Variable-Speed and Multi-Mode Solar Assisted Heat Pumps: System Design and Controls Development

by

Julian Craig Peter Howarth

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Doctor of Philosophy

in

Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

- Michael Collins (Supervisor), Mechanical and Mechatronics Engineering, University of Waterloo
- Xianguo Li, Mechanical and Mechatronics Engineering, University of Waterloo
- Soo Jeon, Mechanical and Mechatronics Engineering, University of Waterloo
- Mohamad Araji, School of Architecture, University of Waterloo
- Cynthia Cruickshank (External), Mechanical and Aerospace Engineering, Carleton University

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Julian Howarth is the sole author for chapters 1, 2, 3, 6, 7, and 8 which were written under the supervision of Dr. Michael Collins and first published in this thesis. This thesis also consists in part of three manuscripts written for publication. Exceptions to the sole authorship of material are as follows:

Research Presented in Chapters 5 and 6

Material presented in chapters 5 and 6 consists of text and figures previously published in conference proceedings with Julian Howarth listed as the first author and Dr. Michael Collins (Supervisor) listed and the 2nd author.

The research was conducted at the University of Waterloo by Julian Howarth, who designed and conducted the experiments, analyzed and reported on the data and results. Draft manuscripts were prepared by Julian Howarth and reviewed by Dr. Michael Collins. As is convention, the supervisor is listed as a co-author of the work in recognition of their support in the preparation of the manuscripts.

Permission to reprint these materials has been obtained from the publisher and a letter of permission is included in this document.

Citations:

Chapter 5:

J. Howarth and M. Collins, "Experimental Characterization of a Variable Refrigerant Flow Heat Pump for Solar-Domestic-Hot-Water Applications," in 5-6th Thermal and Fluids Engineering Conference (TFEC), 2021.

J. Howarth and M. Collins, "3-Stage Control of a Variable-Refrigerant-Flow Heat Pump for Solar Domestic Hot Water Applications," in 7th Thermal and Fluids Engineering Conference (TFEC), Las Vegas, NV, 2022.

Chapter 6:

J. C. Howarth and M. R. Collins, "Annual performance of a simulated multi-mode SDHW/SAHP system with various control options," in 8th Thermal and Fluids Engineering Conference (TFEC), Maryland, MD, 2023.

Abstract

In an era characterized by increasing energy cost and frequent reminders of the climate impact of emissions from combustion, consumers and energy regulators have a heightened interest in solar and other renewable energy sources. This thesis details work that was undertaken to investigate the performance of a Solar-Assisted-Heat-Pump system (SAHP) for Domestic Hot Water (DHW) made novel by the incorporation of a variable capacity heat pump constructed using a 3-phase scroll compressor whose speed can be modulated by a variable frequency drive. The overall purpose of the system being investigated is to meet the DHW load demands of a single-family household, while reducing the annual purchased energy and thereby reducing operating cost and emissions associated with the residence's DHW consumption.

A key characteristic of the system under study is the variable-speed Heat Pump (HP) which is of custom construction for the current research. A modified factorial experiment was designed and completed to characterize and model the HP's performance under source and load inlet temperatures and flow rates typical of a mains-connected SAHP system. The resulting multivariate polynomial expressions for HP compressor work rate, source-side heat transfer rate, and load-side heat transfer rate were programmed into a custom component "TYPE" model for TRNSYS, a transient system simulation software package suited to thermal systems. This work represents an improvement on the default HP model in TRNSYS which does not offer the required flexibility to modulate the compressor speed of the HP with a continuously variable input. Validation of the HP model was performed with a separate set of data from those used to fit the model.

A multi-mode SAHP system was modeled in TRNSYS to match an Experimental Test Unit (ETU) housed in the Solar Thermal Research Lab (STRL) at the University of Waterloo (UW). Components of the ETU are commercially available Solar DHW tanks and hydronic heating components. In parallel, a whole system TRNSYS model and a physical system representing a multi-mode variable-capacity SAHP were constructed with model components configured to match their analogous physical components. The TRNSYS model was validated at the component level for the heat pump, Heat Exchanger (HX), Auxiliary Electric Resistance Heaters (AUX), and storage tank. The model was then validated as a whole-system operating over day-long trials under a simplified control regime. Daily validation trials showed agreement between simulated and experimental results.

Annual performance of a variety of configurations, under a temperature-based control scheme consistent with other studies in the literature were studied. The results of these annual simulations showed some performance benefit of the system under study, but highlighted the need for a more advanced control strategy that would make better use of the variable-capacity HP, and correctly decide when the HP should be used over the HX. Poorer performance of the SAHP system than expected was consistent with other studies findings in the literature that also called for more advanced controls.

A Predictive Controller for the variable-SAHP was developed using MATLAB and TRNSYS. The controller works through iterative calls to the TRNSYS system model, the results of which feed into a time-series of control signals that the controller stores and feeds back to the system being operated. Annual simulations were conducted using a top-level TRNSYS simulation in place of the system being operated, and through a MATLAB-link, a separate instance of TRNSYS was used for the iterative sub-simulations. These simulations showed a marked improvement in the performance of the system under the new predictive control scheme compared with simulations of temperature-based control. This improved performance is taken to represent an approximation of the maximum performance of the SAHP system being studied because the predictive controller has selected the optimum control series for the system under perfect simulation conditions. It is acknowledged that in order to maintain realistic performance predictions from the annual TRNSYS simulation, future work is needed to address how the controller would handle model prediction error when controlling a real SAHP system.

As a final verification and demonstration of the work, the new controller created to control TRNSYS simulations was ported to an instance of LabVIEW running on the ETU. The controller was implemented to operate the equipment in the lab as a form of Hardware-In-The-Loop (HIL) simulation. This exercise demonstrates that the concept of predictive control as implemented in this work is capable of controlling a real system under study with the goal of meeting DHW demands. Some disagreement was noted between simulation and experimental operation of the system and explained within the context of limitations of the ETU to reproduce certain losses, and model timing errors that can lead to missed milestones for collection on some poor solar days. A number of suggestions are offered to address the shortcomings uncovered by these verification trials. These suggestions included model improvements, and changes to the controller itself that would make it more robust and capable of dealing with variation between model inputs and the observed conditions.

