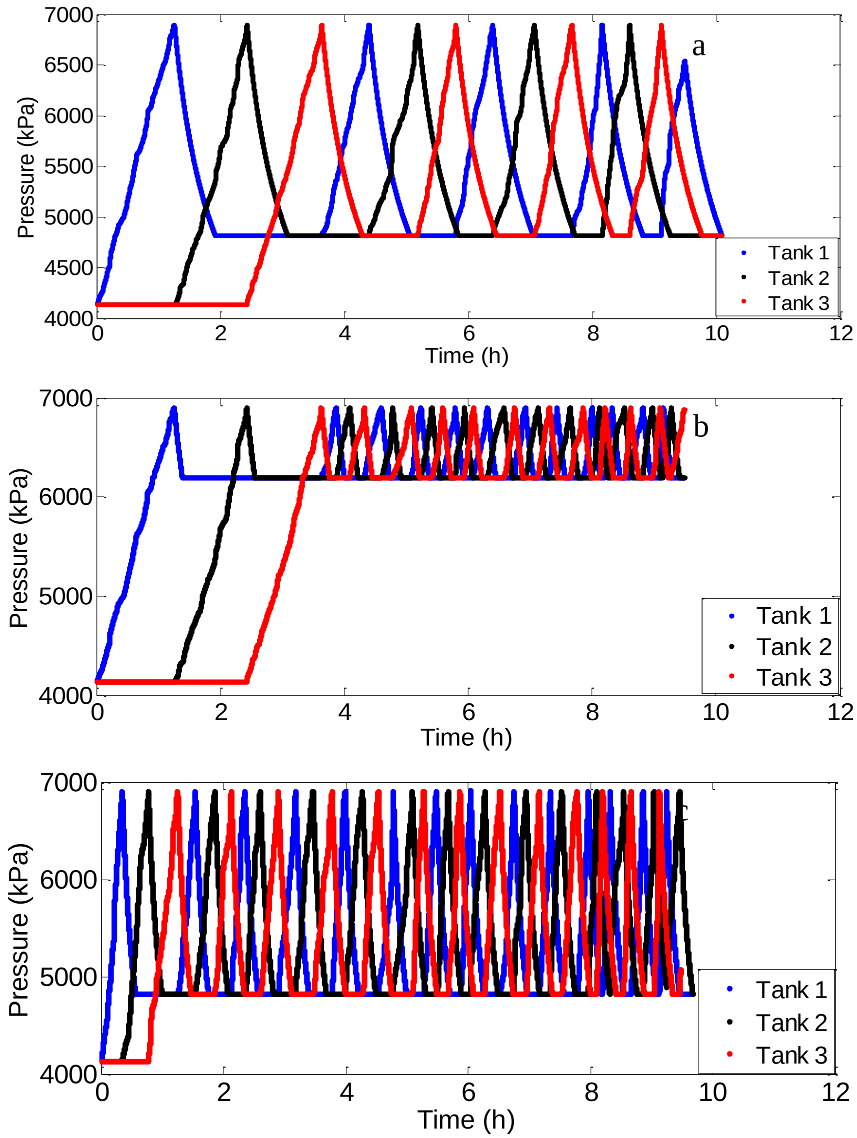


Figure 22. (a) Pressure variation inside energy storage tanks 1, 2 and 3 for wind pattern A.The initial air pressure was 600 psi, tank volume was 15 m^3, number of membrane elements was 5 and lower pressure limit was 700 psi.(b) pressure variation when the lower pressure limit was set to 900 psi. The initial air pressure was 600 psi, tank volume was 15 m^3 and number of membrane elements was 5. (c) pressure variation when the tank volume was set to 5 m^3. The initial air pressure was 600 psi, number of membrane elements was 5 and lower pressure limit was 700 psi.

Effects of varying the initial air pressure and the lower pressure limit

Figure 23 (a) shows the variation in the permeate water flow rates at different initial air pressures. Highest production of water is obtained when the air inside the tanks is at 800 psi; while lowest production of water is obtained at an initial air pressure of 200 psi for wind patterns A, B and C. However, a high water production was also obtained at 14.7 psi (not shown in the figure) for the three energy storage system which is in contrast with the single energy storage tank design, where an increase in the water production was observed as the initial pressures were increased. In the three energy storage tank design, a high permeate flow rate is obtained at 14.7 psi as the process of fill and desalination occurs at a higher frequency. Table 5 compares the number of cycles in the single and three energy storage tank designs at the different initial air pressures. The ratio of number of cycles for each pressure vessel is also shown in the table. It can be seen that, for initial air pressure of 14.7 psi, the number of cycles increased by approximately 2.5 times as compared to an increase by 2 times for initial air pressure of 600 psi. This explains the higher production at 14.7 psi when the number of pressure vessels was increased to three. However, in practice, a system with an initial air pressure of 14.7 is infeasible as there is very little control over the operation and the model developed in this study is unable to predict the infeasibility of such a system. As shown in Table 3, the headspace or the volume of air at 1,000 psi is only 0.22 m^3. In a real system, a spike in the pressure and pump flows will damage the tank and the process would be uncontrollable. These values are therefore disregarded.

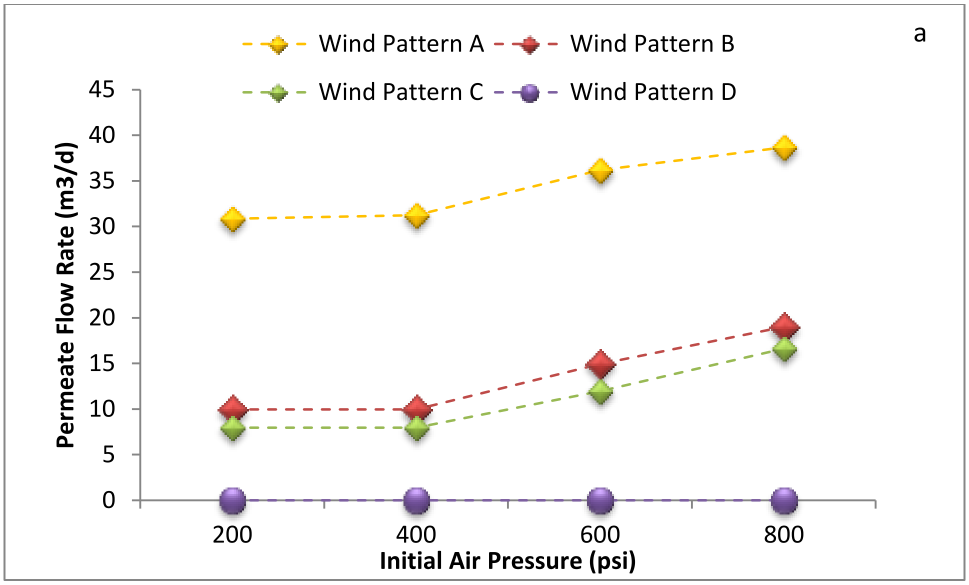
For wind pattern D, there is no permeate flow rate as the wind energy available is insufficient to pressurize the feed water inside the tanks to 1,000 psi. In other words, the filling of the tank could not be completed in 9.5 h.

Table 5. Total number of cycles (filling and desalianation) in single storage tank design and 3 storage tanks design for wind pattern A.

Initial Air Pressure (psi)	**Number of Cycles (Lower pressure limit = 700 psi)**		
	Single Energy Storage Tank	**Three Energy Storage Tanks**	**Ratio**
600	7	13	1.86
400	10	16	1.60
200	19	31	1.63
14.7	259	631	2.44

Figure 23 (b) shows the permeate water concentration for the four wind patterns. As in the single energy storage tanks, the permeate concentration remains fairly constant when the lower pressure limit was maintained constant for first four initial air pressures. At 800 psi, the permeate concentration decreases as the lower pressure limit was set to 805 psi. As explained in chapter 2, the lower pressure limit cannot be set below the initial air pressure, as the tank would run out of feed water. As there was no water production, wind pattern D does not yield any permeate concentration.

The variation in permeate flow rate and salt concentrations with a decrease in the lower pressure limit for wind pattern A is shown in Figure 24 (a) and (b). As the lower pressure limit is decreased, the permeate flow rate increased for each of the initial air pressures. This was unexpected because in the single pressure vessel setup, the lower pressure limit did not have any significant effect on the permeate flow rates. The variation in the three-pressure tank may have happened because of a cumulative membrane downtime during the first filling in the each energy storage tank, which can be seen in Figure 22 (a) and (b) in which the lower pressure limits were set to 700 psi and 900 psi respectively. The permeate salt concentration increased when the lower pressure limit was decreased (Figure 24). The trend of this graph is similar to that obtained for single energy storage tank design.



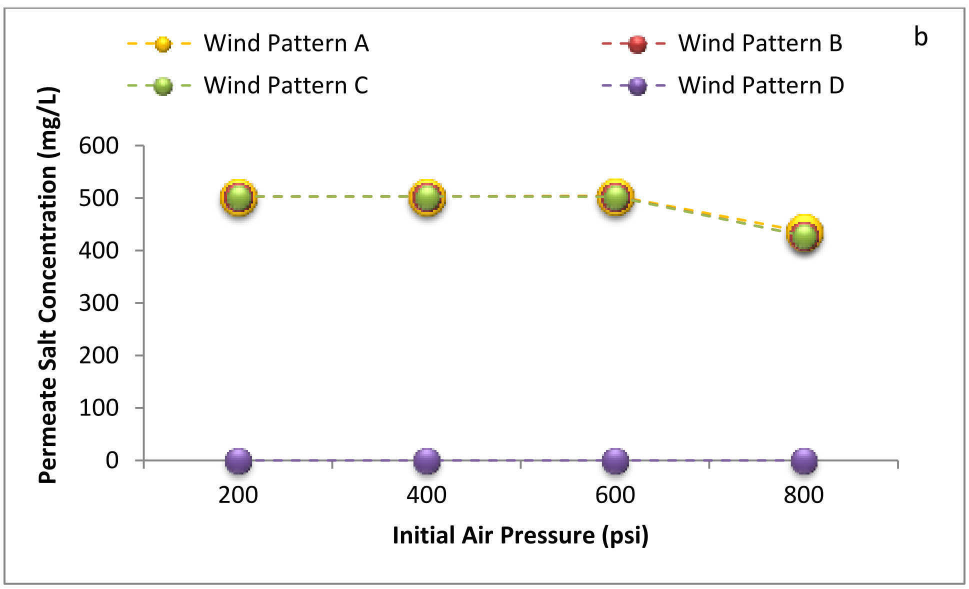


Figure 23. (a) Permeate water flow rates and (b) permeate salt concentrations at different initial air pressures for the three-tank design operating with wind patterns A, B, C and D. The tank volume was 15 m^3, number of membrane elements was 5 and lower pressure limit was 700 psi.

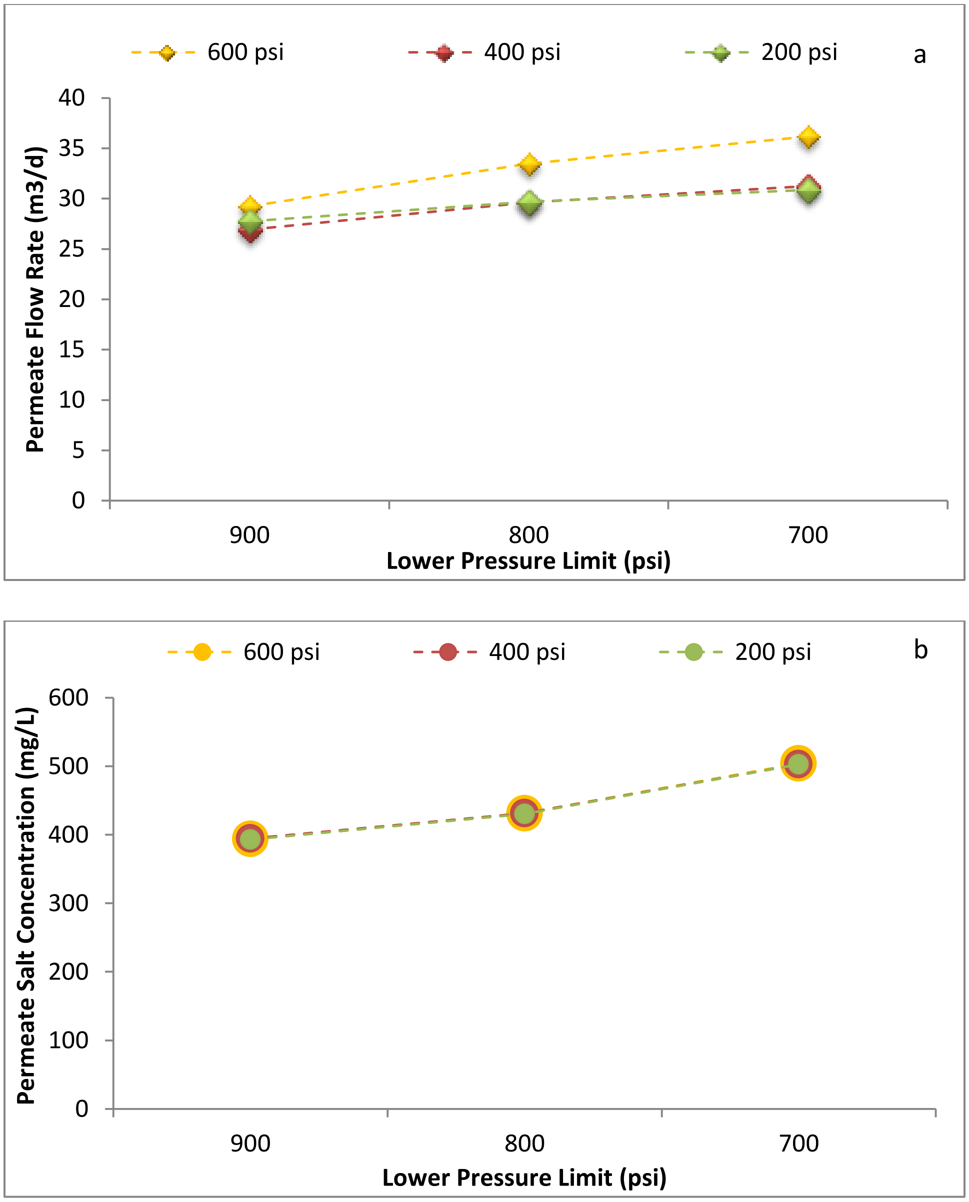
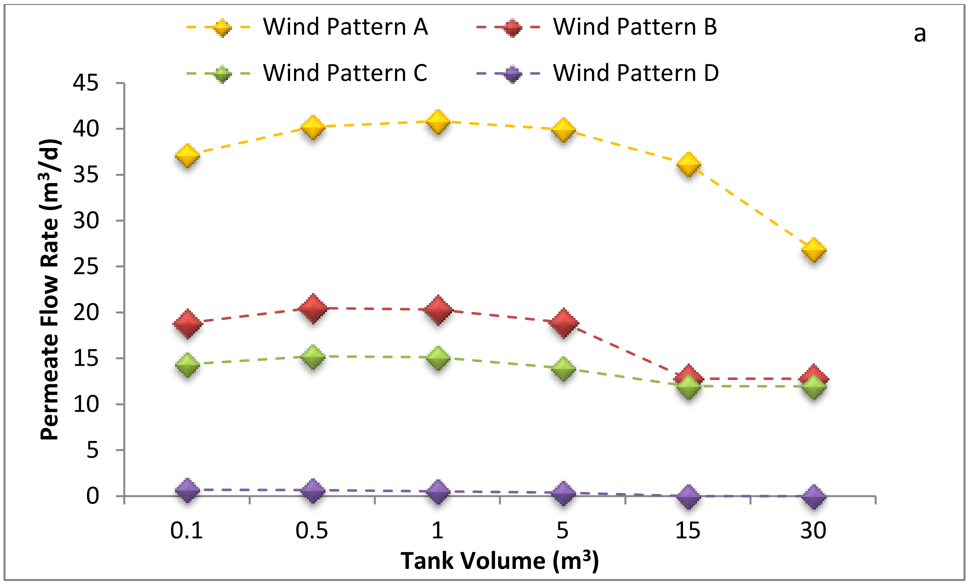


Figure 24. (a) Permeate flow rates and (b) permeate salt concentrations at different lower pressure limits for the three-tank system modeled at the different initial air pressures. The tank volume was 15 m^3, number of membrane elements was 5 and initial air pressure was 600 psi.

Effects of varying the tank volume and the number of RO elements

Figure 25 (a) and (b) shows change in water production and permeate quality by varying the energy storage tank volumes. It can be seen that water production increases as the tank volume increases up to 5 m^3. At higher tank volumes, the water production decreases for wind patterns A, B and C as it takes longer time to fill the tank and less time is dedicated for desalination. As energy available from wind pattern D is low, smaller tank volumes between 0.1 m^3 to 1 m^3 are more suitable. At volumes of 15 m^3 and 30 m^3, the wind energy supplied is insufficient to completely fill the tanks. Increase in the tank volume does not affect he permeate water quality for any of the wind patterns.

Figure 26 (a) and (b) show the performance of the system by changing the number of membrane elements connected to the pressure vessels. As the number of membrane elements are increased, both, the permeate flow rate and water quality, increase.



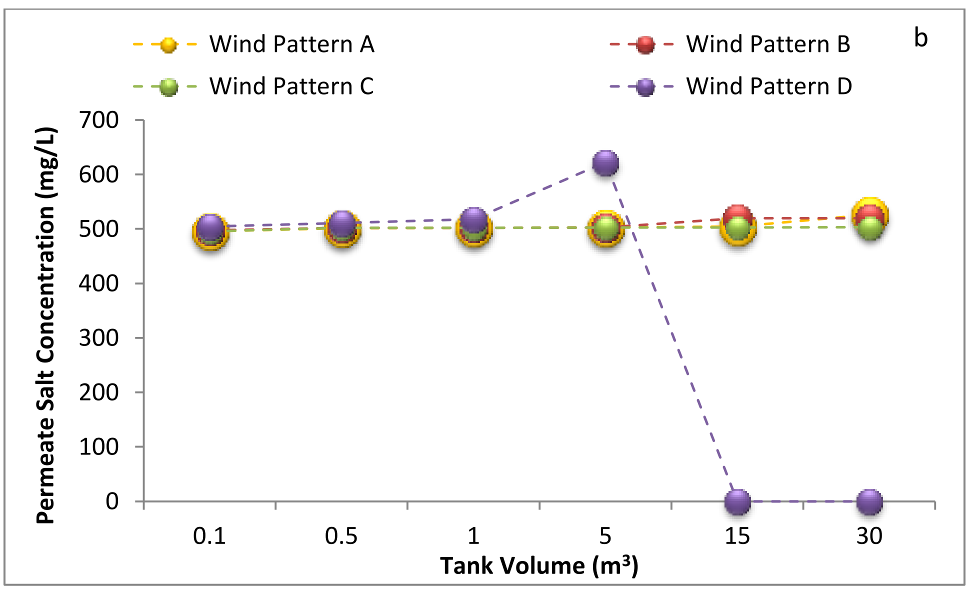


Figure 25. (a) Permeate flow rates and (b) permeate salt concentrations at different tank volumes for a three-tank system modeled with wind patterns A, B, C and D. The initial air pressure was 600 psi, number of membrane elements was 5 and lower pressure limit was 700 psi.

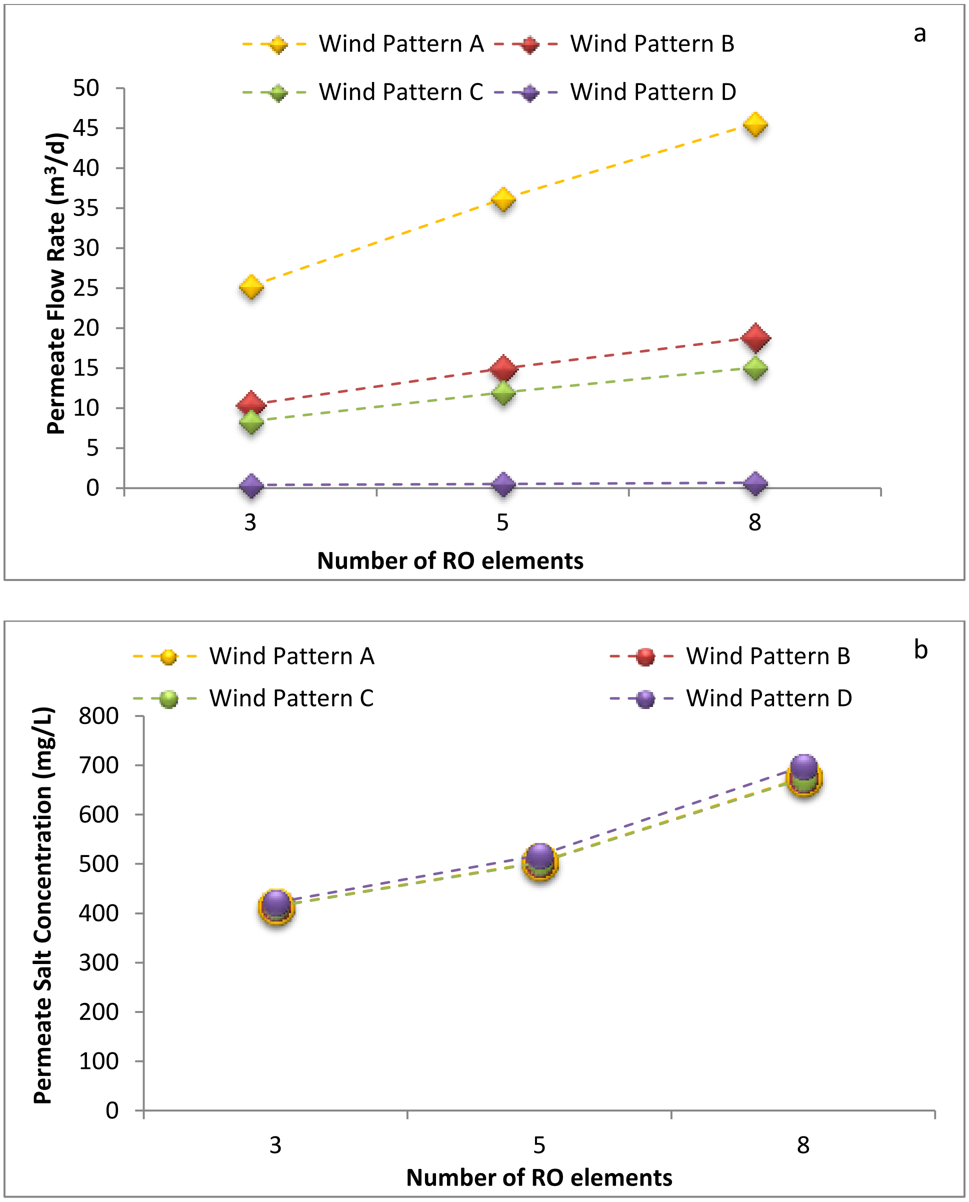


Figure 26. (a) Permeate flow rates (b) permeate salt concentrations for a three-tank system with varying number of RO elements modelled with wind patterns A, B, C and D. The tank volume was 15 m^3, initial air pressure was 600 psi and lower pressure limit was 700 psi.