Acknowledgments

The author wishes to acknowledge the following individuals who have provided support, both directly and indirectly, to the project:

- The author extends thanks to Mike Collins for the years of thoughtful guidance and support as supervisor.
- Within the Solar thermal research lab, the author is grateful for the company and advice of fellow graduate students including Ramin Manouchehri, Ajmeet Dhillon, and Chris Chasse.
- Technical support with instrumentation and LabVIEW programming was provided by Neil Griffett, Electronics Technician, Mechanical and Mechatronics Engineering (MME).
- Software, computing, and network connectivity support, which were instrumental to continuing research during campus shutdowns was dutifully provided by Mike Willson, Mitchell Forsyth, and Martha Morales, Information Technology Specialists, MME.
- In the classroom, relevant topics to the research project were learned from professors: Kyle Daun, David Mather, Mike Collins, Serhiy Yarusevych, John Straub, and Ibrahim Alaefour

In addition, some material presented in this thesis appears in three works published in sequential conference proceedings of the American Society of Thermo-Fluids Engineers. The author extends thanks to the American Society of Thermo-Fluids Engineers, their reviewers, and to the publisher, Begell House Inc. for providing permission to reprint these materials.

Dedication

With thanks for the years of support and encouragement that they have offered, this thesis is dedicated to: my partner Natalie Frixione, my mother Mary Ellen Starodub, my father Frank Howarth, and my late stepfather Dr. Colin Mayfield.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation:	Meaning:
AUX	Auxiliary Heater
COP	Coefficient of Performance
CSA	Canadian Standards Association
CV	Control Volume
DAQ	Data Acquisition
DHW	Domestic Hot Water
DWHR	Drain Water Heat Recovery
ETU	Experimental Test Unit
GEPS	Global Ensemble Prediction System (Weather Forecast)
GHG	Green House Gas
HIL	Hardware in the Loop
HP	Heat Pump
HPWH	Heat Pump Water Heater (Air Source)
HWB	Hottel-Whillier-Bliss Equation
HX	Heat Exchanger
IEA	International Energy Agency
IPCC	International Panel on Climate Change
KPI	Key Performance Indicators
ML	Machine Learning
MPC	Model Predictive Control
NRCan	Natural Resources Canada
PCM	Phase Change Material
PV	Photovoltaics
PV/T	Photovoltaic-Thermal
RDPS	Regional Deterministic Prediction System (Weather Forecast)
RMSE	Root-Mean-Squared Error
SAHP	Solar Assisted Heat Pump
SDHW	Solar Domestic Hot Water
SH	Space Heating
SHC	Solar Heating and Cooling
STC	Solar Thermal Collector
STRL	Solar Thermal Research Lab
TMY	Typical Meteorological Year
UW	University of Waterloo
VFD	Variable Frequency Drive

N.B. Abbreviations used only in the appendix are not listed.

List of Symbols

Symbol:	Meaning:	Typical Units
A	Cross-sectional area of DHW tank	m^2
A_c	Gross collector area	m^2
C subscripts:	Heat capacity rate of denoted flowing stream (HX)	kJ/s ℃
min, max, c, h	Minimum, Maximum, cold, hot	
COP_h	Coefficient of Performance of the heat pump	-
c_p	Specific heat capacity of a fluid	kJ/kg ℃
C_r	Ratio of minimum and maximum capacity rates (HX)	-
$E_{available}$	Energy content of water in DHW tank held above 35°C	kJ
$E_{collected,Solar}$	Annual total solar energy collection	kJ, MJ
$E_{deficit}$	Draw Energy Deficit	kJ
ΔE_{HP}	Energy difference per timestep if HP is operated	kJ
ΔE_{HP-HX}	Energy difference between HX and HP per timestep	kJ
ΔE_{HX}	Energy difference per timestep if HX is operated	kJ
ΔE_{stored}	Annual change in energy stored	kJ, MJ
$E_{DHW\ charge}$	Quantity of energy stored in DHW tank	kJ
$\dot{E}_{DHW\ charge}$	Energy storage rate of the DHW tank	kJ/h
$E_{DHW\ tank}$	Energy content of all water in DHW tank	kJ
$E_{draw,delivered}$	Annual total energy delivery in DHW draws	kJ, MJ
E_{loss}	Annual total system energy losses	kJ, MJ
$E_{purchased}$	Annual total purchased energy	kJ, MJ
E_{solar}	Quantity of solar energy stored in the DHW tank	kJ

Symbol:	Meaning:	Typical Units
$E_{unavailable}$	Energy content of water in DHW tank held below 35°C	kJ
f	Frequency	Hz
$\mathcal{F}$	Solar Fraction on an annual basis	-
f_i	Solar Fraction on a monthly (i) basis	-
F_R	Heat removal factor (collector configuration specific)	-
G_t	Flux rate of solar radiation incident on a tilted plane	W/m^2, kJ/h m^2
I	Electrical Current	A, mA
k	Index number of a timestep	-
k_{sys}	Proportionality constant of piping system	-
L subscripts: S, A	Load Energy (subscripts refer to solar and AUX contribution)	kJ, MJ, kWh
$\dot{m}$ subscripts: $source, load$	Mass flow rate of designated flow stream (HP)	kg/s, kg/min
N_{comp}	Compressor Speed Signal (HP)	%
NTU	Number of Transfer Units (HX)	
ΔP	Pressure Difference	Pa, kPa
q_{act}	Actual or operating heat transfer rate (HX)	W, kW, kJ/h
$\dot{Q}$ subscripts: $cond, load,$ l	Heat transfer rate at condenser/load side (HP)	W, kW, kJ/h
$\dot{Q}$ subscripts: $evap, source, s$	Heat transfer rate at evaporator/source side (HP)	W, kW, kJ/h
q_{max}	Theoretical maximum heat transfer rate (HX)	W, kW, kJ/h
$\dot{Q}_{loss}$	Rate of stray energy transfer to surroundings	W, kW, kJ/h

Symbol:	Meaning:	Typical Units
Q_u	Rate of solar energy collection (useful heat transfer)	W, kW, kJ/h
R^2	Coefficient of Determination (Statistics of Fit)	-
$RMSE$	Root-Mean-Squared Error	-
S	Flux rate of absorbed solar radiation at collection surface	W/m^2, kJ/h m^2
t	Time	h, min, s
T	Temperature	°C
ΔT	Temperature Difference	°C
T_a	Ambient air temperature	°C
$T_{deadband}$	Deadband temperature for temperature-based control	°C
T_{DHW}	Storage temperature of water in DHW tank (additional subscripts indicate location within tank)	°C
$T_{HP,cutin}$	HP activation temperature for temperature-based control	°C
T_i	Inlet fluid temperature (STC)	°C
$T_{l,i}$	Load Temperature Inlet (HP)	°C, K
$T_{l,o}$	Load Temperature Outlet (HP)	
$T_{o,col}$	Collector Outlet temperature	°C
T_{set}	Temperature setpoint for temperature-based control	°C
$T_{s,i}$	Source Temperature Inlet (HP)	°C, K
$T_{s,o}$	Source Temperature Outlet (HP)	
U	Velocity of a flowing fluid	m/s
UA	Overall Heat Transfer Coefficient (HX)	W/°C, kJ/h °C
U_L	Thermal loss coefficient for (STC)	W/m^2 °C

Symbol:	Meaning:	Typical Units
V (Electrical)	Voltage	V
V (Fluid)	Volume	L, m^3
$\dot{V}$	Volumetric Flow Rate	L/min
$\dot{W}_{el}$	Electric work rate or electric power consumption (HP)	W, kW, kJ/h
W_{el}	Quantity of energy from HP compressor work stored	kJ
z	Distance in the vertical dimension	m
α_s	Absorptivity of collection surface in the solar spectrum	- or %
ε	Effectiveness (HX)	%
η_{coll}	Collection efficiency (STC)	%
η_{motor}	Efficiency of electric motor	%
η_{pump}	Mechanical efficiency of a pump	%
$\eta_{Wire-to-Water}$	Efficiency of a motor and pump combined	%
ρ	Density of a fluid	kg/m^3

N.B. Symbols used only in the appendix are not listed.

Chapter 1
Introduction

Renewable energy research is driven by a multitude of social, economic, and environmental pressures. Presently, there is interest in clean and renewable energy technologies because they are seen as a potential solution to help meet growing energy demands, and to offset increasing energy costs while producing fewer Green House Gas (GHG) emissions, which contribute to global climate change. The current work deals with design and controls development for domestic hot water applications utilizing a solar-assisted variable-speed heat pump at the core of the system.

1.1 Motivations

1.1.1 Climate Change and Public Climate Policy

Reference to the greenhouse effect has become ubiquitous in environmental and scientific writing on the topics of climate change [1], and renewable energy systems [2, 3]. It is well-established that the concentration of carbon dioxide, being a major GHG, is linked to the magnitude of the greenhouse warming effect. Consistent in all climate change reporting are predictions of increased severe weather activity and natural disaster frequency, higher sea-levels, loss of wildlife habitat, and strain on human infrastructure [1]. As such, it is a laudable goal to take steps to arrest or at least slow the rate of climate change.

As part of its obligations arising from the 2015 Paris agreement, which seeks to limit the global temperature rise to 1.5°C or 2°C, Canada has pledged to reduce its GHG emissions by 30% from 2005 levels by the year 2030 [4]. The intention is to spread the reduction in GHG emissions across all sectors of the economy. This will no-doubt include the residential and commercial buildings sectors.

As will be shown, domestic hot water production represents a significant percentage of building energy use, and by association, GHG emissions. It can therefore be stated that the development of renewable and clean energy technologies for domestic hot water production is congruent with the current climate policy in Canada and internationally. Adopting such technologies in new and existing housing stock is one of many incremental steps that can be taken towards the reduction of GHG emissions and stabilization of the climate.

1.1.2 Rising Energy Costs

Energy prices, which are constantly on the rise in Canada and the rest of the world, are another motivating factor for research and investment in the renewable and alternative energy sector. Electricity and natural gas are two major utilities that act as energy sources to meet the demands of the residential and commercial sectors. Figure 1-1 shows the energy price indices associated with electricity and natural gas from 1980 to present (natural gas data unavailable after 2016) as reported by Statistics Canada [5, 6] The indexed prices are based on the total reported revenues of energy utility companies, divided by the reported total quantity of energy sold, then normalized to an index with 100 representing the sales-price value of each in the year 2014. The lines on the chart below are a 12-month moving average representation of the energy price indices, which are not adjusted for seasonal trends.