It is desired to have a high permeate flow rate with permeate concentration below the standard of 500 mg/L from a desalination process to obtain portable water. An initial air pressure of 600 psi and a tank volume of 1,000 L for wind patterns A, B and C and 100L for wind pattern D gives the highest flow rate and the best quality water. By comparing the above results, it was found that, 24 membrane elements provided flow rate of approximately 50 m^3/d. However, the permeate concentration exceeded the standard. This suggests that the lower pressure limit must be increased to decrease the permeate salt concentration. Table 6 provides an optimal design configuration for each wind pattern.

Table 6. Optimized parameters for 3 energy storage tanks design for initial air pressure of 600 psi.

Wind Pattern	**Initial Air Pressure (psi)**	**Lower Pressure Limit (psi)**	**Tank Volume (L)**	**Membrane Elements**	**Water Production (m^3/d)**	**Water Quality (mg/L)**
A	600	900	1,000	24	49.63	501.10
B	600	900	1,000	24	24.38	501.29
C	600	900	1,000	24	19.20	500.86
D	600	850	100	24	0.73	492.38

4. CONCLUSIONS AND FUTURE WORK

4.1 Conclusions

The aim of the project was to model a desalination system that could be driven by wind energy. A conventional RO process requires a high and continuous energy input while wind energy is variable in nature. Hence the primary challenge was to conceptualize a system that could account for the variable wind input. An energy storage mechanism is advantageous in such a system. It was proposed to have a simple form of energy storage, compressed air inside an energy storage tank that can serve as an energy source to drive the RO system and also dampen the fluctuations caused by the stochastic nature of the wind. Further, the operation of such a wind-driven RO system was proposed to be in two-stage batches. In the first stage the pump would fill the energy storage tank with feed water that results in the compression of air inside the vessel. In the second stage the pressurized water is discharged into the membrane elements for desalination to occur. This method of operation decouples the process of desalination from the fluctuating flows and pressure inputs. The energy storage tank causes a dampening effect to these fluctuations. This is important because with such a buffering effect the variability of the wind speeds do not affect the process of desalination. The high and low wind speeds only cause variation in the time required to fill the tanks, but the desalination process is the same no matter the filling time.

In the laboratory, a conventional bench-scale RO unit was modified to integrate an energy storage tank and experiments were performed (by a co-worker, Ying Sun) to simulate batch operation of such a system. Using film theory, a bench-scale model was developed and the results obtained were compared with the actual experimental results. Barring a few outliers, the values of permeate flux and permeate concentration from the model were found to coincide with those from experimental results. This validated the working of the model equations and hence was considered as a base design for development of future models for the conceptual full-scale designs.

Models for conceptual full-scale wind-RO systems were developed based on the bench-scale model. For the basis of comparison, a conventional full-scale system without an energy storage mechanism was modeled. The water production and water quality were low in the conventional system due to irregularities in the wind patterns. These irregularities included extreme fluctuations in wind patterns and periods of very low wind speeds. The variable wind speed would be problematic in a real system because high-pressure spikes could damage the membranes.

A model was developed for a full-scale system with a single energy storage tank. The results from single pressure vessel model suggested that such a system is a feasible option to account for variability in the wind patterns. The main conclusion is that with an energy storage tank the system was able to provide more water than a conventional system. This is most apparent for wind pattern D which had the lowest average wind speed; the conventional system was not able to produce water under wind pattern D (Figure 15 (b)) but a system with a single storage tank of 1 m^3 produced 0.72 m^3/d (Figure 20 (a)). For the other wind patterns the energy storage design was also better; the conventional system produced 18.5, 7.6, 4.7 m^3/d while the energy storage system produced 20, 11.6 and 9.1 m^3/d under wind patterns A, B and C, respectively. Wind patterns B and C have similar average wind speeds of 4.3 and 4.86 m/s, but the variability in wind pattern C is greater (Figure 6). In the conventional system pattern C produced 38% less water than pattern B. In the energy storage system pattern C produced 22% less water. The effect of variability is dampened in the energy storage system so it can produce more water.

The foregoing analysis serves only as an example to highlight the main conclusions, but do not represent data from optimized systems. For optimization to achieve better productivities, several parameters can be manipulated. Detailed analysis was conducted by varying parameters such as wind patterns, initial air pressure inside the tank, lower pressure limit, tank volume and number of membrane elements. A summary of the way in which varying those parameters will affect water production and water quality is presented in Table 7.

Table 7. Summary of results for single energy storage tank design.

Parameter	**Water Production**	**Water Quality**
Increase in initial air pressure	Increases	No Effect
Increase in lower pressure limit	No Effect	Improves
Increase in tank volume	Increases	No Effect
Increase in number of RO elements	Increases	Deteriorates

The initial air pressure was important because it determined the total air mass in the tank and thus the energy buffer capacity; with more air mass the tank took longer to fill and longer to depressurize during desalination. This led to greater water productivity. Water quality, however, was not affected by the initial air pressure because the membrane rejection was time-independent. Rejection was determined only by flux, which in turn was dependent only on instantaneous pressure.

The lower pressure limit was not an important parameter for water production. This was unexpected, because one would assume that lower pressure would result in lower production. However, when the storage tank was allowed to reach a lower pressure, it meant that more time was devoted to desalination and less time was devoted to refilling the tank; there were fewer cycles. With a greater lower pressure limit, there were more cycles, meaning more time devoted to refilling the tank. In the end, it was a wash and water production was essentially the same no matter the lower pressure limit. The permeate water quality, however, did depend on the lower pressure limit. As mentioned just above, instantaneous salt rejection depends on the pressure. When the lower pressure limit is elevated, water is produced at a higher overall pressure and the quality is improved.

The energy storage tank volume was important for water production because a larger volume meant a greater energy storage buffer capacity and greater ability to utilize the wind energy available. The tank volume was not important for water quality because, as stated above, water quality depends only on instantaneous pressure and the pressure values are similar for large or small storage tanks.

An increase in the number of membrane elements led to an increase in water productivity, which would be expected. However, this comes at the expense of water quality. Because the membrane elements are arranged in series, the downstream elements receive higher feed concentrations and thus more salt passes through to the permeate in those elements. This is a similar phenomenon as occurs in conventional RO driven by constant power.

One of the major drawbacks of the single-tank system was the inability to capture wind energy during the desalination step. This drastically affected the overall permeate flux of the system. To resolve this issue and utilize the available energy in a more efficient manner, a system with three energy storage tanks was developed.

The primary advantage of a three-tank system is the maximum utilization of available wind energy. This system was able to achieve the best results among all other systems modeled. Results obtained at an initial air pressure of 14.7 psi were discarded due to their impracticality and uncontrollable nature. The optimum configuration of the system had a high initial air pressure and a high lower pressure limit. The key advantage of this system is that this system. The concepts shown in Table 7 applied to the three-tank system as they did to the one-tank system, except that in case of three- tank system, the water productivity decreased as the lower pressure limit was increased. This may be due to high downtime experienced by the membranes during the initial fill for each storage tank.

The main drawback to the three-tank system would be its greater cost. The capital cost would increase because of the extra tanks and membrane modules. The operation and maintenance costs would also increase because of wear and tear on the valves that must open and close frequently.

4.2 Future Work

The future scope of this project is to integrate and model an energy recovery device (ERD) into the system. Since the pressure loss across the RO membranes is considerably low, the energy from the brine flow can be extracted and utilized. This would result in lower specific energy consumption. The present model simulates a system that loses energy when the concentrate is released; addition of an ERD is expected to increase the efficiency significantly.

Another future aspect of the project is to study the model over a longer range of wind data. Here, 9.5 hours of wind data were used, meaning that startup and shutdown times were significant for some model runs. In reality, startup and shutdown effects would be negligible for locations where the wind is consistent enough to keep the system operating continuously. A model simulating the outputs for prolonged wind data would present a more accurate analysis.

Another future study goal is to develop models with different configurations. One configuration would be to have a system of a multi-stage membrane system. In the current model the system uses 3, 5 and 8 RO elements connected in series. In the future a system with more elements that could be arranged in stages (concentrate from one stage entering a subsequent stage) would be interesting. Another configuration would be to design a system in which and a train of membrane elements is connected to a multiple energy storage tanks. This design would reduce the down time experienced by the membranes during the filling stage. However, there is a possibility of overflowing feed flows when the membranes are connected to multiple pressure vessels. Such system is most likely to work in areas with low wind conditions.

5. REFERENCES

Avlonitis, S.; Kouroumbas, K.; Vlachakis, N. Energy consumption and membrane replacement cost for seawater RO desalination plants. *Desalination* **2003**, *157*, 151-158.

Charcosset, C. A review of membrane processes and renewable energies for desalination. *Desalination* **2009**, *245*, 214-231.

Durland-Rathbone Wind Data. http://sustain.ece.ksu.edu/daq/data.php (accessed February, 2012)

El-Ghonemy, A. Waste energy recovery in seawater reverse osmosis desalination plants. Part 2: Case study. *Renewable and Sustainable Energy Reviews* **2012**, *16*, 4016-4028.

Elimelech, M.; Bhattacharjee, S. A novel approach for modeling concentration polarization in crossflow membrane filtration based on the equivalence of osmotic pressure model and filtration theory. *J. Membr. Sci.* **1998**, *145*, 223-241.

Eltawil, M. A.; Zhengming, Z.; Yuan, L. A review of renewable energy technologies integrated with desalination systems. *Renewable and Sustainable Energy Reviews* **2009**, *13*, 2245-2262.

Engineered Wind Systems. http://engineeredwindsystems.com/ (accessed March, 2012)

Forstmeier, M.; Mannerheim, F.; D'Amato, F.; Shah, M.; Liu, Y.; Baldea, M.; Stella, A. Feasibility study on wind-powered desalination. *Desalination* **2007**, *203*, 463-470.

Garcia-Rodriguez, L. Seawater desalination driven by renewable energies: a review. *Desalination* **2002**, *143*, 103-113.

García-Rodríguez, L. Renewable energy applications in desalination: state of the art. *Solar Energy* **2003**, *75*, 381-393.

Geise, G. M.; Park, H. B.; Sagle, A. C.; Freeman, B. D.; McGrath, J. E. Water permeability and water/salt selectivity tradeoff in polymers for desalination. *J. Membr. Sci.* **2011**, *369*, 130-138.

Gilau, A. M.; Small, M. J. Designing cost-effective seawater reverse osmosis system under optimal energy options. *Renewable Energy* **2008**, *33*, 617-630.

Goosen, M. F. A.; Sablani, S. S.; Al-Maskari, S. S.; Al-Belushi, R. H.; Wilf, M. Effect of feed temperature on permeate flux and mass transfer coefficient in spiral-wound reverse osmosis systems. *Desalination* **2002**, *144*, 367-372.

Greenlee, L. F.; Lawler, D. F.; Freeman, B. D.; Marrot, B.; Moulin, P. Reverse osmosis desalination: Water sources, technology, and today's challenges. *Water Res.* **2009**, *43*, 2317-2348.

Huehmer, R.; Voutchkov, N. Process Design. *AWWA Manual 2007*.

Huisman, P. Filmtec Membranes Technical Manual. *Dow Europe Separation Systems, Germany* **1993**.

Infield, D. Performance analysis of a small wind powered reverse osmosis plant. *Solar Energy* **1997**, *61*, 415-421.

Integrated wind energy/desalination system. NREL/SR-500-39485. 2006

Kalogirou, S. A. Seawater desalination using renewable energy sources. *Progress in energy and combustion science* **2005**, *31*, 242-281.

Kershman, S. A.; Rheinländer, J.; Neumann, T.; Goebel, O. Hybrid wind/PV and conventional power for desalination in Libya—GECOL's facility for medium and small scale research at Ras Ejder. *Desalination* **2005**, *183*, 1-12.

Kim, S.; Hoek, E. Modeling concentration polarization in reverse osmosis processes. *Desalination* **2005**, *186*, 111-128.

Kiranoudis, C.; Voros, N.; Maroulis, Z. Wind energy exploitation for reverse osmosis desalination plants. *Desalination* **1997**, *109*, 195-209.

Kumar, M.; Adham, S. S.; Pearce, W. R. Investigation of seawater reverse osmosis fouling and its relationship to pretreatment type. *Environ. Sci. Technol.* **2006**, *40*, 2037-2044.

Ladner, D. A.; Subramani, A.; Kumar, M.; Adham, S. S.; Clark, M. M. Bench-scale evaluation of seawater desalination by reverse osmosis. *Desalination* **2010**, *250*, 490-499.

Li, D.; Wang, H. Recent developments in reverse osmosis desalination membranes. *Journal of Materials Chemistry* **2010**, *20*, 4551-4566.

Liu, C. C. K.; Jae-Woo, P.; Migita, R.; Gang, Q. Experiments of a prototype wind-driven reverse osmosis desalination system with feedback control. *Desalination* **2002**, *150*, 277-287.

Ma, S.; Song, L.; Ong, S. L.; Ng, W. J. A 2-D streamline upwind Petrov/Galerkin finite element model for concentration polarization in spiral wound reverse osmosis modules. *J. Membr. Sci.* **2004**, *244*, 129-139.

Madireddi, K.; Babcock, R.; Levine, B.; Kim, J.; Stenstrom, M. An unsteady-state model to predict concentration polarization in commercial spiral wound membranes. *J. Membr. Sci.* **1999**, *157*, 13-34.

Mathew, S.; Pandey, K. Modelling the integrated output of wind-driven roto-dynamic pumps. *Renewable Energy* **2003**, *28*, 1143-1155.

Mathioulakis, E.; Belessiotis, V.; Delyannis, E. Desalination by using alternative energy: Review and state-of-the-art. *Desalination* **2007**, *203*, 346-365.

Miranda, M. S.; Infield, D. A wind-powered seawater reverse-osmosis system without batteries. *Desalination* **2003**, *153*, 9-16.

Paul, D. Reformulation of the solution-diffusion theory of reverse osmosis. *J. Membr. Sci.* **2004**, *241*, 371-386.

Sablani, S.; Goosen, M.; Al-Belushi, R.; Wilf, M. Concentration polarization in ultrafiltration and reverse osmosis: a critical review. *Desalination* **2001**, *141*, 269-289.

Sassi, K. M.; Mujtaba, I. M. Simulation and optimization of full scale reverse osmosis desalination plant. *Computer Aided Chemical Engineering* **2010**, *28*, 895-900.

Schock, G.; Miquel, A. Mass transfer and pressure loss in spiral wound modules. *Desalination* **1987**, *64*, 339-352.

Sun Y. Reverse Osmosis Desalination with Integrated Compressed Air Energy Storage : Experimental Proof of Concept. Master's Thesis, Clemson University. 2013.

Taniguchi, M.; Kimura, S. Estimation of transport parameters of RO membranes for seawater desalination. *AIChE J.* **2004**, *46*, 1967-1973.

Thomson, M.; Infield, D. A photovoltaic-powered seawater reverse-osmosis system without batteries. *Desalination* **2003**, *153*, 1-8.

Tzen, E.; Morris, R. Renewable energy sources for desalination. *Solar Energy* **2003**, *75*, 375-379.

Voros, N.; Maroulis, Z.; Marinos-Kouris, D. Salt and water permeability in reverse osmosis membranes. *Desalination* **1996**, *104*, 141-154.

Wijmans, J.; Baker, R. The solution-diffusion model: a review. *J. Membr. Sci.* **1995**, *107*, 1-21.

Wiley, D. E.; Fletcher, D. F. Computational fluid dynamics modelling of flow and permeation for pressure-driven membrane processes. *Desalination* **2002**, *145*, 183-186.

Wind Information Networking and Display. http://wind.ece.ksu.edu/dataselect.php (accessed February, 2012)

Zhou, W.; Song, L.; Guan, T. K. A numerical study on concentration polarization and system performance of spiral wound RO membrane modules. *J. Membr. Sci.* **2006**, *271*, 38-46.

Zydney, A. L. Stagnant film model for concentration polarization in membrane systems. *J. Membr. Sci.* **1997**, *130*, 275-281.