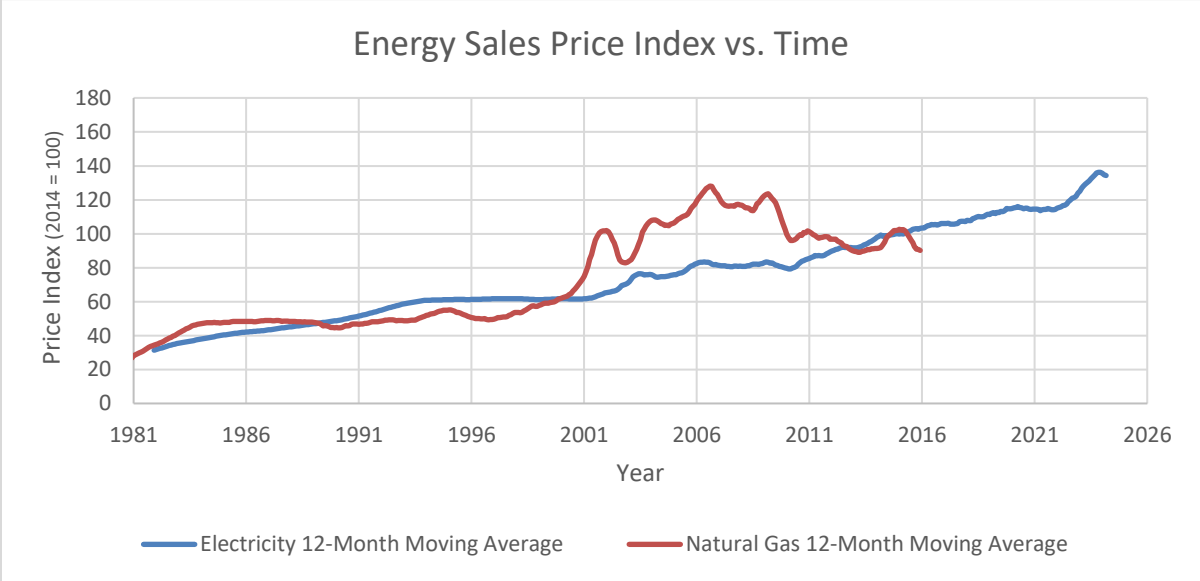
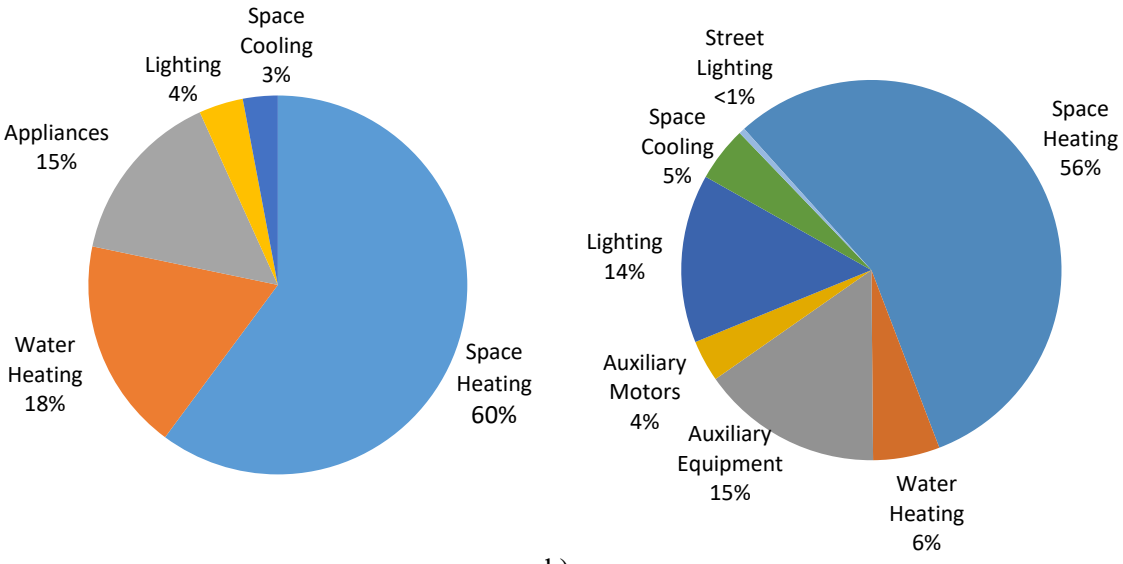


Figure 1-1: Energy sales price index vs. time for electricity and natural gas sold in Canada from 1981 to 2024. Note, natural gas data unavailable after 2015. [5, 6]

From the figure above, although there appears to be more volatility in the price of natural gas in the period since the year 2000, this movement is superposed on an overall upward trend that both utilities exhibit. In particular, the energy prices have, on average, been growing at a higher rate in the two decades since the mid-2000s than they were for the two-decades before. Assuming that the market prices of energy continue to trend as they have, an economic case can be made for further development of high-efficiency, renewable and alternative energy systems.

1.1.3 Energy Use and GHG Emissions in Canada

Annual secondary energy use in Canada had been increasing up to the year 2014, where it peaked at approximately 8,961.40 PJ. Since that time, small reductions in secondary energy use have led to an annual total consumption of 8,786.40 PJ [7]. Of that figure, more than one quarter of secondary energy is used between the residential and commercial/institutional sectors. Combined, the residential and commercial sectors consume 2,455.60 PJ per year and cause GHG emissions of 105.5 Mt of CO_2 equivalent emissions [7]. The energy consumed by residential and commercial sectors can be divided by end-use, which can be seen in Figure 1-2.



a) b)

Figure 1-2: Residential (a) and Commercial (b) energy consumption by end-use [7]

As shown in Figure 1-2, water heating is the second largest energy consumer in residential buildings and a significant end-use in commercial applications, representing 18% and 6% of energy use respectively. The current work is focused on water heating which, based on the figures above, consumes approximately 363.3 PJ of energy and contributes 16.9 Mt of CO_2 equivalent emissions per year in Canada [7]. There is room for significant energy savings and GHG emissions reductions by improving water heating methods across the country.

1.2 Solar Renewables Background

1.2.1 The Solar Resource

The solar resource is an attractive renewable energy source because it is freely available world-wide and because solar radiation can be readily converted into heat using a Solar Thermal Collector (STC), or into electricity using Photovoltaics (PV). Energy is transferred from the Sun to the Earth by radiation. The total annual quantity of solar energy incident on the Earth is approximately 5.4×10^{24} J/year [8]. This quantity is on the order of 14 thousand times greater than the total global energy consumption reported in 2016 [9]. With the exception of the initial capital outlay for manufacturing and installation of solar collection equipment, solar energy can systems operate with little to no operating cost, and zero GHG emissions.

Solar resource availability data provided by Natural Resources Canada (NRCan) are displayed in Figure 1-3 [10]. It can be seen in the figure, that the level of insolation varies widely across Canada from an average daily value of upwards of 18 MJ/m^2 day in the southern center of the country to half of that at as low as 9 MJ/m^2 day in the far north and on the west coast. It can also be seen, that the majority of the urban centers in the country including Calgary, Regina, Winnipeg, Toronto, Ottawa, and Montreal exist in regions that receive an average of above 15 MJ/m^2 day of solar energy. The country stands to benefit from utilizing the solar resource to abate the use of other energy sources and reduce GHG emissions.

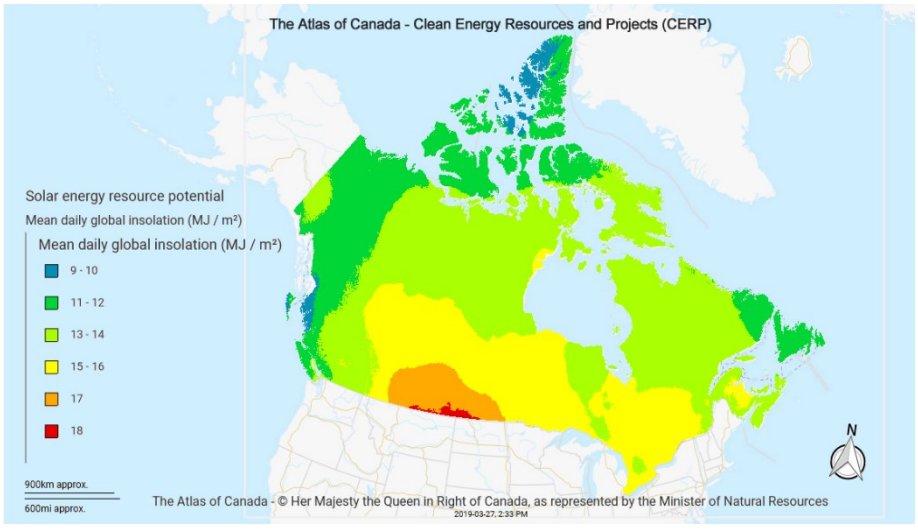


Figure 1-3: Heat map showing solar energy resource potential (adapted from [10])

1.2.2 Solar-Thermal vs. Solar Photovoltaic

The focus of this work is on solar-thermal technologies for water heating applications. While "solar heating" could be achieved through a PV system coupled with an electric resistance element, such a system would be limited by the PV energy collection efficiency which for common silicon cells is often in the range of 20 to 25%, and is theoretically limited to a maximum of 33% efficiency [11]. Even the most efficient research-grade PV cells can only achieve a conversion efficiency of 44 to 46% in generating electricity [12]. On the other hand, when operated in the correct temperature range, STCs are capable of much higher collection efficiencies. Depending on the design of the collector, the theoretical efficiency limit can be above 95%, and collectors can commonly achieve efficiencies between 60 and 80% [13]. The higher collection efficiencies available in STCs over PV cells make them a logical choice for thermal applications such as water-heating.

1.2.3 Solar-Thermal Collectors (STC)

The STC is the element in a solar-thermal system that receives solar radiation and converts the energy into a useful output. In the case of a Solar Domestic Hot Water (SDHW) system, that output is hot water, or some other heat transfer fluid, like an antifreeze. A variety of STC designs have been developed for different applications. Figure 1-4 below shows typical collector performance curves for various designs of collectors [14]. Figure 1-5 contains sample illustrations of three main types of collector design.

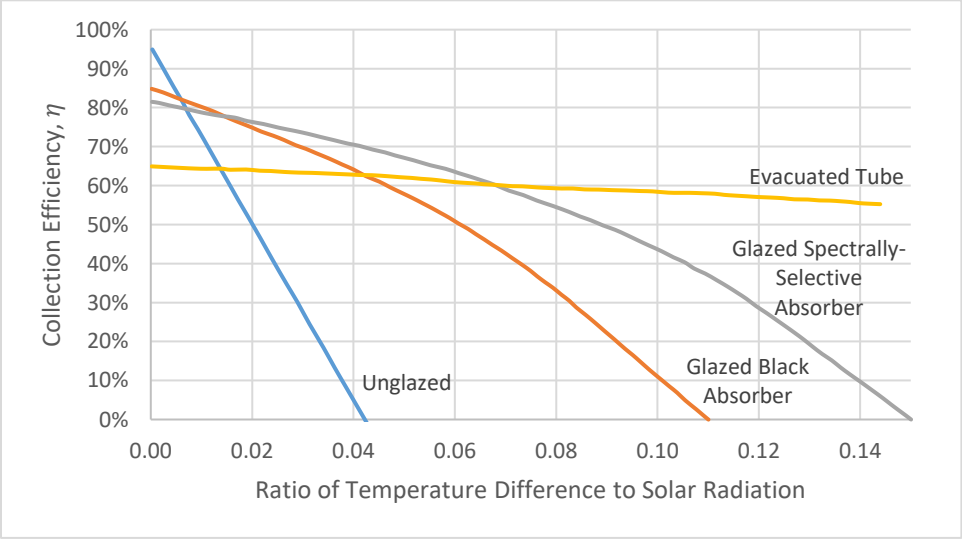


Figure 1-4: Typical STC performance curves for various types of collectors [14]

As can be seen in Figure 1-4, unglazed collectors perform well under low temperature differences between the ambient air and the fluid within the collector and at high solar irradiation levels. This can be explained by their lack of insulation, leading to heat losses as the collector gets hotter than the ambient conditions [13]. Evacuated tube collectors perform relatively consistently at around 65% efficiency regardless of the temperature difference between the collector and the ambient air. These collection tubes are constructed with a glass envelope and gasses that surround the collection plate and tubes are vacuumed from the enclosure before the tube is sealed. Poorer performance of evacuated tube collectors in low ΔT conditions is explained by the added layer of glass, which absorbs and reflects some of the solar radiation and reduces the amount that reaches the collector plate. The higher ΔT performance, however, is boosted by eliminating the convective heat losses from the collector plate to the surroundings [14]. The third and final collector type presented in Figure 1-5 is a glazed flat-plate collector. As seen in Figure 1-4 for both the "Black" and the "Selective" absorber, the performance of a glazed collector is a trade-off between the low-temperature advantages of an unglazed plate, and the high-temperature advantages of the evacuated tube. In a glazed flat-plate collector, back-losses are mitigated by including an insulation material while top-losses are reduced by including the cover glass above the absorber plate to reduce convective and radiative losses [14]. Glazed flat-plate collectors offer better general thermal performance than their unglazed counterparts. They are more economical to construct, and offer slightly better low-end performance than the evacuated tube option [14]. The collector selected for use in the system under current study will be a glazed flat-plate collector.