6. APPENDIX

Matlab Code for the Bench-scale System Model

```
%Developed by Pooja K. Mahajan
clear;
close all;
clc;
format long;

%Set filtering to 1 if you want to let the filterFlux02 subroutine go through
%the permeate mass readings and calculate good flux values with linear
%fitting of portions of the mass readings.  This setting also lets the
%noise be plotted.  Set to 2 to suppress the noise.
filtering = 1;

filename = input ('Enter Filename (with single quotes): ');
if strcmp (filename (1:4), '2010')
    pathname = 'C:\Users\Sam\Desktop\Pooma\research\research\matlab\data\' ;
end
if strcmp (filename (1:4), '2011')
    pathname = 'C:\Users\Sam\Desktop\Pooma\research\research\matlab\data\' ;
end
if strcmp (filename (1:4), '2012')
    pathname = 'C:\Users\Sam\Desktop\Pooma\research\research\matlab\data\' ;
end
file = [pathname,filename];

data = load(file);
datasize = size(data);
Atime = data(2:length(data),1);
Acond = data(2:length(data),2);
Apressure = data(2:length(data),3);
ApermMass = data(2:length(data),4);
condPerm = data(2:length(data),5);
temp = data(2:length(data),6);
TankVOL = data (2:length (data),7);
pumpSpeed = data(2:length(data),8);
%Actvolt = data(2:length(data),9);
clear data;
AtimeMIN = (Atime - Atime(1))*24*60;
AtimeHR = (Atime - Atime(1))*24;
clear AtimeMIN;
%filterFlux is a function to spit out the nice-looking and more accurate
%flux data points

if filtering == 1 || filtering == 2;
[AtimeHRf, AfluxLMHf, filtInd] = filterFlux02(AtimeHR, ApermMass);

AcondF = Acond(filtInd);%(310x1)
ApressureF = Apressure(filtInd);
ApermMassF = ApermMass(filtInd);
tempF = temp(filtInd);
condPermF = condPerm(filtInd);
pumpSpeedF = pumpSpeed(filtInd);
TankVOLF = TankVOL (filtInd);
%ActvoltF = Actvolt (filtInd);
```

```
else
end

pressureKpa = ApressureF.* 6.895; % pressure in Kpa
pressurePa = ApressureF.*6895; % applied pressure in Pa

% %The following converts grams per day into liters per meter squared per hour
% %This is based on the active membrane area in the cell being 0.01387 square
% %meters (14.6 by 9.5 cm)
% %if filtering == 1 || filtering == 3;

Aflux = diff(ApermMass)./mean(diff(AtimeHR));
Aflux = [0 Aflux'];
AfluxLMH = Aflux./0.01387./1000./24; %gives the flux in liters per meter
squared per hour
AfluxMH = (AfluxLMHf .* 10^-3); % flux in m/hr
AfluxCMS = AfluxLMHf.* 10^-3.*100./60; %m/s

% CONCENTRATION CALCULATIONS
tdsConcF = AcondF.*613.11 - 664.62;  %converts feed conductivity (mS/cm) to tds
(mg/L) for NaCl. Feed Concentration
tdsConcgl = tdsConcF./1000;%g/l
tdsPermF = condPermF.*512.1 -15.30; % converts permeate conductivity (ms/cm) to
tds (mg/L) for Nacl. Perm Conc
tdsPermgl = tdsPermF./1000;
rej = (1-tdsPermgl./tdsConcgl).*100;
% BULK OSMOSTIC PRESSURE CALCULATION
bulkopF =  tdsConcF.* 8.505 * 10^-2 - 86.61;% bulkop in Kpa and tdsconc in mg/L
bulkopAVGF(1:length(bulkopF)) = mean (bulkopF);
bulkoppsiF = bulkopF.* 1/6.895; % in PSI
bulkoppsiAVGF(1:length(bulkoppsiF)) = mean(bulkoppsiF);

porosity=0.88; % porosity of the feed spacer
%turbfactor=0.5;
h = 0.00173; % m
W=0.095;%m
L=0.146;%m
Aeff = W.*h .*porosity; % m2  %Area = width * h; % for an open channel m2

% PERMEABILITY COEFFIECIENT CALCULATION
A = 0.0632; % permeability coefficient in lmh/psi
Permcoeff = (A .* 10^-3)./6.894; % m/KPa h
PermCoeff = A./6.894; % lmh/Kpa
D = 1.7 *10^-9; %m2/s
%% FLUX PREDICTION

%predicting feed concentration
InitialConc = tdsConcF (1); % mg/L
InitialVOL = 10;%L assumption - valid or not?
Saltmass = InitialConc.*InitialVOL; %mg
FeedConcPredict(1) = Saltmass(1)./InitialVOL; %mg/L
NewfeedVOL(1)=InitialVOL;

membrArea = (14.6*10^-2)*(9.5*10^-2); %m2
FeedConcPredict(1) = tdsConcF (1); %mg/L
fcp(1)=1.3; % initial guess
OsmoticPrKpa(1)=(FeedConcPredict(1).* 8.505*10^-2)- 86.61;%Kpa
FluxPredict(1)=Permcoeff.*(pressureKpa(1)- fcp(1).*OsmoticPrKpa(1));%m/h
FluxPredictLMH(1)=PermCoeff.*(pressureKpa(1)-fcp(1).*OsmoticPrKpa(1)); % lmh
```

```
Permflowrate(1)=FluxPredict(1).*membrArea;%m3/h
PermflowVOL(1)=(Permflowrate(1).*(mean(diff(AtimeHRf))))*1000;%L
TotPermVOL(1)=PermflowVOL(1);

if pumpSpeedF == 7
   actualpumpSpeed = 6.4;
   crossflowrate(1) = -0.226.*ApressureF(1) + 1026; % ml/min pressure in psi
end
if pumpSpeedF == 8.55
            actualpumpSpeed = 8;
            crossflowrate(1) = -0.048.*ApressureF(1) + 1112; % ml/min
end
crossflowrateMS(1) = crossflowrate(1)./1000./1000./60; %m3/s
crossflowrateMH(1) = crossflowrateMS(1).* 3600; %m3/h
crossflowVEL(1) = crossflowrateMS(1)./Aeff; % m/s
Feedflowrate(1)=crossflowrateMH(1);
count(1:length(AfluxLMHf))=0;
d = 1;

TotalEnergy(1)=0;

for n = 2:length(AtimeHRf)

        if pumpSpeed == 0
        actualpumpSpeed = 0;
        crossflowrate(n) = 0; % ml/min
        else
        if pumpSpeed == 5.4
            actualpumpSpeed = 4.8;
            crossflowrate(n) = -0.228.*ApressureF(n) + 774.9; % ml/min
        end
        if pumpSpeedF == 7
            actualpumpSpeed = 6.4;
            crossflowrate(n) = -0.226.*ApressureF(n) + 1026; % ml/min pressure
in psi
        end
        if pumpSpeedF == 8.55
            actualpumpSpeed = 8;
            crossflowrate(n) = -0.048.*ApressureF(n) + 1112; % ml/min
        end
        crossflowrateMS(n) = crossflowrate(n)./1000./1000./60; %m3/s
        crossflowrateMH(n) = crossflowrateMS(n).* 3600; %m3/h
        crossflowVEL(n) = crossflowrateMS(n)./Aeff; % m/s
        %fprintf('\n cross flow velocity for spacer filled channel = %f m/s',
crossflowVEL);
        %wallshearrate(n) = (crossflowVEL(n) .* 3)./(h.*turbfactor); % per
second
        %wallshearrate(n) = (crossflowVEL(n) .* 3)./((h/2)); % per second
(without turbfactor)
        wallshearrate(n) = (crossflowVEL(n) .* 3)./((h/2).*porosity);%persecond
with porosity
        Feedflowrate(n)=crossflowrateMH(n)+Permflowrate(n-
1);%(crossflowrate=ml/min to m3/h).*m3/h = m3/h

        masstransfercoeff(n)=0.807.*((wallshearrate(n).*D.*D)./L).^(1/3);%m/s
        masstransfer(n) = masstransfercoeff(n).*3600; % in m/h
        MassTransfer = mean (masstransfer(n));%m/h

     end
```

```
    NewfeedVOL(n)=NewfeedVOL(n-1)-PermflowVOL(n-1);%%L required for predicting
feed conc
%     FeedConcPredict(n)=Saltmass./NewfeedVOL(n); %mg/L predicting feed conc
    FeedConcPredict(n)=tdsConcF(n);%mg/L using feed conc from the experimental
data
    %FeedConcPredict(n)=35000; %assuming constant feed concentration
    FeedConcPredictgl(n)=FeedConcPredict(n)./1000; % g/L
    fcp(n)=1.5;
    while d ~= 0
        OsmoticPrKpa(n)=(FeedConcPredict(n).* 8.505*10^-2)- 86.61;
        FluxPredict(n)=Permcoeff.*(pressureKpa(n)-fcp(n).*OsmoticPrKpa(n));
%(m/Kpa h * Kpa = m/h)
        FluxPredictLMH(n)=PermCoeff.*(pressureKpa(n)-fcp(n).*OsmoticPrKpa(n));%
lmh
        %fcpNEW(n) = exp(FluxPredict(n)./MassTransfer);
        fcpNEW(n)=exp(FluxPredict(n)./masstransfer(n));
            d = fcp(n)-fcpNEW(n);
            if abs (d) < 0.01
                break;
            end
            if d > 0
                fcp(n)=fcp(n)-0.001;
            end
            if d < 0
                fcp(n)=fcp(n)+0.001;
            end
            count (n) = count(n)+ 1;
    end
    Permflowrate(n) = FluxPredict(n).*membrArea;%m/h*m2 = m3/h
    PermflowVOL(n) = Permflowrate(n).*(mean(diff(AtimeHRf))).*1000;% m3/h * h *
1000 = L
    TotPermVOL(n)= TotPermVOL(n-1)+PermflowVOL(n);
%     Ks(n) = (tdsPermF(n).*AfluxLMHf(n)')./(fcp(n)'.*tdsConcF(n)); %lmh
%     Soluteflux(n) = fcp(n) .* FeedConcPredict(n) .*Ks(n)'; % (mg/L *L/m2 h =
mg/m2 h)
%     PermConcPredict(n) = Soluteflux(n)./(FluxPredict(n).*1000); % mg/m2 h *
h/m = mg/m3 ; mg/(m3 *1000) = mg/L
    Ks(n)=0.0024;%m/d
    Soluteflux(n)=fcp(n).*(FeedConcPredict(n)./10^-3) .*(Ks(n)'./24); % (mg/m3
*m/h = mg/m2 h)
    PermConcPredict(n)=Soluteflux(n)./(FluxPredict(n).*1000); % mg/m2 h * h/m =
mg/m3 ; mg/(m3 *1000) = mg/L
    RejPredict(n)=(1-(PermConcPredict(n)./FeedConcPredict(n))).*100;
    PowerKW(n)=((Feedflowrate(n)./3600).*pressureKpa(n)); %(m3/h to m3/s).*Kpa
= kilowatts

PowerKWactual(n)=((Feedflowrate(n)./3600).*pressureKpa(n))./0.6;%considering
the pump efficiency
    PowerW(n)=PowerKW(n).*1000;%watts
    Energy(n)=(PowerW(n).*(mean(diff(AtimeHRf)).*3600))./1000;%KJ
    TotalEnergy(n)=TotalEnergy(n-1)+Energy(n);

SpecificEnergy(n)=(Energy(n)./(1000.*PermflowVOL(n)));%energy./(density(g/l)*vo
lume(l) = mass in grams)*1000 = KJ/g
end

PermVOL = ApermMassF./1000; %L
permflowrate = AfluxMH.*membrArea;%m3/h
```

```
TotalPower=TotalEnergy./((mean(diff(AtimeHRf))).*3600.*n);%KW
%% FIGURES

figure;
plot (AtimeHRf, pressureKpa, '.' );
%title ('Predicted Pressure Kpa');
xlabel ('Time (h)');
ylabel ('Pressure (Kpa)');
% legend ('Predicted' , 'Actual');

% figure;

figure;
plot (AtimeHRf(2:n), FeedConcPredictgl(2:n), 'k','Linewidth', 2)
hold on;
plot (AtimeHRf(2:n), tdsConcgl(2:n), 'r.');
hold off;
%title ('FEED CONCENTRATION MG/L');
legend ('Predicted' , 'Actual');
ylabel ('Feed Concentration (g/L)');
xlabel ('Time (hrs)');

figure;
plot (AtimeHRf, PermConcPredict, 'k');
hold on;
plot (AtimeHRf, tdsPermF, '.')
hold off;
%title ('PERMEATE CONCENTRATION');
legend ('Predicted' , 'Actual');
ylabel ('Permeate Concentration (mg/L)');
xlabel ('Time (hrs)');

figure;
plot(AtimeHRf(2:n), AfluxLMHf(2:n), '.r')
hold on;
plot (AtimeHRf(2:n), FluxPredictLMH(2:n), 'k')
%title ('FLUX LMH');
legend ( 'actual','predicted');
ylabel ('Flux (lmh)');
xlabel ('Time (hrs)');

figure;
plot(AtimeHRf(2:n), RejPredict(2:n), 'k')
hold on;
plot(AtimeHRf(2:n), rej(2:n), 'b.')
xlabel('Time (hr)')
ylabel('Rejection % ')
```

Matlab Code for the Conventional Full-Scale Wind-RO System

```
%Developed by Pooja K. Mahajan
clear;
clc;
close all;
A = 0.0647; % permeability coefficient in lmh/psi
Permcoeff = (A.*10^-3)./6.894; % m/KPa h
```

```
PermCoeff = A./6.894; % lmh/Kpa
%By calculating from Dow-Filmtech SW30HR parameters. The following values
%will be used in the pump affinity laws to determine instantaneous pressure
%and flowrate as the pump power varies according to the wind power
StdPermflowrateMD=23;%m3/d
StdRecovery=8;%%
StdFlowrateMD = StdPermflowrateMD./(StdRecovery./100); %m3/d (287.5) (Perm flow
rate = 23 m3/d recovery = 8 %, feed flow rate = 23/0.08=287.5 m3/d)
StdFlowrate = StdFlowrateMD./(24*3600);%m3/s (3.32*10^-3)
StdPressureKpa = 5520; %Kpa or 800 psi
StdPowerKW = (StdFlowrate).*StdPressureKpa; %KW (m3/s * KN/m2 = KN m/s
=KW)(18.36)
%Membrane specifications
MembrA=30; %m2 Area of each element
Ks=0.0027875;%m/d
% Ks=0.0033; %m/d
L=1.106;%m
W=MembrA./L;%m
%h=8.636*10^-4;%m 34 mil (0.0008636)%% 1mil =0.0254 mm
h=34.*0.0254.*10^-3;%m
D=1.7*10^-9;%m2/s
membnum=60;%number of membrane elements
%Turbine specifications
rotordia = 15;%m
rotorarea = (pi*rotordia*rotordia)/4;%m2(for 10 m dia= 78.55 m2)
airdensity = 1.1839; %kg/m3
Cpt =0.4; %power coefficient

%Pump specifications
Pumpefficiency = 0.6 ;%

%Initial conditions
TurbinePowerKW(1,1) = 0; %kw
PumpPowerKW(1,1) = Pumpefficiency.*TurbinePowerKW(1,1);
PowerConsumption(1,1)=TurbinePowerKW(1,1);
Pumpflowrate(1,1) = 0; %m3/s
PressureKpa(1,1) = 0;
TotalPermflowVOL(1:membnum,1)=0;
TotalPumpflowVOL(1,1)=0;

TotalSaltMass(1:membnum,1)=0;%mg
TotalFluxPredict(1:membnum,1)=0;%m/h
TotalFluxPredictLMH(1:membnum,1)=0;%lmh
d =1;
 %Fluctuating wind conditions
file='C:\Users\Sam\Desktop\Pooma\research\research\matlab\data\Wind Regime
A.txt';

    winddata=load(file);
    WindSpeed=winddata(1:length(winddata));%m/s
    ctr=1;
    n=length(WindSpeed);
    for i=1:n
        for j=1:30
            windspeed(1,ctr)=WindSpeed(i);
            ctr=ctr+1;
        end
    end
```

```
%windspeed(1:34140)=10;%m/s%% constant wind speed condition
% PowerConsumption(1,1)=0;
TotalEnergy(1,1)=0;
for N = 2:length(windspeed)
    N
    TurbinePower(1,N) =
0.5.*Cpt.*airdensity.*rotorarea.*windspeed(1,N).*windspeed(1,N).*windspeed(1,N)
;%w
    TurbinePowerKW(1,N)= TurbinePower(1,N)/1000;%kw

    PumpPowerKW(1,N) = Pumpefficiency.*TurbinePowerKW(1,N);%kw
    Pumpflowrate(1,N) = StdFlowrate.*(PumpPowerKW(1,N)./StdPowerKW)^(1/3);%m3/s

    Feedflowrate(1,N)=Pumpflowrate(1,N); %m3/s
    TotalFeedflowrate(1,1)=0;
    TotalFeedflowrate(1,N)=TotalFeedflowrate(1,N-
1)+(Feedflowrate(1,N).*3600);%m3/h
    FeedflowrateLS(1,N)=Feedflowrate(1,N).*1000;%L/s
    PumpflowVOL(1,N) = Pumpflowrate(1,N);%m3 in 1 second
    TotalPumpflowVOL(1,N)=TotalPumpflowVOL(1,N-1)+PumpflowVOL(1,N);%m3
    PressureKpa(1,N) = PumpPowerKW(1,N)./(Feedflowrate(1,N));%Kpa
    PressurePSI(1,N)=PressureKpa(1,N)./6.9;%PSI
    if Feedflowrate(1,N) == 0
        PressureKpa(1,N)=0;
    end
        %Desalination
    %For membrane element 1
    CrossflowVEL(1,N)=Feedflowrate(1,N)./(W.*h.*0.88);%m3/s/(m*m) =m/s
    Wallshearrate(1,N)=3.*CrossflowVEL(1,N)./((h/2).*0.88);%m/s/m = /s
    masstransfer(1,N)=0.807.*((Wallshearrate(1,N).*D.*D)./L)^(1/3);%m/s

    FeedConc(1,N)=32000; %mg/L
    FeedConcGL(1,N)=FeedConc(1,N)./1000;%g/L
    OsmoticPrKpa(1,N)=8.505.*10^-2.*FeedConc(1,N)-86.61;%Kpa
    fcp(1,N) = 1;
        while d~=0
          FluxPredict(1,N)=Permcoeff.*(PressureKpa(1,N)-
fcp(1,N).*OsmoticPrKpa(1,N)) ;%(m/Kpa h * Kpa = m/h)
          FluxPredictLMH(1,N)=PermCoeff.*(PressureKpa(1,N)-
fcp(1,N).*OsmoticPrKpa(1,N));% lmh
          if FluxPredict(1,N)<=0
                 FluxPredict(1,N)=0;
                 FluxPredictLMH(1,N)=0;
                 break;
          end
          fcpNEW(1,N)=exp(FluxPredict(1,N)./(masstransfer(1,N).*3600));
          d=fcp(1,N)-fcpNEW(1,N);
          if abs (d)<0.1
            break;
          else
          if d > 0
            fcp(1,N)=fcp(1,N)-0.01;
          end
          if d < 0
            fcp(1,N)=fcp(1,N)+0.01;
          end
          end
        end
    Permflowrate(1,N)=FluxPredict(1,N).*MembrA;%m3/h
```

```
    TotalPermflowrate(1,N)=0;
    TotalPermflowrate(1,N)=TotalPermflowrate(1,N-1)+Permflowrate(1,N);%m3/h
    PermflowrateMS(1,N)=Permflowrate(1,N)./3600;%m3/s
    PermflowrateLS(1,N)=PermflowrateMS(1,N).*1000;%L/s
    PermflowVOL(1,N)=PermflowrateLS(1,N);%L in 1 sec
    TotalPermflowVOL(1,N)=TotalPermflowVOL(1,N-1)+PermflowVOL(1,N);%L
    Crossflowrate(1,N)=(Feedflowrate(1,N))-PermflowrateMS(1,N);%m3/s
    CrossflowrateLS(1,N)=Crossflowrate(1,N).*1000;%L/s
    CrossflowVOL(1,N)=CrossflowrateLS(1,N);%L in s

    %Perm Conc
    SoluteFlux(1,N)= (Ks./24).*fcp(1,N).*(FeedConc(1,N)./10^-3); %mg/h m2
    PermConc(1,N)= SoluteFlux(1,N)./FluxPredict(1,N);%mg/m3
    if FluxPredict(1,N)==0
        PermConc(1,N)=0;
    end
    PermConcMGL(1,N) = PermConc(1,N)./1000;%mg/L
    CrossflowConc(1,N)= ((Feedflowrate(1,N).*(FeedConc(1,N)))-
(PermflowrateMS(1,N).*PermConcMGL(1,N))))./Crossflowrate(1,N);
    if Crossflowrate(1,N) == 0
        CrossflowConc(1,N) = 0;
    end
    Rej(1,N)= (1-(PermConcMGL(1,N)./FeedConc(1,N)))*100;
    if PermConcMGL(1,N) == 0
        Rej(1,N) = 0;
    end
    recovery(1,N)=(Permflowrate(1,N)./3600)./Feedflowrate(1,N);

    SaltMass(1,N)=PermConcMGL(1,N).*PermflowVOL(1,N);%mg/L * L
    TotalSaltMass(1,N)=TotalSaltMass(1,N-1)+SaltMass(1,N);%mg

    % %For membrane element 2 to 5
    %     %Flux Calculations
    for memb=2:membnum
            Feedflowrate(memb,N)=Crossflowrate(memb-1,N);
            if Feedflowrate(memb,N)<0
                Feedflowrate(memb,N)=0;
            end
            TotalFeedflowrate(memb,1)=0;
            TotalFeedflowrate(memb,N)=TotalFeedflowrate(memb,N-
1)+(Feedflowrate(memb,N).*3600);%m3/h

%
masstransfer(memb,N)=0.807.*((3.*(Feedflowrate(memb,N)).*D.*D)./(2.*(h./2).*(h.
/2).*W.*L.*0.88))^(1/3);%m/s
            CrossflowVEL(memb,N)=Feedflowrate(memb,N)./(W.*h.*0.88);%m3/s/(m*m)
=m/s
            Wallshearrate(memb,N)=3.*CrossflowVEL(memb,N)./((h/2).*0.88);%m/s/m
= /s
masstransfer(memb,N)=0.807.*((Wallshearrate(memb,N).*D.*D)./L)^(1/3);%m/s

            PressureKpa(memb,N) = PressureKpa(memb-1,N)-27.6; %assuming a 4 psi
drop
            if PressureKpa(memb-1,N)==0
                PressureKpa(memb,N)=0;
            end
            FeedConc(memb,N) = CrossflowConc(memb-1,N); %mg/L
            FeedConcGL(memb,N)=FeedConc(memb,N)./1000;%g/L
            OsmoticPrKpa(memb,N) = 8.505.*10^-2.*FeedConc(memb,N)-86.61;
```

```
            if FeedConc(memb,N)==0
                OsmoticPrKpa(memb,N)=0;
                PressureKpa(memb,N)=0;
            end
          fcp(memb,N) = 1;
        while d ~= 0
           FluxPredict(memb,N) = Permcoeff.*(PressureKpa(memb,N)-
fcp(memb,N).*OsmoticPrKpa(memb,N)); %(m/Kpa h * Kpa = m/h)
           FluxPredictLMH(memb,N) = PermCoeff.*(PressureKpa(memb,N)-
fcp(memb,N).*OsmoticPrKpa(memb,N));% lmh
              if FluxPredictLMH(memb,N)<=0
                     FluxPredictLMH(memb,N)=0;
                  end
                  if FluxPredict(memb,N)<=0
                     FluxPredict(memb,N)=0;
                      break;
                  end
                    fcpNEW(memb,N) =
exp(FluxPredict(memb,N)./(masstransfer(memb,N).*3600));
                    d = fcp(memb,N) - fcpNEW(memb,N);
                    if abs (d) < 0.1
                    break;
                    else
                    if d > 0
                    fcp(memb,N) = fcp(memb,N)- 0.01;
                    end
                    if d < 0
                    fcp(memb,N) = fcp(memb,N)+ 0.01;
                    end
                    end
        end
        Permflowrate(memb,N) = FluxPredict(memb,N).*MembrA;%m3/h
        TotalPermflowrate(memb,1)=0;
        TotalPermflowrate(memb,N)=TotalPermflowrate(memb,N-
1)+Permflowrate(memb,N);%m3/h
        PermflowrateMS(memb,N)=Permflowrate(memb,N)./3600;%m3/s
        PermflowVOL(memb,N)=PermflowrateMS(memb,N).*1000;%L in 1 sec
        TotalPermflowVOL(memb,N)=TotalPermflowVOL(memb,N-
1)+PermflowVOL(memb,N);%L
        Crossflowrate(memb,N)=(Crossflowrate(memb-1,N))-
PermflowrateMS(memb,N);%m3/s

        %Perm Conc
        SoluteFlux(memb,N)= (Ks./24).*fcp(memb,N).*(FeedConc(memb,N)./10^-3);
%mg/h m2
        PermConc(memb,N)= SoluteFlux(memb,N)./FluxPredict(memb,N);%mg/m3
        if FluxPredict(memb,N) == 0
            PermConc(memb,N) = 0;
        end
        PermConcMGL(memb,N) = PermConc(memb,N)./1000;%mg/L
        CrossflowConc(memb,N)= ((Crossflowrate(memb-1,N).*(FeedConc(memb,N)))-
(PermflowrateMS(memb,N).*PermConcMGL(memb,N)))./Crossflowrate(memb,N);
        if Crossflowrate(memb,N) == 0
            CrossflowConc(memb,N) = 0;
        end
        Rej(memb,N)= (1 - (PermConcMGL(memb,N)./FeedConc(memb,N)))*100;
        if PermConcMGL(memb,N) == 0
            Rej(memb,N) = 0;
        end
```

```
        SaltMass(memb,N)=PermConcMGL(memb,N).*PermflowVOL(memb,N);%mg/L * L
        TotalSaltMass(memb,N)=TotalSaltMass(memb,N-1)+SaltMass(memb,N);%mg
    end
%     PowerConsumption(1,N)= (TurbinePowerKW(1,N)+PowerConsumption(1,N-1));%kw
    PermflowVOLglobal(1,N)=sum(PermflowVOL(1:memb,N));%L
TotalPermflowVOLglobal(1,N)=sum(TotalPermflowVOL(1:memb,N));%L
    TotalPermflowrateglobal(1,N)=sum(TotalPermflowrate(1:memb,N));%m3/h
    Permflowrateglobal(2,N)=sum(Permflowrate(2:memb,N));%m3/h

    SaltMassglobal(1,N)=sum(SaltMass(1:memb,N));%mg

    TotalSaltMassglobal(1,N)=sum(TotalSaltMass(1:memb,N));%mg
    PermConcglobal(1,N)=SaltMassglobal(1,N)./PermflowVOLglobal(1,N);%mg/L
    if PermflowVOLglobal(1,N)==0
       PermConcglobal(1,N)=0;
    end
    Time(1,N)=N;
    TimeHR(1,N) =N./3600;

    TotalTurbinePowerKW(1,N)=sum(TurbinePowerKW(1,1:N));%KW
    TotalTurbinePowerKWavg(1,N)=TotalTurbinePowerKW(1,N)./N;%KW
    SpecificEnergy(1,N)=TotalTurbinePowerKWavg(1,N)./Permflowrateglobal(2,N);
    if PermflowVOLglobal(1,N)==0
       SpecificEnergy(1,N)=0;
    end
end

fprintf('\n Total Permeate Volume=%f L \n',TotalPermflowVOLglobal(1,N));

Permeateflowrate=TotalPermflowVOLglobal(1,N).*24.*60.*60./1000./Time(1,N);
fprintf('\n Design Permeate flow rate=%f m3/d \n',Permeateflowrate);

SpecificE=TotalTurbinePowerKWavg(1,N)./(Permeateflowrate./24);%KWh/m3
fprintf('\n Specific Energy=%f KWh/m3 \n',SpecificE);

PermSaltConc(1,N)=TotalSaltMassglobal(1,N)./TotalPermflowVOLglobal(1,N);%mg/L
Rejection(1,N)=(1-(PermSaltConc(1,N)./FeedConc(1,N))).*100;
fprintf('\n Permeate water quality=%f mg/L \n',PermSaltConc(1,N));
fprintf('\n Rejection=%f % \n ',Rejection(1,N));

Recovery(1:membnum,N)=(TotalPermflowrate(1:membnum,N)./TotalFeedflowrate(1:memb
num,N)).*100;
fprintf('\n Recovery 1=%f %\n',Recovery(1,N));
fprintf('\n Recovery 2=%f %\n',Recovery(2,N));
fprintf('\n Recovery 3=%f %\n',Recovery(3,N));
% fprintf('\n Recovery 4=%f %\n',Recovery(4,N));
% fprintf('\n Recovery 5=%f %\n',Recovery(5,N));

TotalRecovery(1,N)=sum(Recovery(1:memb,N));
fprintf('\n Total Recovery=%f % \n \n',TotalRecovery(1,N));

figure;
plot(TimeHR, windspeed, 'b')
xlabel('Time (h)', 'fontsize',10);
ylabel('Wind Speed (m/s)', 'fontsize',10)
ylim ([0 20])

figure;
```

```
plot(TimeHR(2:N), PressureKpa(1,2:N),'r','LineWidth', 2)
xlabel('Time (Hr)', 'fontsize',12 );
ylabel('Applied Pressure (Kpa)','fontsize',12 )

figure;
plot(TimeHR(2:N), PermflowVOLglobal(1,2:N), 'b', 'LineWidth' , 2)
xlabel('Time (Hr)', 'fontsize',12 );
ylabel('Total Permeate Volume (L)','fontsize',10 )

figure;
plot(TimeHR(2:N), PermConcglobal(1,2:N), 'r', 'LineWidth', 2)
xlabel('Time (Hr)', 'fontsize',12 );
ylabel('Total Salt Concentration (mg/L)','fontsize',12 )
```