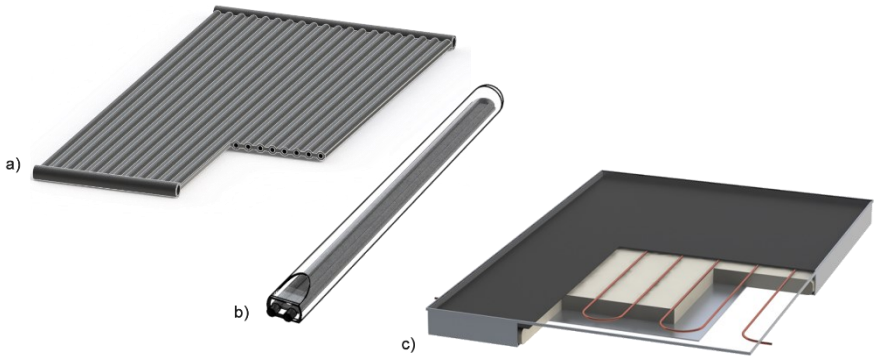


Figure 1-5: Three common STC designs; a) Unglazed black plate collector; b) Evacuated Tube Collector; and c) Glazed flat-plate Collector

1.2.4 Thermal Storage

One of the challenges with making use of solar energy is its time-varying intensity and intermittent availability. Loads on a solar-thermal system also vary in time based on the activities and schedules of the end-user and these load schedules are generally not synchronized or consistently equal to the rate of energy collection.

On a minute-to-minute basis, clouds can cast shadows onto the collection system and reduce or eliminate the amount of incident radiation. On the shortest time-scale, momentary interruptions to solar irradiation can be buffered by adding thermal capacity to the collection side of the system. This is important when momentary interruptions to the rate of collection are not acceptable, for instance, when a heat pump is being used and would freeze out or cycle off if the collector temperature dropped too low. This has previously been addressed by the addition of a small, well-mixed buffering tank that smooths the temperature collection profile [2, 3].

On a diurnal basis, the resource is available only from dawn to dusk, with overnight periods of darkness. Storage employed on a longer time scale can be used to collect solar energy during the day while the insolation levels are high, and carry it forward for use in the evening or night time. Thermal storage in an SDHW system naturally uses the water in a Domestic Hot Water (DHW) tank as the storage medium [14].

In SDHW storage tanks, stratification is an important factor that plays into the performance of the system. A highly stratified tank is able to deliver water from the top of the tank at a useful temperature to the drawing load, while sending relatively colder water out to the STC, allowing it to operate at a higher efficiency and collect energy at a higher rate. In a common tank porting arrangement, the heated water from the STC is returned high enough in the tank such that it does not fully mix with the cold strata of water at the bottom, but also low enough that if the water at the top of the tank is hotter than the water returning from the collector, they will remain unmixed [14].

Other diurnal storage technologies, such as solar walls, packed-beds, and phase changing materials have been developed and investigated as well. These technologies lend themselves well to space heating/cooling systems and less-so to SDHW [14]. As these systems are not the focus of the current work, they will not be described any further here.

Finally, on a seasonal basis, the intensity of solar radiation at the surface of the earth varies because the change in average position of the sun in the sky causes radiation to pass through a longer or

shorter attenuating path through the atmosphere. In larger systems, such as those used for solar district heating, ground-source heat exchangers can be installed to sequester excess solar energy during warmer seasons, and extract that heat during cooler months [14]. Again, because these systems are not the focus of the current work, they will not be described further.

1.2.5 Solar Domestic Hot Water Systems (SDHW)

Traditional SDHW systems can be divided into passive (thermosiphon) and active (pumped) systems. The most basic of passive systems is pictured in Figure 1-6. In a thermosiphon system, the natural variation in the density of water with temperature leads to buoyant forces and natural convection within the system. Hot water in the collector flows toward the top of the stratified storage tank, and cold water from the bottom of the storage tank flows toward the collector [16]. A disadvantage of these systems is that the storage tank must be located at the same level or above the collector, so that the hot water can rise away from the collector and draw new cold water in to be heated. A common installation location for solar-systems is on the rooftop. In climates such as Canada's, where ambient temperatures often fall below freezing, rooftop hot-water tanks would experience tremendous heat-loss and suffer from poor performance, if not damage, over the fall, winter, and spring.

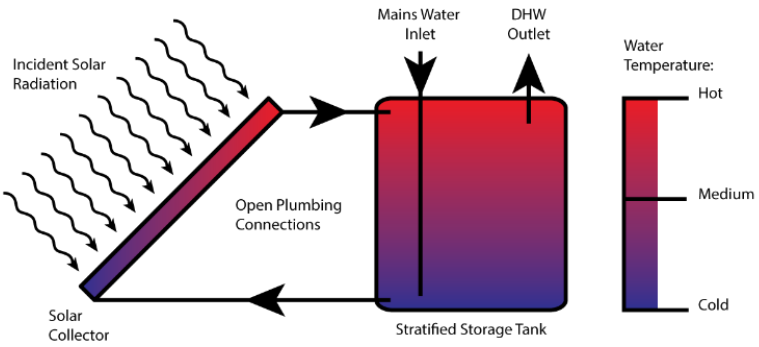


Figure 1-6: Thermosiphon SDHW System Schematic adapted from schematic presented in [14]

An alternative to passive systems that allows more flexibility in component selection and placement is to move to an active design that includes at least one pump and control mechanism. Three configurations that represent the range of traditional active SDHW systems are pictured in Figure 1-7 on the following page [14]. System a) in Figure 1-7 is an open feed system where the heat-transfer fluid in the solar collector is simply water from the DHW tank. Such a system has an effectiveness advantage over the other two systems pictured, which have Heat Exchangers (HX), because all of the

collected heat (neglecting losses in the plumbing) ends up in the storage tank. The pickups in the tank are configured such that cold water is drawn from the bottom of the tank and heated water is returned part of the way up. The height of the water return is designed such that stratification in the tank is maintained even when lower temperature water returns from the collector [14]. In systems b) and c), the closed-loop heat exchangers enable the use of an antifreeze as the heat transfer fluid in the collector. This design is necessary in colder climates where freeze-protection is required, but comes at both the capital cost of additional pumps and heat exchangers, and also a reduced thermal performance as heat must be transferred between the working fluid and the water in the DHW tank [14]. System c) is the closest system in terms of configuration to the one studied in the current work.