Matlab Code for Full-Scale Single Energy Storage System

```
%Developed by Pooja K. Mahajan
clear;
close all;
clc;
format short;
% PERMEABILITY COEFFIECIENT CALCULATION

A = 0.0647; % permeability coefficient in lmh/psi
Permcoeff = (A .* 10^-3)./6.894; % m/KPa h
PermCoeff = A./6.894; % lmh/Kpa

%By calculating from Dow-Filmtech SW30HR parameters. The following values
%will be used in the pump affinity laws to determine instantaneous pressure
%and flowrate as the pump power varies according to the wind power
StdPermflowrateMD=23;%m3/d
StdRecovery=8;%%
StdFlowrateMD = StdPermflowrateMD./(StdRecovery./100); %m3/d (Perm flow rate =
23 m3/d recovery = 8 %, feed flow rate = 23/0.08=287.5 m3/d)
StdFlowrate = StdFlowrateMD./(24*3600);%m3/s
StdCrossflowrateMD=StdFlowrateMD-StdPermflowrateMD;%m3/d
StdPressureKpa = 5520; %Kpa or 800 psi
StdPowerKW = (StdFlowrate).*StdPressureKpa; %KW
%Membrane specifications
MembrA = 30; %m2 Area of each element
Length=40;%inches
L= Length.*2.54./100;%m
W=MembrA./L;%m
h=8.636*10^-4;%m
D=1.7*10^-9;%m2/s

counter=1;

%masstransfer = 0.03983;%m/h
Ks=0.0027875;%m/d
% Ks = 0.0033; %m/d
Recovery=0.08;

%Turbine specifications
rotordia = 15;%m
rotorarea = (pi*rotordia*rotordia)/4;%m2
```

```
airdensity = 1.1839; %kg/m3
Cpt=0.4;

%Pump specifications
Pumpefficiency = 0.6 ;%

%Initial conditions
TurbinePowerKW(1,1) = 0; %kw
PumpPowerKW(1) = Pumpefficiency.*TurbinePowerKW(1);
Pumpflowrate = 0; %m3/s
PressureKpa1(1) = 0;

%*******************************************************
InitialPressure = 600.*6.89; %Kpa
InitialVolume =15000.*10^-3; %m3 volume of pressure vessel
membnum=6;%membranes in each pressure vessel tube=5;excluding The PV tank which
is assigned number=1. Therefore add 1 to the actual number of membranes
Pressurelim = 680*6.89; %Kpa Lower Pressure Limit

TotalTurbinePowerKW(1,1)=TurbinePowerKW(1,1);%kw
TotalEnergy(1,1)=TurbinePowerKW(1,1);
PumpPower(1) = Pumpefficiency.*TurbinePowerKW(1);
nRT = InitialPressure.*InitialVolume; %Kpa*m3
Pumpflowrate(1) = TurbinePowerKW(1)./InitialPressure; %m3/s
PumpflowVOL(1) = Pumpflowrate(1);%m3 in 1 second
AirVOL(1) = InitialVolume - PumpflowVOL(1); %m3
NewWaterVOL(1) = InitialVolume - AirVOL(1); %m3
PressureKpa(1,1) = InitialPressure.*InitialVolume./AirVOL(1); %kpa
%PressureKpa(2,1)=101.32;%atmospheric pressure Kpa
TotalPermflowrate(2:membnum,1)=0;
TotalSaltMass(2:membnum,1)=0;
TotalSaltMassglobal(2,1)=0;
Time(1) = 1;
TotalPermflowrateglobal(2,1) = 0;
TotalPermflowVOL(2:membnum,1)=0;
TotalPermflowVOLglobal(2,1)=0;
Permflowrateglobal(2,1)=0;%%%%%%%%%%%%%%%
TotalPermConcglobal(2,1)=0;
Rejectionglobal(2,1)=0;
cyclenum=1;
cycletime=1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%       FILL     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

;
file='C:\Users\Sam\Desktop\Pooma\research\research\matlab\data\Wind Regime
A.txt';
    winddata=load(file);
    WindSpeed=winddata(1:length(winddata));%m/s
    ctr=1;
    N1=length(WindSpeed);
    for i=1:N1
        for j=1:30
            windspeed(ctr)=WindSpeed(i);
            ctr=ctr+1;
        end
    end
% windspeed(1:86400)=10.1;%m/s
    N=2;
while N<=length(windspeed)
```

```
    TimeVariableFill=N./60;%min

while PressureKpa(1,N-1) <= 6900 && N<= length(windspeed) %1000 psi
    TurbinePower(1,N) =
0.5.*Cpt.*airdensity.*rotorarea.*windspeed(1,N).*windspeed(1,N).*windspeed(1,N)
;%w
    TurbinePowerKW(1,N)= TurbinePower(1,N)/1000;%kw
%     Energy(1,N)=TurbinePowerKW(1,N);%KWsec
%     TotalEnergy(1,N)=TotalEnergy(1,N-1)+Energy(1,N);
    PumpPower(1,N) = Pumpefficiency.* TurbinePowerKW(1,N);%kw
    Pumpflowrate(1,N) = PumpPower(1,N)./PressureKpa(1,N-1);%m3/s
    PumpflowVOL(1,N) = Pumpflowrate(1,N);%m3 in 1 second
    AirVOL(1,N)=AirVOL(1,N-1)-PumpflowVOL(1,N); %m3
    NewWaterVOL(1,N)=InitialVolume-AirVOL(1,N); %m3
    NewWaterVOLL(1,N)=NewWaterVOL(1,N).*1000; %L
    %PressureKpa(1,N)=PressureKpa(1,N-1).*AirVOL(1,N-1)./AirVOL(1,N);%kpa
    PressureKpa(1,N)=nRT./AirVOL(1,N);%Kpa.m3/m3

    TotalPermflowrate(2:membnum,N)=TotalPermflowrate(2:membnum,N-1);
    TotalPermflowrateglobal(2,N)=TotalPermflowrateglobal(2,N-1);
    TotalPermflowVOL(2:membnum,N)=TotalPermflowVOL(2:membnum,N-1);
%     TotalPermflowVOLglobal(2,N)=TotalPermflowVOLglobal(2,N-1);
    TotalSaltMass(2:membnum,N)=TotalSaltMass(2:membnum,N-1);
%     TotalSaltMassglobal(2,N)=TotalSaltMassglobal(2,N-1);
    Permflowrateglobal(2,N)=0;%%%%%%%%%%%%%
   Feedflowrate(2,N)=0;%m3/h
   PermConcglobal(2,N)=0;

    N=N+1;
end

FillTimeMIN(counter)=((N-1)./60)-TimeVariableFill;%min
TimeVariableDesal=N./60;%min

N
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%        DESALINATION
%%%%%%%%%%%%%%%%%%%%%%%%%%%
TurbinePowerKW(1,N)=0;
Pumpflowrate(1,N)=0;
 NewFeedConc(2,N)=32000;%mg/L
NewfeedVOL(2,N) = NewWaterVOL(1,N-1).*1000;% L
%Saltmass = FeedConc(1,N).* NewfeedVOL(1,N);% mg
AirVOL(2,N)= AirVOL(1,N-1).*1000; %L
PressureKpa(2,N) = PressureKpa(1,N-1); %Kpa
PressureKpa(1,N) = PressureKpa(2,N);%Kpa
OsmoticPrKpa(2,N)=(NewFeedConc(2,N).* 8.505.*10^-2)- 86.61;
FluxPredict(2,N) = Permcoeff.*(PressureKpa(2,N)-OsmoticPrKpa(2,N)) ;%(m/Kpa h *
Kpa = m/h)
FluxPredictLMH(2,N)=PermCoeff.*(PressureKpa(2,N)-OsmoticPrKpa(2,N));% lmh
Permflowrate(2,N)=FluxPredict(2,N).*MembrA;%m3/h
Feedflowrate(2,N)=Permflowrate(2,N)./Recovery;%m3/h
PermflowVOL(2,N)=Permflowrate(2,N).*1000./3600;%L in 1 sec
TotalSaltMass(2:membnum,N)=TotalSaltMass(2:membnum,N-1);
% TotalSaltMassglobal(2,N)=TotalSaltMassglobal(2,N-1);
TotalPermflowrate(2:membnum,N)=TotalPermflowrate(2:membnum,N-1);
TotalPermflowrateglobal(2,N)=TotalPermflowrateglobal(2,N-1);
TotalPermflowVOL(2:membnum,N)=TotalPermflowVOL(2:membnum,N-1);
% TotalPermflowVOLglobal(2,N)=TotalPermflowVOLglobal(2,N-1);
Permflowrateglobal(2,N)=0;%%%%%%%%%%%%%%%%%%%%%%
```

```
%
d=1;
N=N+1;
TurbinePowerKW(1,N)=0;
while PressureKpa(2,N-1) >= Pressurelim
    Pumpflowrate(1,N)=0;
    %TurbinePowerKW(1,N)=0;
    if N<= length(windspeed)
    TurbinePower(1,N) =
0.5.*Cpt.*airdensity.*rotorarea.*windspeed(1,N).*windspeed(1,N).*windspeed(1,N)
;%w
    TurbinePowerKW(1,N)= TurbinePower(1,N)/1000;%kw
    else
        TurbinePower(1,N)=0;
        TurbinePowerKW(1,N)=0;
    end
    %membrane element 1
        CrossflowVEL(2,N)=Feedflowrate(2,N-1)./(3600.*W.*h.*0.88);%m3/s/(m*m)
=m/s
    Wallshearrate(2,N)=3.*CrossflowVEL(2,N)./((h/2).*0.88);%m/s/m = /s
    masstransfer(2,N)=0.807.*((Wallshearrate(2,N).*D.*D)./L)^(1/3);%m/s

    NewfeedVOL(2,N)= NewfeedVOL(2,N-1)-((Feedflowrate(2,N-1)./3600).*1000);%L
    NewFeedConc(2,N) = 32000; %mg/L
    OsmoticPrKpa(2,N) = (NewFeedConc(2,N).* 8.505.*10^-2)- 86.61;
    AirVOL(2,N) = InitialVolume.*1000 - NewfeedVOL(2,N);%L
    dP(2,N) = nRT.*1000.*((1./AirVOL(2,N))- (1./AirVOL(2,N-1)));% L Kpa * (1/L)
= Kpa
    PressureKpa(2,N) = dP(2,N) + PressureKpa(2,N-1); % Kpa
    PressureKpa(1,N)=PressureKpa(2,N);%Kpa
    AirVOL(1,N)=AirVOL(2,N)./1000;%L to m3
%
    PressurePSI(2,N)=PressureKpa(2,N)./6.9;

    fcp(2,N) = 1.3;
    while d~=0
          FluxPredict(2,N)=Permcoeff.*(PressureKpa(2,N)-
fcp(2,N).*OsmoticPrKpa(2,N)) ;%(m/Kpa h * Kpa = m/h)
          FluxPredictLMH(2,N)=PermCoeff.*(PressureKpa(2,N)-
fcp(2,N).*OsmoticPrKpa(2,N));% lmh
          if FluxPredict(2,N)<=0
                 FluxPredict(2,N)=0;
                 FluxPredictLMH(2,N)=0;
                 break;
          end
          fcpNEW(2,N)=exp(FluxPredict(2,N)./(masstransfer(2,N).*3600));
          d=fcp(2,N)-fcpNEW(2,N);
          if abs (d)<0.1
            break;
          else
              if d > 0
                fcp(2,N)=fcp(2,N)-0.01;
              end
              if d < 0
                fcp(2,N)=fcp(2,N)+0.01;
              end
          end
    end
    Permflowrate(2,N)=FluxPredict(2,N).*MembrA;%m3/h
```

```
    TotalPermflowrate(2,N)=(TotalPermflowrate(2,N-1)*(N-
1)+Permflowrate(2,N))/N;%m3/h per sec

    PermflowrateMS(2,N)=Permflowrate(2,N)./3600;%m3/s
    PermflowrateLS(2,N)=PermflowrateMS(2,N).*1000;%L/s
    PermflowVOL(2,N)=PermflowrateLS(2,N);%L in 1 sec
    TotalPermflowVOL(2,N)=TotalPermflowVOL(2,N-1)+PermflowVOL(2,N);%L

    Feedflowrate(2,N)=Permflowrate(2,N)./Recovery;%m3/h
    Crossflowrate(2,N)=Feedflowrate(2,N)-Permflowrate(2,N);%m/h
      %Perm Conc
    SoluteFlux(2,N)= (Ks./24).*fcp(2,N).*(NewFeedConc(2,N)./10^-3); %mg/h m2
    PermConc(2,N)= SoluteFlux(2,N)./FluxPredict(2,N);%mg/m3
    if FluxPredict(2,N)==0
        PermConc(2,N)=0;
    end
    PermConcMGL(2,N) = PermConc(2,N)./1000;%mg/L
    CrossflowConc(2,N)= ((Feedflowrate(2,N).*(NewFeedConc(2,N)))-
(Permflowrate(2,N).*PermConcMGL(2,N)))./Crossflowrate(2,N);
    if Crossflowrate(2,N) == 0
        CrossflowConc(2,N) = 0;
    end
    Rej(2,N)= (1-(PermConcMGL(2,N)./NewFeedConc(2,N)))*100;
        if PermConcMGL(2,N) == 0
            Rej(2,N) = 0;
        end

    SaltMass(2,N)=PermConcMGL(2,N).*PermflowVOL(2,N);%mg/L * L
    TotalSaltMass(2,N)=TotalSaltMass(2,N-1)+SaltMass(2,N);%mg

      % %For membrane element 2 to 5
    %     %Flux Calculations
    for memb=3:membnum
            Feedflowrate(memb,N)=Crossflowrate(memb-1,N);%m3/h
             if Feedflowrate(memb,N)<0
                 Feedflowrate(memb,N)=0;
             end
            TotalFeedflowrate(memb,1)=0;

CrossflowVEL(memb,N)=Feedflowrate(memb,N)./(3600.*W.*h.*0.88);%m3/s/(m*m) =m/s
            Wallshearrate(memb,N)=3.*CrossflowVEL(memb,N)./((h/2).*0.88);%m/s/m
= /s

masstransfer(memb,N)=0.807.*((Wallshearrate(memb,N).*D.*D)./L)^(1/3);%m/s

            PressureKpa(memb,N) = PressureKpa(memb-1,N)-27.6; %assuming a 4 psi
drop
            if PressureKpa(memb-1,N)==0
                PressureKpa(memb,N)=0;
            end
            FeedConc(memb,N) = CrossflowConc(memb-1,N); %mg/L
            FeedConcGL(memb,N)=FeedConc(memb,N)./1000;%g/L
            OsmoticPrKpa(memb,N) = 8.505.*10^-2.*FeedConc(memb,N)-86.61;
            if FeedConc(memb,N)==0
                OsmoticPrKpa(memb,N)=0;
                PressureKpa(memb,N)=0;
            end
          fcp(memb,N) = 1;
```

```
        while d ~= 0
           FluxPredict(memb,N) = Permcoeff.*(PressureKpa(memb,N)-
fcp(memb,N).*OsmoticPrKpa(memb,N)); %(m/Kpa h * Kpa = m/h)
           FluxPredictLMH(memb,N) = PermCoeff.*(PressureKpa(memb,N)-
fcp(memb,N).*OsmoticPrKpa(memb,N));% lmh
              if FluxPredictLMH(memb,N)<=0
                     FluxPredictLMH(memb,N)=0;
                  end
                  if FluxPredict(memb,N)<=0
                     FluxPredict(memb,N)=0;
                      break;
                  end
                    fcpNEW(memb,N) =
exp(FluxPredict(memb,N)./(masstransfer(memb,N).*3600));
                    d = fcp(memb,N) - fcpNEW(memb,N);
                    if abs (d) < 0.1
                    break;
                    else
                    if d > 0
                    fcp(memb,N) = fcp(memb,N)- 0.01;
                    end
                    if d < 0
                    fcp(memb,N) = fcp(memb,N)+ 0.01;
                    end
                    end
        end
        Permflowrate(memb,N) = FluxPredict(memb,N).*MembrA;%m3/h
        TotalPermflowrate(memb,N)=(TotalPermflowrate(memb,N-1)*(N-
1)+Permflowrate(memb,N))/N;%m3/h  per sec
        PermflowrateMS(memb,N)=Permflowrate(memb,N)./3600;%m3/s
        PermflowVOL(memb,N)=PermflowrateMS(memb,N).*1000;%L in 1 sec
        TotalPermflowVOL(memb,N)=TotalPermflowVOL(memb,N-
1)+PermflowVOL(memb,N);%L
        Crossflowrate(memb,N)=(Crossflowrate(memb-1,N))-
PermflowrateMS(memb,N);%m3/s

        %Feedflowrate(memb,N)=Permflowrate(memb,N)./Recovery;%m3/h
        Crossflowrate(memb,N)=Feedflowrate(memb,N)-Permflowrate(memb,N);%m/h
        %Perm Conc
        SoluteFlux(memb,N)= (Ks./24).*fcp(memb,N).*(FeedConc(memb,N)./10^-3);
%mg/h m2
        PermConc(memb,N)= SoluteFlux(memb,N)./FluxPredict(memb,N);%mg/m3
        if FluxPredict(memb,N) == 0
            PermConc(memb,N) = 0;
        end
        PermConcMGL(memb,N) = PermConc(memb,N)./1000;%mg/L
        CrossflowConc(memb,N)= ((Crossflowrate(memb-1,N).*(FeedConc(memb,N)))-
(Permflowrate(memb,N).*PermConcMGL(memb,N)))./Crossflowrate(memb,N);
        if Crossflowrate(memb,N) == 0
            CrossflowConc(memb,N) = 0;
        end
        Rej(memb,N)= (1 - (PermConcMGL(memb,N)./FeedConc(memb,N)))*100;
        if PermConcMGL(memb,N) == 0
            Rej(memb,N) = 0;
        end

        SaltMass(memb,N)=PermConcMGL(memb,N).*PermflowVOL(memb,N);%mg/L * L
        TotalSaltMass(memb,N)=TotalSaltMass(memb,N-1)+SaltMass(memb,N);%mg
    end
```

```
    PermflowVOLglobal(2,N)=sum(PermflowVOL(2:memb,N));%L
    TotalPermflowVOLglobal1(2,N)=sum(TotalPermflowVOL(2:memb,N));%L
%TotalPermflowVolglobal is a 1xN matrix of summations of TotalPermflowVol(5xN)


    TotalPermflowrateglobal(2,N)=sum(TotalPermflowrate(2:memb,N));%m3/h
    Permflowrateglobal(2,N)=sum(Permflowrate(2:memb,N));%m3/h

    SaltMassglobal(2,N)=sum(SaltMass(2:memb,N));%mg
    %TotalSaltMassglobal(2,N)=sum(TotalSaltMass(2:memb,N));%mg
    PermConcglobal(2,N)=SaltMassglobal(2,N)./PermflowVOLglobal(2,N);%mg/L
    %
    if PermflowVOLglobal(2,N)==0
       PermConcglobal(2,N)=0;
       %TotalPermConcglobal(2,N)=0;
    end
    %TotalTurbinePowerKW(1,N)=sum(TurbinePowerKW(1,1:N));%KW
    TotalTurbinePowerKW=sum(TurbinePowerKW(1,1:N));%KW
    %TotalTurbinePowerKWavg(1,N)=TotalTurbinePowerKW(1,N)./N;%KW
    TotalTurbinePowerKWavg=TotalTurbinePowerKW./N;%KW
    SpecificEnergy(1,N)=TotalTurbinePowerKWavg./Permflowrateglobal(2,N);
N=N+1
end

N=N-1;

DesalTimeMIN(counter)=(N./60)-TimeVariableDesal;%min
TimeVariableDesalsec = TimeVariableDesal*60;
DesalTimesec=DesalTimeMIN*60;
PressureKpaAvg(counter)=mean(PressureKpa(2,TimeVariableDesalsec:N));
counter=counter+1;
cyclenum=cyclenum+1;
cycletime=N;
%SpecificEnergy(1,N)=TotalEnergy(1,N)./TotalPermflowrateglobal(2,N);%KWh/m3
Time=1:N;
TimeHR=Time./3600;%hr
end
cyclenum=cyclenum-1;
TotalPermflowVOLglobal=sum(PermflowVOLglobal(2,:));
TotalSaltMassglobal=sum(SaltMassglobal(2,:));%mg
TotalPermConcglobal=TotalSaltMassglobal./TotalPermflowVOLglobal;%mg/L

fprintf('\n Permeate water quality=%f mg/L \n',TotalPermConcglobal);
fprintf('\n Total Permeate Volume =%f L \n',TotalPermflowVOLglobal);

Designflowrate=(TotalPermflowVOLglobal./(N.*1000)).*3600.*24;%m3/d
fprintf('\n Permeate flow rate=%f m3/d \n',Designflowrate);

SpecificE=TotalTurbinePowerKWavg./(Designflowrate./24);%KWh/m3
fprintf('\n Specific Energy=%f KWh/m3 \n',SpecificE);

fprintf('\n Number of cycles=%f \n',cyclenum);

Rejglobal=(1-(TotalPermConcglobal./32000)).*100;
fprintf('\n Rejection=%f KWh/m3 \n',Rejglobal);
AvgFillTimeMIN=mean(FillTimeMIN)
AvgDesalTimeMIN=mean(DesalTimeMIN)
AvgPressureKpa=mean(PressureKpaAvg)
```

```
fprintf('\n %f',TotalPermflowVOLglobal);
fprintf('\n %f',Designflowrate);
fprintf('\n %f',TotalPermConcglobal);
fprintf('\n %f',Rejglobal)
fprintf('\n %f',SpecificE);
fprintf('\n %f',cyclenum);


figure;
plot(TimeHR, WindSpeed(1,1:N))
xlabel('Time (h)', 'fontsize',12);
ylabel('Wind Speed (m/s)', 'fontsize',12)

figure
plot(TimeHR, PressureKpa(1,1:N), '.r')
hold on;
plot(TimeHR, PressureKpa(2, 1:N), 'r.')
xlabel('Time (h)', 'fontsize',12);
ylabel('Pressure (kPa)', 'fontsize',12)
ylim([1000 7000]);


figure;
plot(TimeHR, Permflowrateglobal(2,1:N), '.')
xlabel('Time(h)');
ylabel('Permeate flow rate (m3/h)');

figure;
plot(TimeHR, PermConcglobal(2,1:N), '.')
xlabel('Time (h)', 'fontsize',12)
ylabel('Permeate concentration (mg/L)', 'fontsize',12)
```