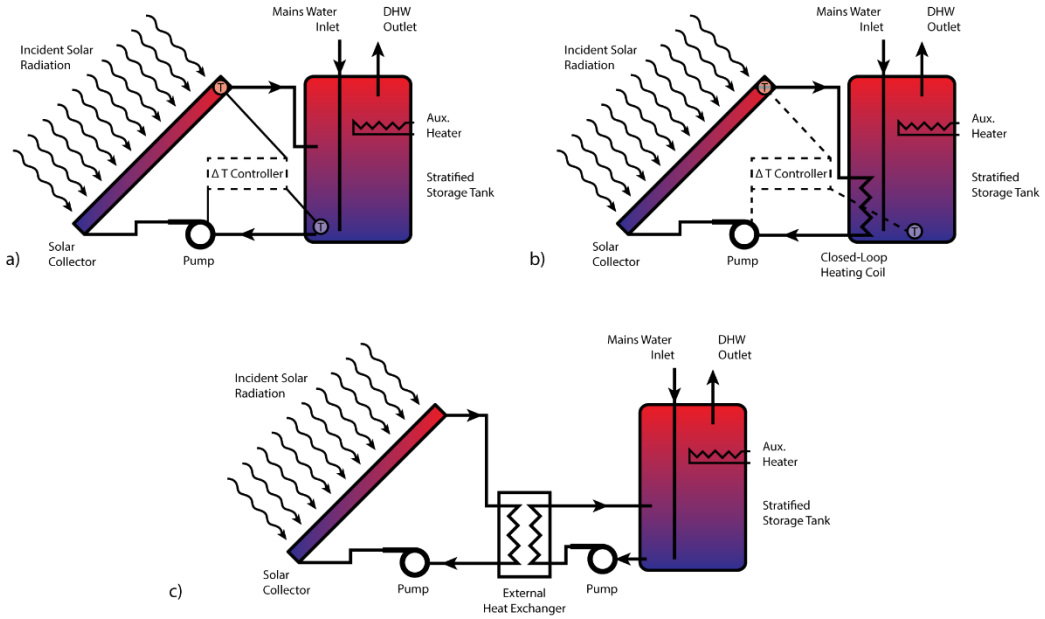


Figure 1-7: Traditional Active SDHW configurations; a) Open feed system; b) Internal closed-loop heating coil; c) External closed-loop heat exchanger

Control systems applied to traditional SDHW systems are typically a simple differential temperature on-off type of control with a temperature dead-band to prevent short-cycling. The controller judges that there is energy available for collection if the temperature at the top of the STC is higher than the temperature of the bottom of the DHW tank by a pre-determined amount and triggers the pump to run until the temperature difference is reduced to zero [14].

1.2.6 Solar-Assisted-Heat-Pumps (SAHP)

The second law of thermodynamics limits the natural direction of heat transfer to be from a hot-source to a cool-sink unless there is a net-work input to the system. A Heat Pump (HP) is a machine that utilizes a net-work input to move heat in the reverse direction to what would naturally occur [15]. That is, it moves heat from a cold-source to a relatively hot-sink. In doing so, a heat pump is able to provide heating to temperatures above those of the heat source, or cooling to temperatures below the heat sink.

A common implementation of a heat pump is one that uses the vapor-compression-refrigeration cycle as illustrated in Figure 1-8. The cycle is composed of a compressor, two heat exchangers, and an expansion valve. A refrigerant circulates within the system and undergoes a phase change in each of the heat exchangers where it either absorbs heat (in the evaporator) providing a cooling effect on the other heat transfer fluid, or rejects heat (in the condenser) providing a heating effect.

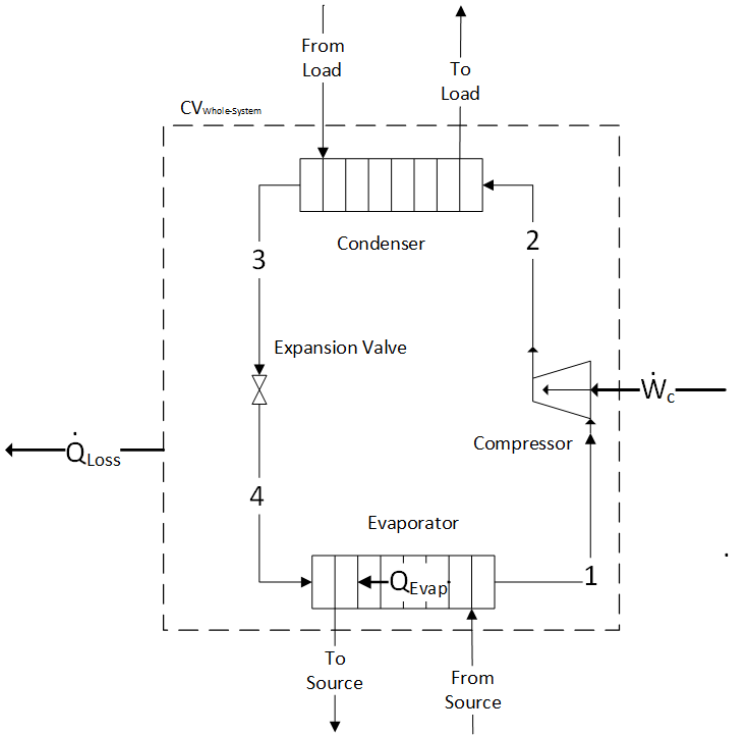


Figure 1-8: Schematic of a heat pump based on a vapor-compression refrigeration cycle