Matlab Code for Full-Scale, Three Energy Storage Tank System

```
%Developed by Pooja K. Mahajan
clear;
close all;
clc;
format short;
% PERMEABILITY COEFFIECIENT
A = 0.0647; % permeability coefficient in lmh/psi
Permcoeff = (A .* 10^-3)./6.894; % m/KPa h
PermCoeff = A./6.894; % lmh/Kpa

%By calculating from Dow-Filmtech SW30HR parameters. The following values
%will be used in the pump affinity laws to determine instantaneous pressure
%and flowrate as the pump power varies according to the wind power
StdPermflowrateMD=23;%m3/d
StdRecovery=8;%%
StdFlowrateMD=StdPermflowrateMD./(StdRecovery./100); %m3/d (Perm flow rate = 23
m3/d recovery = 8 %, feed flow rate = 23/0.08=287.5 m3/d)
StdFlowrate=StdFlowrateMD./(24*3600);%m3/s
StdCrossflowrateMD=StdFlowrateMD-StdPermflowrateMD;%m3/d
StdPressureKpa=5520; %Kpa or 800 psi
StdPowerKW=(StdFlowrate).*StdPressureKpa; %KW
%Membrane specifications
```

```
MembrA = 30; %m2 Area of each element
Length=40;%inches
L= Length.*2.54./100;%m
W=MembrA./L;%m
h=8.636*10^-4;%m
D=1.7*10^-9;%m2/s
%masstransfer = 0.03983;%m/h
Ks = 0.0027875; %m/d
recoveryA=0.08;
recoveryB=0.08;
recoveryC=0.08;

%Turbine specifications
rotordia = 15;%m
rotorarea = (pi*rotordia*rotordia)/4;%m2
airdensity = 1.1839; %kg/m3
Cpt=0.4;
counter=0;
%Pump specifications
Pumpefficiency = 0.6 ;%

TotalPermeateVOLABC=0;
TotalPermeateVOLA=0;
TotalPermeateVOLB=0;
TotalPermeateVOLC=0;
TotalSaltMassABC=0;
TotalSaltMassA=0;
TotalSaltMassB=0;
TotalSaltMassC=0;
TotalSaltMassglobalA(1)=0;
TotalSaltMassglobalB(1)=0;
TotalSaltMassglobalC(1)=0;
TotalPermVOLglobalA(1)=0;
TotalPermVOLglobalB(1)=0;
TotalPermVOLglobalC(1)=0;

cyclenum=0;
%%  ************** Initial FILL/Desal  steps ****************
InitialPressure=600.*6.89; %Kpa
InitialVolume=15; %m3 volume of pressure vessel
membnum=6;%number of membrane elements. Add 1 to the total number for the way
the matrices are numbered
Pressurelim=900.*6.89;%kpa

% file = 'C:\Users\Student\Desktop\research\matlab\data\wind speeds.txt' ;
file='C:\Users\Sam\Desktop\Pooma\research\research\matlab\data\Wind Regime
A.txt';
    winddata=load(file);
    WindSpeed=winddata(1:length(winddata));%miles/s
    ctr=1;
    N1=length(WindSpeed);
    for i=1:N1
        for j=1:30
            windspeed(ctr)=WindSpeed(i);
            ctr=ctr+1;
        end
    end
%Initial conditions for A ---> First Fill Conditions
TurbinePowerKW(1,1)=0; %kw
```

```
PumpPowerKW(1,1)=Pumpefficiency.*TurbinePowerKW(1);

nRT=InitialPressure.*InitialVolume; %Kpa*m3
PumpflowrateA(1,1)=TurbinePowerKW(1)./InitialPressure; %m3/s
PumpflowVOLA(1,1)=PumpflowrateA(1);%m3 in 1 second
NewAirVOLA(1,1)=InitialVolume-PumpflowVOLA(1,1); %m3
NewWaterVOLA(1,1)=InitialVolume-NewAirVOLA(1,1); %m3
PressurePVA(1,1)=InitialPressure.*InitialVolume./NewAirVOLA(1,1); %kpa
PressurePVB(1,1)=InitialPressure;%kpa
PressurePVC(1,1)=InitialPressure;%Kpa

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
n = 2;
%% # 1 Fill A
while PressurePVA(1,n-1) <= 6900 && n<=length(windspeed) %1000 psi

TurbinePower(1,n)=0.5.*Cpt.*airdensity.*rotorarea.*windspeed(1,n).*windspeed(1,
n).*windspeed(1,n);%w
      TurbinePowerKW(1,n)=TurbinePower(1,n)/1000;%kw
      PumpPower(1,n)=Pumpefficiency.* TurbinePowerKW(1,n);%kw
      PumpflowrateA(1,n)=PumpPower(1,n)./PressurePVA(1,n-1);%m3/s
      PumpflowVOLA(1,n)=PumpflowrateA(1,n);%m3 in 1 second
      NewAirVOLA(1,n)=NewAirVOLA(1,n-1)-PumpflowVOLA(1,n); %m3
      NewWaterVOLA(1,n)=InitialVolume-NewAirVOLA(1,n); %m3
      NewWaterVOLLA(1,n)=NewWaterVOLA(1,n).*1000; %L
      %PressurePVA(1,n)=PressurePVA(1,n-1).*NewAirVOLA(1,n-
1)./NewAirVOLA(1,n);%kpa
      PressurePVA(1,n)=nRT./NewAirVOLA(1,n);%Kpa.m3/m3=Kpa
      PressurePVB(1,n)=InitialPressure;%Kpa (since only PVA is filling and PVB
and PVC have initial gas pressure in them)
      PressurePVC(1,n)=InitialPressure;%Kpa (since only PVA is filling and PVB
and PVC have initial gas pressure in them)
      FillTimeA(1,n)=n;
      FillTimeHRA(1,n)=FillTimeA(1,n)./3600;
%

       TotalPermVOLglobalA(n)=0;%
TotalPermVOLglobalB(n)=0;%L
      TotalPermVOLglobalC(n)=0;%L
      TfA = n;
      n=n+1;
end

n=TfA+1;%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% # 2 Desal A and Fill B
%% at TfA, PV A is still filling. Therefore, neither B begins filling, nor A
begins desalinating

%Initial conditions for B ---> First Fill conditions
% TurbinePowerKW(1,TfA)=0;
PumpPowerKW(1,TfA)=Pumpefficiency.*TurbinePowerKW(1,TfA);
PumpflowrateB(1,TfA)=TurbinePowerKW(1,TfA)./InitialPressure; %m3/s
PumpflowVOLB(1,TfA)=PumpflowrateB(1,TfA);%m3 in 1 second
NewAirVOLB(1,TfA)=InitialVolume-PumpflowVOLB(1,TfA); %m3
NewWaterVOLB(1,TfA)=InitialVolume-NewAirVOLB(1,TfA); %m3
PressurePVB(1,TfA)=InitialPressure.*InitialVolume./NewAirVOLB(1,TfA); %kpa
```

```
%%Initial conditions for Desal A ----> First Desal conditions
AirVOLA(2,TfA)=NewAirVOLA(1,TfA).*1000;
PressurePVA(2,TfA)=PressurePVA(1,TfA);
NewFeedConcA(2,TfA)=32000; %mg/L
OsmoticPrKpaA(2,TfA)=(NewFeedConcA(2,TfA).* 8.505.*10^-2)- 86.61;%Kpa
FluxPredictA(2,TfA)=Permcoeff.*(PressurePVA(2,TfA)-OsmoticPrKpaA(2,TfA));%m/h
PermflowrateA(2,TfA)=FluxPredictA(2,TfA).*MembrA;%m3/h
FeedflowrateA(2,TfA)=PermflowrateA(2,TfA)./recoveryA;%m3/h
FeedflowVOLA(2,TfA)=FeedflowrateA(2,TfA)./3600.*1000;%L
NewfeedVOLA(2,TfA)=NewWaterVOLA(1,TfA).*1000-FeedflowVOLA(2,TfA);%L
PermflowVOLA(2,TfA)=PermflowrateA(2,TfA).*1000./3600;%L
TotalPermVOLA(2,TfA)=PermflowVOLA(2,TfA);%L
TotalSaltMassA(2,TfA)=0;%
TotalPermVOLA(3:membnum,TfA)=0;%

PressurePVC(1,TfA)=InitialPressure;%kpa
d=1;
TfB=TfA;
TdA=TfA;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
while (PressurePVB(1,n-1) <= 6900 || PressurePVA(2,n-1)>Pressurelim )  %
    if PressurePVA(2,n-1)< Pressurelim && n >=length(windspeed)
        break;
    end
    %%%%%       Fill B   %%%%%%%%%%%%%%%%
    if  PressurePVB(1,n-1) <= 6900 && n<=length(windspeed)

TurbinePower(1,n)=0.5.*Cpt.*airdensity.*rotorarea.*windspeed(1,n).*windspeed(1,
n).*windspeed(1,n);%w
        TurbinePowerKW(1,n)=TurbinePower(1,n)/1000;%kw
        PumpPower(1,n)=Pumpefficiency.* TurbinePowerKW(1,n);%kw
        PumpflowrateB(1,n)=PumpPower(1,n)./PressurePVB(1,n-1);%m3/s
        PumpflowVOLB(1,n)=PumpflowrateB(1,n);%m3 in 1 second
        NewAirVOLB(1,n)=NewAirVOLB(1,n-1)-PumpflowVOLB(1,n); %m3
        NewWaterVOLB(1,n)=InitialVolume-NewAirVOLB(1,n); %m3
        NewWaterVOLLB(1,n)=NewWaterVOLB(1,n).*1000; %L
%          PressurePVB(1,n)=PressurePVB(1,n-1).*NewAirVOLB(1,n-
1)./NewAirVOLB(1,n);%kpa
        PressurePVB(1,n)=nRT./NewAirVOLB(1,n);%Kpa.m3/m3=Kpa
        FillTimeB(1,n)=n;
        FillTimeHRB(1,n)=FillTimeB(1,n)./3600;
        TfB = n;
    else
        PressurePVB(1,n)=PressurePVB(1,n-1);
    end


    %%%%%%%%%%%%%        Desal A         %%%%%%%%%%%%%%%%%%%%%%%%%
    if  PressurePVA(2,n-1)> Pressurelim

        CrossflowVELA(2,n)=FeedflowrateA(2,n-1)./(3600.*W.*h.*0.88);%m3/s/(m*m)
=m/s
        WallshearrateA(2,n)=3.*CrossflowVELA(2,n)./((h/2).*0.88);%m/s/m = /s
        masstransferA(2,n)=0.807.*((WallshearrateA(2,n).*D.*D)./L)^(1/3);%m/s

        %masstransferA(2,n)=0.807.*((3.*(FeedflowrateA(2,n-
1)./3600).*D.*D)./(2.*(h./2).*(h./2).*W.*L))^(1/3);%m/s
        NewfeedVOLA(2,n)= NewfeedVOLA(2,n-1)-FeedflowVOLA(2,n-1);%L
```

```
        NewFeedConcA(2,n)=32000; %mg/L
        OsmoticPrKpaA(2,n) = (NewFeedConcA(2,n).* 8.505.*10^-2)- 86.61;
        AirVOLA(2,n) = (InitialVolume.*1000) - NewfeedVOLA(2,n);%L
        dPA(2,n) = nRT.*1000.*((1./AirVOLA(2,n))- (1./AirVOLA(2,n-1)));% L Kpa
* (1/L) = Kpa
        PressurePVA(2,n) = dPA(2,n) + PressurePVA(2,n-1); % Kpa
        PressurePSIA(2,n)=PressurePVA(2,n)./6.9;
        fcpA(2,n) = 1.3;
       while d ~= 0
            FluxPredictA(2,n) = Permcoeff.*(PressurePVA(2,n) -
fcpA(2,n).*OsmoticPrKpaA(2,n)); %(m/Kpa h * Kpa = m/h)
            FluxPredictLMHA(2,n) = PermCoeff.*(PressurePVA(2,n) -
fcpA(2,n).*OsmoticPrKpaA(2,n));% lmh
            fcpNEWA(2,n) = exp(FluxPredictA(2,n)./(masstransferA(2,n).*3600));
                d = fcpA(2,n) - fcpNEWA(2,n);
                 if abs (d) < 0.1
                    break;
                 else
                if d > 0
                    fcpA(2,n) = fcpA(2,n) - 0.01;
                end
                if d < 0
                    fcpA(2,n) = fcpA(2,n) + 0.01;
                end
                end

       end
        if  FluxPredictA(2,n)<0
            FluxPredictA(2,n)=0;
            FluxPredictLMHA(2,n)=0;
        end
          PermflowrateA(2,n) = FluxPredictA(2,n).*MembrA;%m/h*m2 = m3/h
          PermflowrateLSA(2,n) = PermflowrateA(2,n).*1000./3600;%l/s
          PermflowVOLA(2,n) = PermflowrateA(2,n).*1.*1000./3600;% m3/h * h *
1000 = L in 1 sec
          TotalPermVOLA(2,n) = TotalPermVOLA(2,n-1) + PermflowVOLA(2,n);%L

          FeedflowrateA(2,n)=PermflowrateA(2,n)./0.08;%m3/h
          FeedflowVOLA(2,n)=FeedflowrateA(2,n)./3600.*1000;%L
          CrossflowrateA(2,n)=FeedflowrateA(2,n)-PermflowrateA(2,n);%m3/h
          %Perm Conc
          SoluteFluxA(2,n)= (Ks./24).*fcpA(2,n).*(NewFeedConcA(2,n)./10^-3);
%mg/h m2
          PermConcA(2,n)= SoluteFluxA(2,n)./FluxPredictA(2,n);%mg/m3
          if FluxPredictA(2,n) == 0
            PermConcA(2,n) = 0;
          end
          PermConcMGLA(2,n) = PermConcA(2,n)./1000;%mg/L
          CrossflowConcA(2,n)= ((FeedflowrateA(2,n).*(NewFeedConcA(2,n)))-
(PermflowrateA(2,n).*PermConcMGLA(2,n)))./CrossflowrateA(2,n);%m3/h & mg/L
          if CrossflowrateA(2,n) == 0
            CrossflowConcA(2,n) = 0;
          end
          RejA(2,n)= (1 - (PermConcMGLA(2,n)./NewFeedConcA(2,n)))*100;
          %recoveryA(1,n)=(PermflowrateA(1,n)./FeedflowrateA(1,n)).*100;
            SaltMassA(2,n)=PermConcMGLA(2,n).*PermflowVOLA(2,n);%mg/L * L
            TotalSaltMassA(2,n)=TotalSaltMassA(2,n-1)+SaltMassA(2,n);%mg
            %%%%%          Membranes 2,3,4,5         %%%%%%%%%%%%%%%%%%%%
            %Number of membrane elements =5
```

```
            for memb=3:membnum
             PressurePVA(memb,n) = PressurePVA(memb-1,n) - 13.8;
             PressurePSIA(memb,n)=PressurePVA(memb,n)./6.9;
             FeedflowrateA(memb,n)=CrossflowrateA(memb-1,n);%m3/h
             FeedflowVOLA(memb,n)=FeedflowrateA(memb,n)./3600.*1000;%L

CrossflowVELA(memb,n)=FeedflowrateA(memb,n)./(3600.*W.*h.*0.88);%m3/s/(m*m)
=m/s

WallshearrateA(memb,n)=3.*CrossflowVELA(memb,n)./((h/2).*0.88);%m/s/m = /s

masstransferA(memb,n)=0.807.*((WallshearrateA(memb,n).*D.*D)./L)^(1/3);%m/s

             FeedConcA(memb,n)= CrossflowConcA(memb-1,n);%mg/L
             OsmoticPrKpaA(memb,n)=8.505.*10^-2.*FeedConcA(memb,n)-86.61;
             fcpA(memb,n) = 1.3;
                while d ~= 0
                   FluxPredictA(memb,n) = Permcoeff.*(PressurePVA(memb,n) -
fcpA(memb,n).*OsmoticPrKpaA(memb,n)); %(m/Kpa h * Kpa = m/h)
                   FluxPredictLMHA(memb,n) = PermCoeff.*(PressurePVA(memb,n) -
fcpA(memb,n).*OsmoticPrKpaA(memb,n));% lmh
                    fcpNEWA(memb,n) =
exp(FluxPredictA(memb,n)./(masstransferA(memb,n).*3600));
                    d = fcpA(memb,n) - fcpNEWA(memb,n);
                    if abs (d) < 0.1
                    break;
                    else
                    if d > 0
                    fcpA(memb,n) = fcpA(memb,n) - 0.01;
                    end
                    if d < 0
                    fcpA(memb,n) = fcpA(memb,n) + 0.01;
                    end
                    end
                end
             if FluxPredictA(memb,n)<0
                 FluxPredictA(memb,n)=0;
                 FluxPredictLMHA(memb,n)=0;
             end
             PermflowrateA(memb,n) = FluxPredictA(memb,n).*MembrA;%m/h*m2 = m3/h
             PermflowrateLSA(memb,n) = PermflowrateA(memb,n).*1000./3600;%l/s
             PermflowVOLA(memb,n) = PermflowrateA(memb,n).*1.*1000./3600;% m3/h
* h * 1000 = L in 1 sec
             TotalPermVOLA(memb,n) = TotalPermVOLA(memb,n-1) +
PermflowVOLA(memb,n);%L
             CrossflowrateA(memb,n)=FeedflowrateA(memb,n)-
PermflowrateA(memb,n);%m3/h
             %Perm Conc
             SoluteFluxA(memb,n)=
(Ks./24).*fcpA(memb,n).*(FeedConcA(memb,n)./10^-3); %mg/h m2
             PermConcA(memb,n)= SoluteFluxA(memb,n)./FluxPredictA(memb,n);%mg/m3
             if FluxPredictA(memb,n) == 0
             PermConcA(memb,n) = 0;
             end
             PermConcMGLA(memb,n) = PermConcA(memb,n)./1000;%mg/L
             CrossflowConcA(memb,n)=
((FeedflowrateA(memb,n).*(FeedConcA(memb,n)))-
(PermflowrateA(memb,n).*PermConcMGLA(memb,n)))./CrossflowrateA(memb,n);%m3/h
and mg/L
```

```
            if CrossflowrateA(memb,n) == 0
            CrossflowConcA(memb,n) = 0;
            end
            RejA(memb,n)= (1 - (PermConcMGLA(memb,n)./FeedConcA(memb,n)))*100;
            TotalSaltMassA(memb,1)=0;
            SaltMassA(memb,n)=PermConcMGLA(memb,n).*PermflowVOLA(memb,n);%mg/L
* L
            TotalSaltMassA(memb,n)=TotalSaltMassA(memb,n-
1)+SaltMassA(memb,n);%mg
           end

%              TotalPermVOLglobalA(n) = sum(PermflowVOLA(1:memb,n));%L
             TotalSaltMassglobalA(n)=sum(SaltMassA(1:memb,n));%mg
             TimeA(n)=n;
             TimeHRA(n)=TimeA(n)./3600;%hr
             TdA = n;
             PressurePVA(1,n)=PressurePVA(2,n);
    else
        PressurePVA(1,n)=PressurePVA(1,n-1);
        %TotalPermVOLglobalA(n) = TotalPermVOLglobalA(n-1);
        TotalPermVOLglobalA(n)=0;
%         TotalSaltMassglobalA(n) = TotalSaltMassglobalA(n-1);
        TotalSaltMassglobalA(n)=0;
    end
%     TotalPermeateVOLA(n)=TotalPermeateVOLA(n-1)+TotalPermVOLglobalA(n);%%L

    PressurePVC(1,n)=InitialPressure;%Kpa (PVA is desalinating, PVB is filling,
PVC is initial gas pressure)
    %TotalSaltMassglobalB(n) = TotalSaltMassglobalB(n-1);%mg
    TotalSaltMassglobalB(n) =0;
TotalSaltMassglobalC(n)=0;
        %         TotalPermVOLglobalC(n)=TotalPermVOLglobalC(n-1);%L
    n=n+1;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  TotalPermVOLglobalA = sum(PermflowVOLA(1:memb,:));%L
TotalPermeateVOLA = sum(TotalPermVOLglobalA);%L
TotalSaltMassA = sum(TotalSaltMassglobalA(1:n-1));
 cyclenum = cyclenum+1;
n=TfB+1;
%***********************************************************************
%%% STEP 3 ---> Desal B and Fill C

%%%  Initial Conditions to Fill C -----> First Fill Conditions
%TurbinePowerKW(1,n)=0;%%%here n = TfB
PumpPowerKW(1,TfB) = Pumpefficiency.*TurbinePowerKW(1,TfB);
PumpflowrateC(1,TfB) = TurbinePowerKW(1,TfB)./InitialPressure; %m3/s
PumpflowVOLC(1,TfB) = PumpflowrateB(1,TfB);%m3 in 1 second
NewAirVOLC(1,TfB) = InitialVolume-PumpflowVOLC(1,TfB); %m3
NewWaterVOLC(1,TfB)=InitialVolume-NewAirVOLC(1,TfB); %m3
PressurePVC(1,TfB) = InitialPressure.*InitialVolume./NewAirVOLC(1,TfB); %kpa

%%Initial conditions for Desal B ----> First Desal conditions
AirVOLB(2,TfB) = NewAirVOLB(1,TfB).*1000;
PressurePVB(2,TfB)=PressurePVB(1,TfB);
NewFeedConcB(2,TfB)=32000; %mg/L
OsmoticPrKpaB(2,TfB)=(NewFeedConcB(2,TfB).* 8.505.*10^-2)- 86.61;%Kpa
FluxPredictB(2,TfB)=Permcoeff.*(PressurePVB(2,TfB)-OsmoticPrKpaB(2,TfB));%m/h
PermflowrateB(2,TfB)=FluxPredictB(2,TfB).*MembrA;%m3/h
```

```
FeedflowrateB(2,TfB)=PermflowrateB(2,TfB)./recoveryB;%m3/h
FeedflowVOLB(2,TfB)=FeedflowrateB(2,TfB)./3600.*1000;%L
NewfeedVOLB(2,TfB)=NewWaterVOLB(1,TfB).*1000-FeedflowVOLB(2,TfB);%L
PermflowVOLB(2,TfB)=PermflowrateB(2,TfB).*1000./3600;%L
TotalPermVOLB(2,TfB)=PermflowVOLB(2,TfB);%L
TotalSaltMassB(2,TfB)=0;%
TotalPermVOLB(3:membnum,TfB)=0;%


PressurePVA(1,TfB)=PressurePVA(1,TdA);

TfC=TfB;
TdB=TfB;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
while PressurePVC(1,n-1) <= 6900 || PressurePVB(2,n-1)>Pressurelim  %1000 psi
   if PressurePVB(2,n-1)< Pressurelim && n >=length(windspeed)
        break;
    end
    if PressurePVC(1,n-1) <= 6900 && n<=length(windspeed)

TurbinePower(1,n)=0.5.*Cpt.*airdensity.*rotorarea.*windspeed(1,n).*windspeed(1,
n).*windspeed(n);%w
        TurbinePowerKW(1,n)=TurbinePower(1,n)/1000;%kw
        PumpPower(1,n)=Pumpefficiency.* TurbinePowerKW(1,n);%kw
        PumpflowrateC(1,n)=PumpPower(1,n)./PressurePVC(1,n-1);%m3/s
        PumpflowVOLC(1,n)=PumpflowrateC(1,n);%m3 in 1 second
        NewAirVOLC(1,n)=NewAirVOLC(1,n-1)-PumpflowVOLC(1,n); %m3
        NewWaterVOLC(1,n)=InitialVolume-NewAirVOLC(1,n); %m3
        NewWaterVOLLC(1,n)=NewWaterVOLC(1,n).*1000; %L
%          PressurePVC(1,n)=PressurePVC(1,n-1).*NewAirVOLC(1,n-
1)./NewAirVOLC(1,n);%kpa
        PressurePVC(1,n)=nRT./NewAirVOLC(1,n);%Kpa.m3/m3=Kpa
        FillTimeC(1,n)=n;
        FillTimeHRC(1,n)=FillTimeC(1,n)./3600;
        TfC = n;
    else
        PressurePVC(1,n)=PressurePVC(1,n-1);
    end


    if PressurePVB(2,n-1)>Pressurelim

        CrossflowVELB(2,n)=FeedflowrateB(2,n-1)./(3600.*W.*h.*0.88);%m3/s/(m*m)
=m/s
        WallshearrateB(2,n)=3.*CrossflowVELB(2,n)./((h/2).*0.88);%m/s/m = /s
        masstransferB(2,n)=0.807.*((WallshearrateB(2,n).*D.*D)./L)^(1/3);%m/s

%          masstransferB(2,n)=0.807.*((3.*(FeedflowrateB(2,n-
1)./3600).*D.*D)./(2.*(h./2).*(h./2).*W.*L))^(1/3);%m/s
        NewfeedVOLB(2,n)= NewfeedVOLB(2,n-1)-FeedflowVOLB(2,n-1);%L
        NewFeedConcB(2,n) = 32000; %mg/L
        OsmoticPrKpaB(2,n) = (NewFeedConcB(2,n).* 8.505.*10^-2)- 86.61;
        AirVOLB(2,n) = InitialVolume.*1000 - NewfeedVOLB(2,n);%L
        dPB(2,n) = nRT.*1000.*((1./AirVOLB(2,n))- (1./AirVOLB(2,n-1)));% L Kpa
* (1/L) = Kpa
        PressurePVB(2,n) = dPB(2,n) + PressurePVB(2,n-1); % Kpa
        PressurePSIB(2,n)=PressurePVB(2,n)./6.9;
```

```
        fcpB(2,n) = 1.3;
       while d ~= 0
          FluxPredictB(2,n) = Permcoeff.*(PressurePVB(2,n) -
fcpB(2,n).*OsmoticPrKpaB(2,n)); %(m/Kpa h * Kpa = m/h)
          FluxPredictLMHB(2,n) = PermCoeff.*(PressurePVB(2,n) -
fcpB(2,n).*OsmoticPrKpaB(2,n));% lmh

          fcpNEWB(2,n) = exp(FluxPredictB(2,n)./(masstransferB(2,n).*3600));
                d = fcpB(2,n) - fcpNEWB(2,n);
                if abs (d) < 0.1
                break;
                else
                if d > 0
                fcpB(2,n) = fcpB(2,n) - 0.01;
                end
                if d < 0
                fcpB(2,n) = fcpB(2,n) + 0.01;
                end
                end
       end
        if FluxPredictB(2,n)<0
            FluxPredictB(2,n)=0;
            FluxPredictLMHB(2,n)=0;
        end
          PermflowrateB(2,n) = FluxPredictB(2,n).*MembrA;%m/h*m2 = m3/h
          PermflowrateLSB(2,n) = PermflowrateB(2,n).*1000./3600;%l/s
          PermflowVOLB(2,n) = PermflowrateB(2,n).*1.*1000./3600;% m3/h * h *
1000 = L in 1 sec
          TotalPermVOLB(2,n) = TotalPermVOLB(2,n-1) + PermflowVOLB(2,n);%L
          FeedflowrateB(2,n)=PermflowrateB(2,n)./0.08;%m3/h
          FeedflowVOLB(2,n)=FeedflowrateB(2,n).*1000./3600;%L in 1 sec
          CrossflowrateB(2,n)=FeedflowrateB(2,n)-PermflowrateB(2,n);%m3/h
          %Perm Conc
          SoluteFluxB(2,n)= (Ks./24).*fcpB(2,n).*(NewFeedConcB(2,n)./10^-3);
%mg/h m2
          PermConcB(2,n)= SoluteFluxB(2,n)./FluxPredictB(2,n);%mg/m3
          if FluxPredictB(2,n) == 0
          PermConcB(2,n) = 0;
          end
          PermConcMGLB(2,n) = PermConcB(2,n)./1000;%mg/L
          CrossflowConcB(2,n)= ((FeedflowrateB(2,n).*(NewFeedConcB(2,n)))-
(PermflowrateB(2,n).*PermConcMGLB(2,n)))./CrossflowrateB(2,n);
          if CrossflowrateB(2,n) == 0
          CrossflowConcB(2,n) = 0;
          end
          RejB(2,n)= (1-(PermConcMGLB(2,n)./NewFeedConcB(2,n)))*100;
          %recoveryB(1,n)=(PermflowrateB(1,n)./FeedflowrateB(1,n)).*100;

            TotalSaltMassB(2,1)=0;
            SaltMassB(2,n)=PermConcMGLB(2,n).*PermflowVOLB(2,n);%mg/L * L
            TotalSaltMassB(2,n)=TotalSaltMassB(2,n-1)+SaltMassB(2,n);%mg
            %%%%%          Membranes 2,3,4,5         %%%%%%%%%%%%%%%%%%%%
            %Number of membrane elements =5
           for memb=3:membnum
            PressurePVB(memb,n)=PressurePVB(memb-1,n)-13.8;
            PressurePSIB(memb,n)=PressurePVB(memb,n)./6.9;
            FeedflowrateB(memb,n)=CrossflowrateB(memb-1,n);%m3/h
            FeedflowVOLB(memb,n)=FeedflowrateB(memb,n).*1000./3600;%L in 1 sec
```

```
%
masstransferB(memb,n)=0.807.*((3.*(FeedflowrateB(memb,n)./3600).*D.*D)./(2.*(h.
/2).*(h./2).*W.*L))^(1/3);%m/s


CrossflowVELB(memb,n)=FeedflowrateB(memb,n)./(3600.*W.*h.*0.88);%m3/s/(m*m)
=m/s

WallshearrateB(memb,n)=3.*CrossflowVELB(memb,n)./((h/2).*0.88);%m/s/m = /s

masstransferB(memb,n)=0.807.*((WallshearrateB(memb,n).*D.*D)./L)^(1/3);%m/s

            FeedConcB(memb,n)=CrossflowConcB(memb-1,n);%mg/L
            OsmoticPrKpaB(memb,n)=8.505.*10^-2.*FeedConcB(memb,n)-86.61;
            fcpB(memb,n) = 1.3;
               while d ~= 0
                  FluxPredictB(memb,n) = Permcoeff.*(PressurePVB(memb,n)-
fcpB(memb,n).*OsmoticPrKpaB(memb,n)); %(m/Kpa h * Kpa = m/h)
                  FluxPredictLMHB(memb,n) = PermCoeff.*(PressurePVB(memb,n)-
fcpB(memb,n).*OsmoticPrKpaB(memb,n));% lmh
                  if FluxPredictB(memb,n)<0
                    FluxPredictB(memb,n)=0;
                    FluxPredictLMHB(memb,n)=0;
                  end
                    fcpNEWB(memb,n) =
exp(FluxPredictB(memb,n)./(masstransferB(memb,n).*3600));
                    d = fcpB(memb,n) - fcpNEWB(memb,n);
                    if abs (d) < 0.1
                        break;
                    else
                    if d > 0
                        fcpB(memb,n) = fcpB(memb,n) - 0.01;
                    end
                    if d < 0
                        fcpB(memb,n) = fcpB(memb,n) + 0.01;
                    end
                    end
               end

            PermflowrateB(memb,n)=FluxPredictB(memb,n).*MembrA;%m/h*m2 = m3/h
            PermflowrateLSB(memb,n)=PermflowrateB(memb,n).*1000./3600;%l/s
            PermflowVOLB(memb,n)=PermflowrateB(memb,n).*1.*1000./3600;% m3/h *
h * 1000 = L in 1 sec
            TotalPermVOLB(memb,n) = TotalPermVOLB(memb,n-1) +
PermflowVOLB(memb,n);%L
            CrossflowrateB(memb,n)=FeedflowrateB(memb,n)-
PermflowrateB(memb,n);%m3/h
            %Perm Conc
            SoluteFluxB(memb,n)=
(Ks./24).*fcpB(memb,n).*(FeedConcB(memb,n)./10^-3); %mg/h m2
            PermConcB(memb,n)= SoluteFluxB(memb,n)./FluxPredictB(memb,n);%mg/m3
            if FluxPredictB(memb,n) == 0
                PermConcB(memb,n) = 0;
            end
                PermConcMGLB(memb,n) = PermConcB(memb,n)./1000;%mg/L
                CrossflowConcB(memb,n)=
((FeedflowrateB(memb,n).*(FeedConcB(memb,n)))-
(PermflowrateB(memb,n).*PermConcMGLB(memb,n)))./CrossflowrateB(memb,n);
            if  CrossflowrateB(memb,n) == 0
```

```
                CrossflowConcB(memb,n) = 0;
            end
                RejB(memb,n)= (1-
(PermConcMGLB(memb,n)./FeedConcB(memb,n)))*100;
                TotalSaltMassB(memb,1)=0;

SaltMassB(memb,n)=PermConcMGLB(memb,n).*PermflowVOLB(memb,n);%mg/L * L
                TotalSaltMassB(memb,n)=TotalSaltMassB(memb,n-
1)+SaltMassB(memb,n);%mg
           end
             TotalSaltMassglobalB(n)=sum(SaltMassB(1:memb,n));%mg
          %%%%%%%%%%%%%%%
%

             PressurePVB(1,n)=PressurePVB(2,n);
             TimeB(n)=n;
             TimeHRB(n)=TimeB(n)./3600;%hr
             TdB = n;
           else
                PressurePVB(1,n)=PressurePVB(1,n-1);
        TotalSaltMassglobalB(n)=0;
    end
    PressurePVA(1,n)=PressurePVA(1,TdA);%since PVA is at the desal pressure now
    FluxPredictA(2:membnum, n)=0;
            TotalPermVOLglobalC(n)=0;%L
    n=n+1;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 TotalPermVOLglobalB = sum(PermflowVOLB(1:memb,:));
TotalPermeateVOLB = sum(TotalPermVOLglobalB);%L
TotalSaltMassB = sum(TotalSaltMassglobalB(1:n-1));
cyclenum = cyclenum+1;
% TotalPermeateVOLABC = TotalPermeateVOLA+TotalPermeateVOLB+TotalPermeateVOLC;
TotalPermeateVOLA=0;
TotalPermeateVOLB=0;
TotalPermeateVOLC=0;
% TotalSaltMassABC=TotalSaltMassA+TotalSaltMassB+TotalSaltMassC;
TotalSaltMassA=0;
TotalSaltMassB=0;
TotalSaltMassC=0;

%*************************************************************************
%************   FILL/DESAL Continuous cycles begin
here******************************%
%%% Fill PVA from 4800 to 6900 and PVC desal
%n=n-1;
n=TfC+1;
flag = 1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
while flag==1
    n
%     n=TfC+1;
        if n>length(windspeed) && PressurePVA(2,n-1)<= Pressurelim &&
PressurePVB(2,n-1)<= Pressurelim && PressurePVC(2,n-1)<= Pressurelim
         flag=0;
            break;
    end
%     if n<=length(windspeed)
```

```
%         n=TfC+1;
%     end

   % PressurePVA(1,TfB:TfC)=PressurePVA(1,TfB);
    NewAirVOLA(1,TfC)=nRT./PressurePVA(1,TfC);

    AirVOLC(2,TfC) = NewAirVOLC(1,TfC).*1000;%L
    PressurePVC(2,TfC)=PressurePVC(1,TfC);
    %AirVOLC(2,TfC)=(nRT./PressurePVC(1,TfC)).*1000;%Kpa.m3 / Kpa *1000= L
    NewfeedVOLC(2,TfC)=(InitialVolume.*1000)-AirVOLC(2,TfC);%L
    NewFeedConcC(2,TfC)=32000; %mg/L
    OsmoticPrKpaC(2,TfC)=(NewFeedConcC(2,TfC).* 8.505.*10^-2)- 86.61;%Kpa
    FluxPredictC(2,TfC)=Permcoeff.*(PressurePVC(2,TfC)-
OsmoticPrKpaC(2,TfC));%m/h
    PermflowrateC(2,TfC)=FluxPredictC(2,TfC).*MembrA;%m3/h
    FeedflowrateC(2,TfC)=PermflowrateC(2,TfC)./recoveryC;%m3/h
    FeedflowVOLC(2,TfC)=FeedflowrateC(2,TfC)./3600.*1000;%L
    PermflowVOLC(2,TfC)=PermflowrateC(2,TfC).*1000./3600;%L
    TotalPermVOLC(2,TfC)=PermflowVOLC(2,TfC);%L
    TotalSaltMassC(2,TfC)=0;
    TotalPermVOLC(2:membnum,TfC)=0;

    TdC = TfC;

%     TotalPermeateVOLA(TfC)=TotalPermeateVOLA(n-1);%%L
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    while PressurePVA(1,n-1) <= 6900 || PressurePVC(2,n-1)>Pressurelim %1000
psi
          if n>length(windspeed) && PressurePVC(2,n-1)<= Pressurelim
         break;
    end
        if PressurePVA(1,n-1)<=6900 && n<=length(windspeed)

TurbinePower(1,n)=0.5.*Cpt.*airdensity.*rotorarea.*windspeed(1,n).*windspeed(1,
n).*windspeed(1,n);%w
            TurbinePowerKW(1,n)=TurbinePower(1,n)/1000;%kw
            PumpPower(1,n)=Pumpefficiency.*TurbinePowerKW(1,n);%kw
            PumpflowrateA(1,n)=PumpPower(1,n)./PressurePVA(1,n-1);%m3/s
            PumpflowVOLA(1,n)=PumpflowrateA(1,n);%m3 in 1 second
            NewAirVOLA(1,n)=NewAirVOLA(1,n-1)-PumpflowVOLA(1,n); %m3
            NewWaterVOLA(1,n)=InitialVolume-NewAirVOLA(1,n); %m3
            NewWaterVOLLA(1,n)=NewWaterVOLA(1,n).*1000; %L
%             PressurePVA(1,n)=PressurePVA(1,n-1).*NewAirVOLA(1,n-
1)./NewAirVOLA(1,n);%kpa
            PressurePVA(1,n)=nRT./NewAirVOLA(1,n);%Kpa.m3/m3=Kpa
            FillTimeA(1,n)=n;
            FillTimeHRA(1,n)=FillTimeA(1,n)./3600;
            TfA = n;
        else
            PressurePVA(1,n)=PressurePVA(1,n-1);
        end

        if PressurePVC(2,n-1)>Pressurelim

            CrossflowVELC(2,n)=FeedflowrateC(2,n-
1)./(3600.*W.*h.*0.88);%m3/s/(m*m) =m/s
            WallshearrateC(2,n)=3.*CrossflowVELC(2,n)./((h/2).*0.88);%m/s/m =
/s
```

```
masstransferC(2,n)=0.807.*((WallshearrateC(2,n).*D.*D)./L)^(1/3);%m/s
%
            NewfeedVOLC(2,n)= NewfeedVOLC(2,n-1)-(FeedflowVOLC(2,n-1));%L
            NewFeedConcC(2,n) = 32000; %mg/L
            OsmoticPrKpaC(2,n) = (NewFeedConcC(2,n).* 8.505.*10^-2)- 86.61;
            AirVOLC(2,n) = (InitialVolume.*1000)-NewfeedVOLC(2,n);%L
            dPC(2,n) = nRT.*1000.*((1./AirVOLC(2,n))- (1./AirVOLC(2,n-1)));% L
Kpa * (1/L) = Kpa
            PressurePVC(2,n) = dPC(2,n) + PressurePVC(2,n-1); % Kpa
            PressurePSIC(2,n)=PressurePVC(2,n)./6.9;

            fcpC(2,n) = 1.3;
           while d ~= 0
              FluxPredictC(2,n) = Permcoeff.*(PressurePVC(2,n)-
fcpC(2,n).*OsmoticPrKpaC(2,n)); %(m/Kpa h * Kpa = m/h)
              FluxPredictLMHC(2,n) = PermCoeff.*(PressurePVC(2,n)-
fcpC(2,n).*OsmoticPrKpaC(2,n));% lmh
              fcpNEWC(2,n) =
exp(FluxPredictC(2,n)./(masstransferC(2,n).*3600));
                    d = fcpC(2,n)-fcpNEWC(2,n);
                    if abs(d)<0.1
                    break;
                    else
                    if d > 0
                    fcpC(2,n)=fcpC(2,n)-0.01;
                    end
                    if d < 0
                    fcpC(2,n)=fcpC(2,n)+0.01;
                    end
                    end
           end
            if FluxPredictC(2,n)<0
               FluxPredictC(2,n)=0;
               FluxPredictLMHC(2,n)=0;
            end
              PermflowrateC(2,n)=FluxPredictC(2,n).*MembrA;%m/h*m2 = m3/h
              PermflowrateLSC(2,n)=PermflowrateC(2,n).*1000./3600;%l/s
              PermflowVOLC(2,n)=PermflowrateC(2,n).*1.*1000./3600;% m3/h * h *
1000 = L in 1 sec
              TotalPermVOLC(2,n)=TotalPermVOLC(2,n-1)+PermflowVOLC(2,n);%L

              FeedflowrateC(2,n)=PermflowrateC(2,n)./0.08;%m3/h
              FeedflowVOLC(2,n)=FeedflowrateC(2,n)./3600.*1000;%L
              CrossflowrateC(2,n)=FeedflowrateC(2,n)-PermflowrateC(2,n);%m3/h
              %Perm Conc
              SoluteFluxC(2,n)= (Ks./24).*fcpC(2,n).*(NewFeedConcC(2,n)./10^-
3); %mg/h m2
              PermConcC(2,n)= SoluteFluxC(2,n)./FluxPredictC(2,n);%mg/m3
              if FluxPredictC(2,n) == 0
              PermConcC(2,n) = 0;
              end
              PermConcMGLC(2,n) = PermConcC(2,n)./1000;%mg/L
              CrossflowConcC(2,n)= ((FeedflowrateC(2,n).*(NewFeedConcC(2,n)))-
(PermflowrateC(2,n).*PermConcMGLC(2,n)))./CrossflowrateC(2,n);%mg/L
              if CrossflowrateC(2,n) == 0
              CrossflowConcC(2,n) = 0;
              end
              RejC(2,n)= (1-(PermConcMGLC(2,n)./NewFeedConcC(2,n)))*100;
```

```
              %recoveryC(1,n)=(PermflowrateC(1,n)./FeedflowrateC(1,n)).*100;
              TotalSaltMassC(2,TfC)=0;
              SaltMassC(2,n)=PermConcMGLC(2,n).*PermflowVOLC(2,n);%mg/L * L
              TotalSaltMassC(2,n)=TotalSaltMassC(2,n-1)+SaltMassC(2,n);%mg
              %%%%%         Membranes 2,3,4,5         %%%%%%%%%%%%%%%%%%%%
              %Number of membrane elements =5
               for memb=3:membnum

                PressurePVC(memb,n) = PressurePVC(memb-1,n) - 13.8;
                PressurePSIC(memb,n)=PressurePVC(memb,n)./6.9;
                FeedflowrateC(memb,n)=CrossflowrateC(memb-1,n);%m3/h


CrossflowVELC(memb,n)=FeedflowrateC(memb,n)./(3600.*W.*h.*0.88);%m3/s/(m*m)
=m/s

WallshearrateC(memb,n)=3.*CrossflowVELC(memb,n)./((h/2).*0.88);%m/s/m = /s

masstransferC(memb,n)=0.807.*((WallshearrateC(memb,n).*D.*D)./L)^(1/3);%m/s

%
masstransferC(memb,n)=0.807.*((3.*(FeedflowrateC(memb,n)./3600).*D.*D)./(2.*(h.
/2).*(h./2).*W.*L))^(1/3);%m/s
                FeedConcC(memb,n)= CrossflowConcC(memb-1,n);%mg/L
                OsmoticPrKpaC(memb,n)=8.505.*10^-2.*FeedConcC(memb,n)-86.61;

                fcpC(memb,n) = 1.3;
                   while d ~= 0
                      FluxPredictC(memb,n) = Permcoeff.*(PressurePVC(memb,n) -
fcpC(memb,n).*OsmoticPrKpaC(memb,n)); %(m/Kpa h * Kpa = m/h)
                      FluxPredictLMHC(memb,n) = PermCoeff.*(PressurePVC(memb,n)
- fcpC(memb,n).*OsmoticPrKpaC(memb,n));% lmh
                        fcpNEWC(memb,n) =
exp(FluxPredictC(memb,n)./(masstransferC(memb,n).*3600));
                        d = fcpC(memb,n) - fcpNEWC(memb,n);
                        if abs (d) < 0.1
                        break;
                        else
                        if d > 0
                        fcpC(memb,n) = fcpC(memb,n) - 0.01;
                        end
                        if d < 0
                        fcpC(memb,n) = fcpC(memb,n) + 0.01;
                        end

                        end
                   end
                if FluxPredictC(memb,n)<0
                    FluxPredictC(memb,n)=0;
                    FluxPredictLMHC(memb,n)=0;
                end
                PermflowrateC(memb,n) = FluxPredictC(memb,n).*MembrA;%m/h*m2 =
m3/h
                PermflowrateLSC(memb,n) =
PermflowrateC(memb,n).*1000./3600;%l/s
                PermflowVOLC(memb,n) = PermflowrateC(memb,n).*1.*1000./3600;%
m3/h * h * 1000 = L in 1 sec
                TotalPermVOLC(memb,n) = TotalPermVOLC(memb,n-1) +
PermflowVOLC(memb,n);%L
```

```
                CrossflowrateC(memb,n) = FeedflowrateC(memb,n)-
PermflowrateC(memb,n);%m3/h

                %Perm Conc
                SoluteFluxC(memb,n)=
(Ks./24).*fcpC(memb,n).*(FeedConcC(memb,n)./10^-3); %mg/h m2
                PermConcC(memb,n)=
SoluteFluxC(memb,n)./FluxPredictC(memb,n);%mg/m3
                if FluxPredictC(memb,n) == 0
                PermConcC(memb,n) = 0;
                end
                PermConcMGLC(memb,n) = PermConcC(memb,n)./1000;%mg/L
                CrossflowConcC(memb,n)=
((FeedflowrateC(memb,n).*(FeedConcC(memb,n)))-
(PermflowrateC(memb,n).*PermConcMGLC(memb,n)))./CrossflowrateC(memb,n);
                if CrossflowrateC(memb,n) == 0
                CrossflowConcC(memb,n) = 0;
                end
                RejC(memb,n)= (1 -
(PermConcMGLC(memb,n)./FeedConcC(memb,n)))*100;
                TotalSaltMassC(memb,1)=0;

SaltMassC(memb,n)=PermConcMGLC(memb,n).*PermflowVOLC(memb,n);%mg/L * L
                TotalSaltMassC(memb,n)=TotalSaltMassC(memb,n-
1)+SaltMassC(memb,n);%mg
               end
%                TotalPermVOLglobalC(n) = sum(PermflowVOLC(1:memb,n));%L
               TotalSaltMassglobalC(n)=sum(SaltMassC(1:memb,n));%mg
            %%%%%%%%%%%%%%%%%%%%%%
%                PermVOLglobalC(n) = sum(PermflowVOLC(2:memb,n));%L
%                SaltMassglobalC(n)=sum(SaltMassC(2:memb,n));%mg
               TimeC(n)=n;
               TimeHRC(n)=TimeC(n)./3600;%hr
               TdC = n;
               PressurePVC(1,n)=PressurePVC(2,n);
        else
            PressurePVC(1,n)=PressurePVC(1,n-1);
%              TotalPermVOLglobalC(n) = TotalPermVOLglobalC(n-1);
%          TotalSaltMassglobalC(n) = TotalSaltMassglobalC(n-1);
        TotalSaltMassglobalC(n)=0;
        end

        n=n+1;
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    PressurePVB(1,TdB:n-1) = PressurePVB (1,TdB);
 TotalPermVOLglobalC = sum(PermflowVOLC(1:memb,:));
TotalPermeateVOLC = sum(TotalPermVOLglobalC);%L
TotalSaltMassC = sum(TotalSaltMassglobalC(1:n-1));
cyclenum = cyclenum+1;

    %%% Fill PVB from 4800 to 6900 and PVA desal
if n>length(windspeed) && PressurePVA(1,n-1)<= Pressurelim && PressurePVB(2,n-
1)<= Pressurelim && PressurePVC(2,n-1)<= Pressurelim
         flag=0;
            break;
    end
    %%Fill PVB
```

```
   % if n<=length(windspeed)
        n=TfA+1;
   % end
   % PressurePVB(1,TfC:TfA)=PressurePVB(1,TfC);
    NewAirVOLB(1,TfA)=nRT./PressurePVB(1,TfA);

    %% Desal PVA
    AirVOLA(2,TfA) = NewAirVOLA(1,TfA).*1000;%L
    PressurePVA(2,TfA)=PressurePVA(1,TfA);
    %AirVOLA(2,TfA)=(nRT./PressurePVA(1,TfA)).*1000;%Kpa.m3 / Kpa *1000= L
    NewfeedVOLA(2,TfA)=(InitialVolume.*1000)-AirVOLA(2,TfA);%L
    NewFeedConcA(2,TfA)=32000; %mg/L
    OsmoticPrKpaA(2,TfA)=(NewFeedConcA(2,TfA).* 8.505.*10^-2)- 86.61;%Kpa
    FluxPredictA(2,TfA)=Permcoeff.*(PressurePVA(2,TfA)-
OsmoticPrKpaA(2,TfA));%m/h
    PermflowrateA(2,TfA)=FluxPredictA(2,TfA).*MembrA;%m3/h
    FeedflowrateA(2,TfA)=PermflowrateA(2,TfA)./recoveryA;%m3/h
    FeedflowVOLA(2,TfA)=FeedflowrateA(2,TfA)./3600.*1000;%L
    PermflowVOLA(2,TfA)=PermflowrateA(2,TfA).*1000./3600;%L
    TotalPermVOLA(2,TfA)=PermflowVOLA(2,TfA);%L
    TotalSaltMassA(2,TfA)=0;
    TotalPermVOLA(2:membnum,TfA)=0;
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    while PressurePVB(1,n-1) <= 6900 || PressurePVA(2,n-1)>Pressurelim   %1000
psi

        if n>length(windspeed) && PressurePVA(2,n-1)<= Pressurelim
         break;
        end
        if PressurePVB(1,n-1)<=6900 && n<=length(windspeed)

TurbinePower(1,n)=0.5.*Cpt.*airdensity.*rotorarea.*windspeed(1,n).*windspeed(1,
n).*windspeed(1,n);%w
            TurbinePowerKW(1,n)=TurbinePower(1,n)/1000;%kw
            PumpPower(1,n)=Pumpefficiency.*TurbinePowerKW(1,n);%kw
            PumpflowrateB(1,n)=PumpPower(1,n)./PressurePVB(1,n-1);%m3/s
            PumpflowVOLB(1,n)=PumpflowrateB(1,n);%m3 in 1 second
            NewAirVOLB(1,n)=NewAirVOLB(1,n-1)-PumpflowVOLB(1,n); %m3
            NewWaterVOLB(1,n)=InitialVolume-NewAirVOLB(1,n); %m3
            NewWaterVOLLB(1,n)=NewWaterVOLB(1,n).*1000; %L
%             PressurePVB(1,n)=PressurePVB(1,n-1).*NewAirVOLB(1,n-
1)./NewAirVOLB(1,n);%kpa
            PressurePVB(1,n)=nRT./NewAirVOLB(1,n);%Kpa.m3/m3=Kpa
            FillTimeB(1,n)=n;
            FillTimeHRB(1,n)=FillTimeB(1,n)./3600;
            TfB = n;

            FluxPredictA(2:membnum,
n)=0;%%%%%%%%%%%%%%%%%%%%%%%%%???????????????????????????????????????
        else
            PressurePVB(1,n)=PressurePVB(1,n-1);
%             PressurePVC(1,n)=PressurePVC(1,n-1);
        end


         if  PressurePVA(2,n-1)>Pressurelim
```

```
            CrossflowVELA(2,n)=FeedflowrateA(2,n-
1)./(3600.*W.*h.*0.88);%m3/s/(m*m) =m/s
            WallshearrateA(2,n)=3.*CrossflowVELA(2,n)./((h/2).*0.88);%m/s/m =
/s

masstransferA(2,n)=0.807.*((WallshearrateA(2,n).*D.*D)./L)^(1/3);%m/s
            NewfeedVOLA(2,n)= NewfeedVOLA(2,n-1)-FeedflowVOLA(2,n-1);%L
            NewFeedConcA(2,n)=32000; %mg/L
            OsmoticPrKpaA(2,n) = (NewFeedConcA(2,n).* 8.505.*10^-2)- 86.61;
            AirVOLA(2,n) = (InitialVolume.*1000) - NewfeedVOLA(2,n);%L
            dPA(2,n) = nRT.*1000.*((1./AirVOLA(2,n))- (1./AirVOLA(2,n-1)));% L
Kpa * (1/L) = Kpa
            PressurePVA(2,n) = dPA(2,n) + PressurePVA(2,n-1); % Kpa
            PressurePSIA(2,n)=PressurePVA(2,n)./6.9;
            fcpA(2,n) = 1.3;
           while d ~= 0
                FluxPredictA(2,n) = Permcoeff.*(PressurePVA(2,n) -
fcpA(2,n).*OsmoticPrKpaA(2,n)); %(m/Kpa h * Kpa = m/h)
                FluxPredictLMHA(2,n) = PermCoeff.*(PressurePVA(2,n) -
fcpA(2,n).*OsmoticPrKpaA(2,n));% lmh
                fcpNEWA(2,n) =
exp(FluxPredictA(2,n)./(masstransferA(2,n).*3600));
                    d = fcpA(2,n) - fcpNEWA(2,n);
                     if abs (d) < 0.1
                        break;
                     else
                    if d > 0
                        fcpA(2,n) = fcpA(2,n) - 0.01;
                    end
                    if d < 0
                        fcpA(2,n) = fcpA(2,n) + 0.01;
                    end
                    end
           end
            if  FluxPredictA(2,n)<0
                FluxPredictA(2,n)=0;
                FluxPredictLMHA(2,n)=0;
            end
              PermflowrateA(2,n) = FluxPredictA(2,n).*MembrA;%m/h*m2 = m3/h
              PermflowrateLSA(2,n) = PermflowrateA(2,n).*1000./3600;%l/s
              PermflowVOLA(2,n) = PermflowrateA(2,n).*1.*1000./3600;% m3/h * h
* 1000 = L in 1 sec
              TotalPermVOLA(2,n) = TotalPermVOLA(2,n-1) + PermflowVOLA(2,n);%L

              FeedflowrateA(2,n)=PermflowrateA(2,n)./0.08;%m3/h
              FeedflowVOLA(2,n)=FeedflowrateA(2,n)./3600.*1000;%L
              CrossflowrateA(2,n)=FeedflowrateA(2,n)-PermflowrateA(2,n);%m3/h
              %Perm Conc
              SoluteFluxA(2,n)= (Ks./24).*fcpA(2,n).*(NewFeedConcA(2,n)./10^-
3); %mg/h m2
              PermConcA(2,n)= SoluteFluxA(2,n)./FluxPredictA(2,n);%mg/m3
              if FluxPredictA(2,n) == 0
                PermConcA(2,n) = 0;
              end
              PermConcMGLA(2,n) = PermConcA(2,n)./1000;%mg/L
              CrossflowConcA(2,n)= ((FeedflowrateA(2,n).*(NewFeedConcA(2,n)))-
(PermflowrateA(2,n).*PermConcMGLA(2,n)))./CrossflowrateA(2,n);%m3/h & mg/L
              if CrossflowrateA(2,n) == 0
                CrossflowConcA(2,n) = 0;
```

```
                end
                RejA(2,n)= (1 - (PermConcMGLA(2,n)./NewFeedConcA(2,n)))*100;

SaltMassA(2,n)=PermConcMGLA(2,n).*PermflowVOLA(2,n);%mg/L * L
                 TotalSaltMassA(2,n)=TotalSaltMassA(2,n-1)+SaltMassA(2,n);%mg
                 %%%%%          Membranes 2,3,4,5          %%%%%%%%%%%%%%%%%%%%
                 %Number of membrane elements =5
                for memb=3:membnum
                 PressurePVA(memb,n) = PressurePVA(memb-1,n) - 13.8;
                 PressurePSIA(memb,n)=PressurePVA(memb,n)./6.9;
                 FeedflowrateA(memb,n)=CrossflowrateA(memb-1,n);%m3/h
                 FeedflowVOLA(memb,n)=FeedflowrateA(memb,n)./3600.*1000;%L


CrossflowVELA(memb,n)=FeedflowrateA(memb,n)./(3600.*W.*h.*0.88);%m3/s/(m*m)
=m/s

WallshearrateA(memb,n)=3.*CrossflowVELA(memb,n)./((h/2).*0.88);%m/s/m = /s

masstransferA(memb,n)=0.807.*((WallshearrateA(memb,n).*D.*D)./L)^(1/3);%m/s

%
masstransferA(memb,n)=0.807.*((3.*(FeedflowrateA(memb,n)./3600).*D.*D)./(2.*(h.
/2).*(h./2).*W.*L))^(1/3);%m/s
                 FeedConcA(memb,n)= CrossflowConcA(memb-1,n);%mg/L
                 OsmoticPrKpaA(memb,n)=8.505.*10^-2.*FeedConcA(memb,n)-86.61;
                 fcpA(memb,n) = 1.3;
                    while d ~= 0
                       FluxPredictA(memb,n) = Permcoeff.*(PressurePVA(memb,n) -
fcpA(memb,n).*OsmoticPrKpaA(memb,n)); %(m/Kpa h * Kpa = m/h)
                       FluxPredictLMHA(memb,n) = PermCoeff.*(PressurePVA(memb,n)
- fcpA(memb,n).*OsmoticPrKpaA(memb,n));% lmh
                         fcpNEWA(memb,n) =
exp(FluxPredictA(memb,n)./(masstransferA(memb,n).*3600));
                         d = fcpA(memb,n) - fcpNEWA(memb,n);
                         if abs (d) < 0.1
                         break;
                         else
                         if d > 0
                         fcpA(memb,n) = fcpA(memb,n) - 0.01;
                         end
                         if d < 0
                         fcpA(memb,n) = fcpA(memb,n) + 0.01;
                         end
                         end
                    end
                 if FluxPredictA(memb,n)<0
                     FluxPredictA(memb,n)=0;
                     FluxPredictLMHA(memb,n)=0;
                 end
                 PermflowrateA(memb,n) = FluxPredictA(memb,n).*MembrA;%m/h*m2 =
m3/h
                 PermflowrateLSA(memb,n) =
PermflowrateA(memb,n).*1000./3600;%l/s
                 PermflowVOLA(memb,n) = PermflowrateA(memb,n).*1.*1000./3600;%
m3/h * h * 1000 = L in 1 sec
                 TotalPermVOLA(memb,n) = TotalPermVOLA(memb,n-1) +
PermflowVOLA(memb,n);%L
```

```
                CrossflowrateA(memb,n)=FeedflowrateA(memb,n)-
PermflowrateA(memb,n);%m3/h
                %Perm Conc
                SoluteFluxA(memb,n)=
(Ks./24).*fcpA(memb,n).*(FeedConcA(memb,n)./10^-3); %mg/h m2
                PermConcA(memb,n)=
SoluteFluxA(memb,n)./FluxPredictA(memb,n);%mg/m3
                if FluxPredictA(memb,n) == 0
                PermConcA(memb,n) = 0;
                end
                PermConcMGLA(memb,n) = PermConcA(memb,n)./1000;%mg/L
                CrossflowConcA(memb,n)=
((FeedflowrateA(memb,n).*(FeedConcA(memb,n)))-
(PermflowrateA(memb,n).*PermConcMGLA(memb,n)))./CrossflowrateA(memb,n);%m3/h
and mg/L
                if CrossflowrateA(memb,n) == 0
                CrossflowConcA(memb,n) = 0;
                end
                RejA(memb,n)= (1 -
(PermConcMGLA(memb,n)./FeedConcA(memb,n)))*100;
                TotalSaltMassA(memb,1)=0;

SaltMassA(memb,n)=PermConcMGLA(memb,n).*PermflowVOLA(memb,n);%mg/L * L
                TotalSaltMassA(memb,n)=TotalSaltMassA(memb,n-
1)+SaltMassA(memb,n);%mg
               end
%                   TotalPermVOLglobalA(n) = sum(PermflowVOLA(1:memb,n));%L
             TotalSaltMassglobalA(n)=sum(SaltMassA(1:memb,n));%mg
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%              PermVOLglobalA(n) = sum(PermflowVOLA(2:memb,n));%L
%              SaltMassglobalA(n)=sum(SaltMassA(2:memb,n));%mg
                TimeA(n)=n;
                TimeHRA(n)=TimeA(n)./3600;%hr
                TdA = n;
                PressurePVA(1,n)=PressurePVA(2,n);
        else
            PressurePVA(1,n)=PressurePVA(1,n-1);

            TotalSaltMassglobalA(n)=0;
          end
         n=n+1;
    end

    PressurePVC(1,TdC:n-1) = PressurePVC (1,TdC);
    TotalPermVOLglobalA = sum(PermflowVOLA(1:memb,:));
TotalPermeateVOLA = sum(TotalPermVOLglobalA);%L
TotalSaltMassA = sum(TotalSaltMassglobalA(1:n-1));%mg
cyclenum = cyclenum+1;

if n>length(windspeed) && PressurePVA(2,n-1)<= Pressurelim && PressurePVB(2,n-
1)<= Pressurelim && PressurePVC(1,n-1)<= Pressurelim
         flag=0;
            break;
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%% Fill PVC from 4800 to 6900 and PVB desal

    %%Fill PVC
   % if n<length(windspeed)
```

```
        n=TfB+1;
   % end

    %PressurePVC(1,TfA:TfB)=PressurePVC(1,TfA);
    NewAirVOLC(1,TfB)=nRT./PressurePVC(1,TfB);

    %% Desal PVB
    AirVOLB(2,TfB) = NewAirVOLB(1,TfB).*1000;%L
    PressurePVB(2,TfB)=PressurePVB(1,TfB);
    %AirVOLA(2,TfA)=(nRT./PressurePVA(1,TfA)).*1000;%Kpa.m3 / Kpa *1000= L
    NewfeedVOLB(2,TfB)=(InitialVolume.*1000)-AirVOLB(2,TfB);%L
    NewFeedConcB(2,TfB)=32000; %mg/L
    OsmoticPrKpaB(2,TfB)=(NewFeedConcB(2,TfB).* 8.505.*10^-2)- 86.61;%Kpa
    FluxPredictB(2,TfB)=Permcoeff.*(PressurePVB(2,TfB)-
OsmoticPrKpaB(2,TfB));%m/h
    PermflowrateB(2,TfB)=FluxPredictB(2,TfB).*MembrA;%m3/h

    FeedflowrateB(2,TfB)=PermflowrateB(2,TfB)./recoveryB;%m3/h
    FeedflowVOLB(2,TfB)=FeedflowrateB(2,TfB)./3600.*1000;%L
    PermflowVOLB(2,TfB)=PermflowrateB(2,TfB).*1000./3600;%L
    TotalPermVOLB(2,TfB)=PermflowVOLB(2,TfB);%L
    TotalSaltMassB(2,TfB)=0;
    TotalPermVOLB(2:membnum,TfB)=0;
    %
%     TotalSaltMassglobalC(TfB) = TotalSaltMassglobalC(TfB-1);%mg
%    TotalPermVOLglobalC(TfB)=TotalPermVOLglobalC(TfB-1);%L

%%%%%%  STEP %%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    while (PressurePVC(1,n-1) <= 6900 || PressurePVB(2,n-1)>Pressurelim)

        FluxPredictA(2:membnum, n)=0;
         TotalPermVOLglobalA(n)=0;
         if n>length(windspeed) && PressurePVB(2,n-1)<= Pressurelim
         break;
    end
        if PressurePVC(1,n-1) <= 6900 && n<=length(windspeed)

TurbinePower(1,n)=0.5.*Cpt.*airdensity.*rotorarea.*windspeed(1,n).*windspeed(1,
n).*windspeed(n);%w
            TurbinePowerKW(1,n)=TurbinePower(1,n)/1000;%kw
            PumpPower(1,n)=Pumpefficiency.* TurbinePowerKW(1,n);%kw
            PumpflowrateC(1,n)=PumpPower(1,n)./PressurePVC(1,n-1);%m3/s
            PumpflowVOLC(1,n)=PumpflowrateC(1,n);%m3 in 1 second
            NewAirVOLC(1,n)=NewAirVOLC(1,n-1)-PumpflowVOLC(1,n); %m3
            NewWaterVOLC(1,n)=InitialVolume-NewAirVOLC(1,n); %m3 CHANGE
            NewWaterVOLLC(1,n)=NewWaterVOLC(1,n).*1000; %L
%             PressurePVC(1,n)=PressurePVC(1,n-1).*NewAirVOLC(1,n-
1)./NewAirVOLC(1,n);%kpa
            PressurePVC(1,n)=nRT./NewAirVOLC(1,n);%Kpa.m3/m3=Kpa
            FillTimeC(1,n)=n;
            FillTimeHRC(1,n)=FillTimeC(1,n)./3600;
            TfC = n;
        else
            PressurePVC(1,n)=PressurePVC(1,n-1);
        end


        if PressurePVB(2,n-1)>Pressurelim
```

```
            CrossflowVELB(2,n)=FeedflowrateB(2,n-
1)./(3600.*W.*h.*0.88);%m3/s/(m*m) =m/s
            WallshearrateB(2,n)=3.*CrossflowVELB(2,n)./((h/2).*0.88);%m/s/m =
/s

masstransferB(2,n)=0.807.*((WallshearrateB(2,n).*D.*D)./L)^(1/3);%m/s
%              masstransferB(2,n)=0.807.*((3.*(FeedflowrateB(2,n-
1)./3600).*D.*D)./(2.*(h./2).*(h./2).*W.*L))^(1/3);%m/s
            NewfeedVOLB(2,n)= NewfeedVOLB(2,n-1)-FeedflowVOLB(2,n-1);%L
            NewFeedConcB(2,n) = 32000; %mg/L
            OsmoticPrKpaB(2,n) = (NewFeedConcB(2,n).* 8.505.*10^-2)- 86.61;
            AirVOLB(2,n) = InitialVolume.*1000 - NewfeedVOLB(2,n);%L
            dPB(2,n) = nRT.*1000.*((1./AirVOLB(2,n))- (1./AirVOLB(2,n-1)));% L
Kpa * (1/L) = Kpa
            PressurePVB(2,n) = dPB(2,n) + PressurePVB(2,n-1); % Kpa
            PressurePSIB(2,n)=PressurePVB(2,n)./6.9;

            fcpB(2,n) = 1.3;
           while d ~= 0
              FluxPredictB(2,n) = Permcoeff.*(PressurePVB(2,n) -
fcpB(2,n).*OsmoticPrKpaB(2,n)); %(m/Kpa h * Kpa = m/h)
              FluxPredictLMHB(2,n) = PermCoeff.*(PressurePVB(2,n) -
fcpB(2,n).*OsmoticPrKpaB(2,n));% lmh

              fcpNEWB(2,n) =
exp(FluxPredictB(2,n)./(masstransferB(2,n).*3600));
                    d = fcpB(2,n) - fcpNEWB(2,n);
                    if abs (d) < 0.1
                    break;
                    else
                    if d > 0
                    fcpB(2,n) = fcpB(2,n) - 0.01;
                    end
                    if d < 0
                    fcpB(2,n) = fcpB(2,n) + 0.01;
                    end
                    end
           end
            if FluxPredictB(2,n)<0
                FluxPredictB(2,n)=0;
                FluxPredictLMHB(2,n)=0;
            end
              PermflowrateB(2,n) = FluxPredictB(2,n).*MembrA;%m/h*m2 = m3/h
              PermflowrateLSB(2,n) = PermflowrateB(2,n).*1000./3600;%l/s
              PermflowVOLB(2,n) = PermflowrateB(2,n).*1.*1000./3600;% m3/h * h
* 1000 = L in 1 sec
              TotalPermVOLB(2,n) = TotalPermVOLB(2,n-1) + PermflowVOLB(2,n);%L
              FeedflowrateB(2,n)=PermflowrateB(2,n)./0.08;%m3/h
              FeedflowVOLB(2,n)=FeedflowrateB(2,n).*1000./3600;%L in 1 sec
              CrossflowrateB(2,n)=FeedflowrateB(2,n)-PermflowrateB(2,n);%m3/h
              %Perm Conc
              SoluteFluxB(2,n)= (Ks./24).*fcpB(2,n).*(NewFeedConcB(2,n)./10^-
3); %mg/h m2
              PermConcB(2,n)= SoluteFluxB(2,n)./FluxPredictB(2,n);%mg/m3
              if FluxPredictB(2,n) == 0
              PermConcB(2,n) = 0;
              end
              PermConcMGLB(2,n) = PermConcB(2,n)./1000;%mg/L
```

```
              CrossflowConcB(2,n)= ((FeedflowrateB(2,n).*(NewFeedConcB(2,n)))-
(PermflowrateB(2,n).*PermConcMGLB(2,n)))./CrossflowrateB(2,n);
              if CrossflowrateB(2,n) == 0
              CrossflowConcB(2,n) = 0;
              end
              RejB(2,n)= (1-(PermConcMGLB(2,n)./NewFeedConcB(2,n)))*100;
              %recoveryB(1,n)=(PermflowrateB(1,n)./FeedflowrateB(1,n)).*100;

                TotalSaltMassB(2,1)=0;
                SaltMassB(2,n)=PermConcMGLB(2,n).*PermflowVOLB(2,n);%mg/L * L
                TotalSaltMassB(2,n)=TotalSaltMassB(2,n-1)+SaltMassB(2,n);%mg
                %%%%%          Membranes 2,3,4,5          %%%%%%%%%%%%%%%%%%%%
                %Number of membrane elements =5
               for memb=3:membnum
                PressurePVB(memb,n)=PressurePVB(memb-1,n)-13.8;
                PressurePSIB(memb,n)=PressurePVB(memb,n)./6.9;
                FeedflowrateB(memb,n)=CrossflowrateB(memb-1,n);%m3/h
                FeedflowVOLB(memb,n)=FeedflowrateB(memb,n).*1000./3600;%L in 1
sec


CrossflowVELB(memb,n)=FeedflowrateB(memb,n)./(3600.*W.*h.*0.88);%m3/s/(m*m)
=m/s

WallshearrateB(memb,n)=3.*CrossflowVELB(memb,n)./((h/2).*0.88);%m/s/m = /s

masstransferB(memb,n)=0.807.*((WallshearrateB(memb,n).*D.*D)./L)^(1/3);%m/s
%                                  FeedConcB(memb,n)=CrossflowConcB(memb-
1,n);%mg/L
                OsmoticPrKpaB(memb,n)=8.505.*10^-2.*FeedConcB(memb,n)-86.61;
                fcpB(memb,n) = 1.3;
                       while d ~= 0
                          FluxPredictB(memb,n) =
Permcoeff.*(PressurePVB(memb,n)-fcpB(memb,n).*OsmoticPrKpaB(memb,n)); %(m/Kpa h
* Kpa = m/h)
                          FluxPredictLMHB(memb,n) =
PermCoeff.*(PressurePVB(memb,n)-fcpB(memb,n).*OsmoticPrKpaB(memb,n));% lmh
                          if FluxPredictB(memb,n)<0
                            FluxPredictB(memb,n)=0;
                            FluxPredictLMHB(memb,n)=0;
                          end
                            fcpNEWB(memb,n) =
exp(FluxPredictB(memb,n)./(masstransferB(memb,n).*3600));
                            d = fcpB(memb,n) - fcpNEWB(memb,n);
                           if abs (d) < 0.1
                                break;
                           else
                            if d > 0
                                fcpB(memb,n) = fcpB(memb,n) - 0.01;
                            end
                            if d < 0
                                fcpB(memb,n) = fcpB(memb,n) + 0.01;
                            end
                            end
                       end

                    PermflowrateB(memb,n)=FluxPredictB(memb,n).*MembrA;%m/h*m2
= m3/h
```

```
PermflowrateLSB(memb,n)=PermflowrateB(memb,n).*1000./3600;%l/s
                    PermflowVOLB(memb,n)=PermflowrateB(memb,n).*1.*1000./3600;%
m3/h * h * 1000 = L in 1 sec
                    TotalPermVOLB(memb,n) = TotalPermVOLB(memb,n-1) +
PermflowVOLB(memb,n);%L
                    CrossflowrateB(memb,n)=FeedflowrateB(memb,n)-
PermflowrateB(memb,n);%m3/h
                    %Perm Conc
                    SoluteFluxB(memb,n)=
(Ks./24).*fcpB(memb,n).*(FeedConcB(memb,n)./10^-3); %mg/h m2
                    PermConcB(memb,n)=
SoluteFluxB(memb,n)./FluxPredictB(memb,n);%mg/m3
                    if FluxPredictB(memb,n) == 0
                        PermConcB(memb,n) = 0;
                    end
                        PermConcMGLB(memb,n) = PermConcB(memb,n)./1000;%mg/L
                        CrossflowConcB(memb,n)=
((FeedflowrateB(memb,n).*(FeedConcB(memb,n)))-
(PermflowrateB(memb,n).*PermConcMGLB(memb,n)))./CrossflowrateB(memb,n);
                    if  CrossflowrateB(memb,n) == 0
                        CrossflowConcB(memb,n) = 0;
                    end
                        RejB(memb,n)= (1-
(PermConcMGLB(memb,n)./FeedConcB(memb,n)))*100;
                        TotalSaltMassB(memb,1)=0;

SaltMassB(memb,n)=PermConcMGLB(memb,n).*PermflowVOLB(memb,n);%mg/L * L
                        TotalSaltMassB(memb,n)=TotalSaltMassB(memb,n-
1)+SaltMassB(memb,n);%mg
               end


%              TotalPermVOLglobalB(n) = sum(PermflowVOLB(1:memb,n));%L
             TotalSaltMassglobalB(n)=sum(SaltMassB(1:memb,n));%mg
          %%%%%%%%%%%%%%%%%%
%              PermVOLglobalB(n) = sum(PermflowVOLB(2:memb,n));%L
%              SaltMassglobalB(n)=sum(SaltMassB(2:memb,n));%mg
                    TimeB(n)=n;
                    TimeHRB(n)=TimeB(n)./3600;%hr
                    TdB = n;
                    PressurePVB(1,n)=PressurePVB(2,n);
               else
                    PressurePVB(1,n)=PressurePVB(1,n-1);
%                     TotalSaltMassglobalB(n) = TotalSaltMassglobalB(n-1);%mg
                    TotalSaltMassglobalB(n) =0;
%
          end
%
          n=n+1;
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
     PressurePVA(1,TdA:n-1) = PressurePVA (1,TdA);
    TotalPermVOLglobalB = sum(PermflowVOLB(1:memb,:));
     TotalPermeateVOLB = sum(TotalPermVOLglobalB);%L
TotalSaltMassB = sum(TotalSaltMassglobalB(1:n-1));%mg
cyclenum = cyclenum+1;
if n>length(windspeed) && PressurePVA(2,n-1)<= Pressurelim && PressurePVB(1,n-
1)<= Pressurelim && PressurePVC(2,n-1)<= Pressurelim
```

```
        flag=0;
           break;
end

n=TfC+1;

TotalPermeateVOLA=0;
TotalPermeateVOLB=0;
TotalPermeateVOLC=0;
TotalSaltMassA=0;
TotalSaltMassB=0;
TotalSaltMassC=0;
counter=counter+1;
end
TotalPermeateVOLA = sum(TotalPermVOLglobalA);%L
TotalPermeateVOLB = sum(TotalPermVOLglobalB);%L
TotalPermeateVOLC = sum(TotalPermVOLglobalC);%L
TotalPermeateVOLABC =
TotalPermeateVOLABC+TotalPermeateVOLA+TotalPermeateVOLB+TotalPermeateVOLC;
TotalSaltMassA = sum(TotalSaltMassglobalA);%mg
TotalSaltMassB = sum(TotalSaltMassglobalB);%mg
TotalSaltMassC = sum(TotalSaltMassglobalC);%mg
TotalSaltMassABC=TotalSaltMassABC+TotalSaltMassA+TotalSaltMassB+TotalSaltMassC;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n=n-1;
TotalTime=1:n;%total cycle time in seconds
TotalTimeHR=(1:n)./3600;

%%Calculation of Specific Energy
TurbinePowerKW(1,1:length(windspeed))=0.5.*Cpt.*airdensity.*rotorarea.*windspee
d(1,1:length(windspeed)).*windspeed(1,1:length(windspeed)).*windspeed(1,1:lengt
h(windspeed))./1000;%kw
TotalTurbinePowerKW=sum(TurbinePowerKW(1,1:length(windspeed)));
AvgTurbinePowerKW = TotalTurbinePowerKW/n;%KW
TotalPermeateflowrate = TotalPermeateVOLABC./(1000*n/(3600*24));%m3/day
SpecificEnergy = AvgTurbinePowerKW*24/(TotalPermeateflowrate);%KW*hr/m3

PermeateConcentration=TotalSaltMassABC./TotalPermeateVOLABC;%mg/L
fprintf('\n Total Permeate Volume=%f L \n',TotalPermeateVOLABC);

fprintf('\n Total Permeate flow rate=%f m3/d \n',TotalPermeateflowrate);

fprintf('\n Specific Energy=%f KWh/m3 \n',SpecificEnergy);

fprintf('\n Permeate water concentration=%f mg/L \n',PermeateConcentration);
Rejection=(1-(PermeateConcentration./32000)).*100;
fprintf('\n Rejection=%f % \n ',Rejection);
fprintf('\n Number of cycles=%f \n',cyclenum);
%%%Printing the results again so that I can copy and paste in excel directly
fprintf('\n %f',TotalPermeateVOLABC);
fprintf('\n %f',TotalPermeateflowrate);
fprintf('\n %f',PermeateConcentration);
fprintf('\n %f',Rejection);
fprintf('\n %f',SpecificEnergy);
fprintf('\n %f',cyclenum);

figure;
plot(TotalTimeHR(1:n), PressurePVA(1,1:n),'.')
```

```
hold on;
plot(TotalTimeHR(1:n), PressurePVB(1,1:n), 'k.')
hold on;
plot(TotalTimeHR(1:n),PressurePVC(1,1:n), 'g.')
xlabel('Time (Hr)')
ylabel('Pressure (Kpa)')
```