A solar-thermal system that incorporates an STC and a HP is called a Solar Assisted Heat Pump (SAHP) system. The system is termed "assisted" due to the positive interaction of both the STC and the HP on one another. Solar thermal collectors are sensitive in terms of their collection efficiency to

the ambient temperature and the mean collector temperature. The mean collector temperature is heavily influenced by the temperature of the heat transfer fluid being circulated to the STC. A heat pump connected to an STC on its source side can often remove heat from the solar loop at a higher rate than a simple heat exchanger. As a result, the temperature of the fluid within the solar loop can be expected to decrease. A decrease in mean fluid temperature in the solar loop shifts the operating point of the STC resulting in lower losses, higher overall collection rate, and a higher collection efficiency. This effect is illustrated in Figure 1-9.

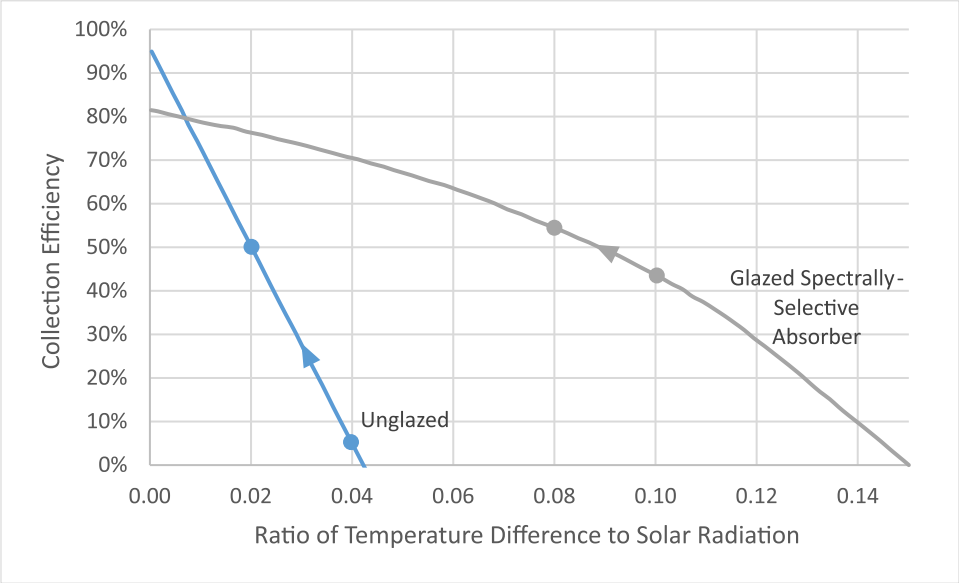


Figure 1-9: Operating Points marked on typical STC performance curves for two types of collectors illustrating the potential performance benefit of attaching a heat pump

As can be seen in the figure above, the benefit of coupling an STC with a HP would not be consistent across all STC designs, nor would it be consistent with respect to the weather and solar conditions. It can be seen from the relative slope of the glazed absorber curve, that a greater benefit would be expected in the domain of low ambient conditions relative to the mean collector fluid temperature (i.e. towards the right-hand side of the plot where efficiencies are quite low). These would be the conditions where SDHW systems are least effective and would often be switched to auxiliary heat. As a result of these benefits, it is predicted that a properly configured SAHP system will collect more energy on both a daily basis, into the shoulders of dusk and dawn, as well as on a seasonal basis, into the shoulders of spring and autumn, when these adverse conditions are common.

Just as the HP is able to increase the performance of an STC, an STC can also positively impact the performance of a HP. The performance of a HP is theoretically related to the Carnot efficiency for a cycle operating between the low temperature source and the high temperature sink. Comparing a HP system that uses a STC source rather than a Ground-Source or Air-Source, the source temperatures in the SAHP system has the potential to be 30 to 80°C higher depending on the weather conditions. This too could result in significant performance increases as the HP would need to consume less electricity in order to charge a DHW tank. These benefits were reported and investigated by Sterling, Wagar, and Banister [16, 2, 3]. They form the basis for the current line of inquiry.

There are a number of SAHP-DHW system configurations that have been proposed and many have been investigated in the literature. Systems can generally be divided into direct-expansion and indirect-expansion styles. In the context of SAHP systems, a direct-expansion system uses the refrigerant from the HP as the heat transfer medium at the solar collector. In effect, the collector becomes the heat pump's evaporator, while the condenser is typically a heat-exchanger that provides heating to the load. On the other hand, indirect-expansion systems typically use a self-contained heat-pump unit with a heat exchanger for the evaporator and the condenser. These units are then plumbed into an SDHW system using water or antifreeze as the heat-transfer fluid between the evaporator and the collector, and potable water on the load-side of the condenser. This research focusses on an indirect-expansion system, so only indirect systems are depicted in the example figures to follow.

Figure 1-10 shows 3 examples of indirect SAHP-DHW configurations described as follows:

- In a), water is circulated between the DHW tank and the condenser of the HP, as well as between the STC and the evaporator of the HP. In effect, this boosts the outlet temperature of the STC providing higher quality heat to the DHW, while cooling the inlet to the STC, which increases its collection efficiency.
- In b), the HP is on the solar-side only, and heat transfer fluid circulates between the STC and both sides of the HP. This allows heat to bypass the STC, keeping the mean STC temperature cooler and theoretically increasing the collection efficiency.
- In c) there is a second "float" tank which operates similar to an open-feed SDHW system and receives heated water directly from the STC. After low-temperature storage in the float-tank, the heat is transferred to the DHW tank and the temperature is simultaneously boosted by the HP to meet higher temperature demands of the DHW system.

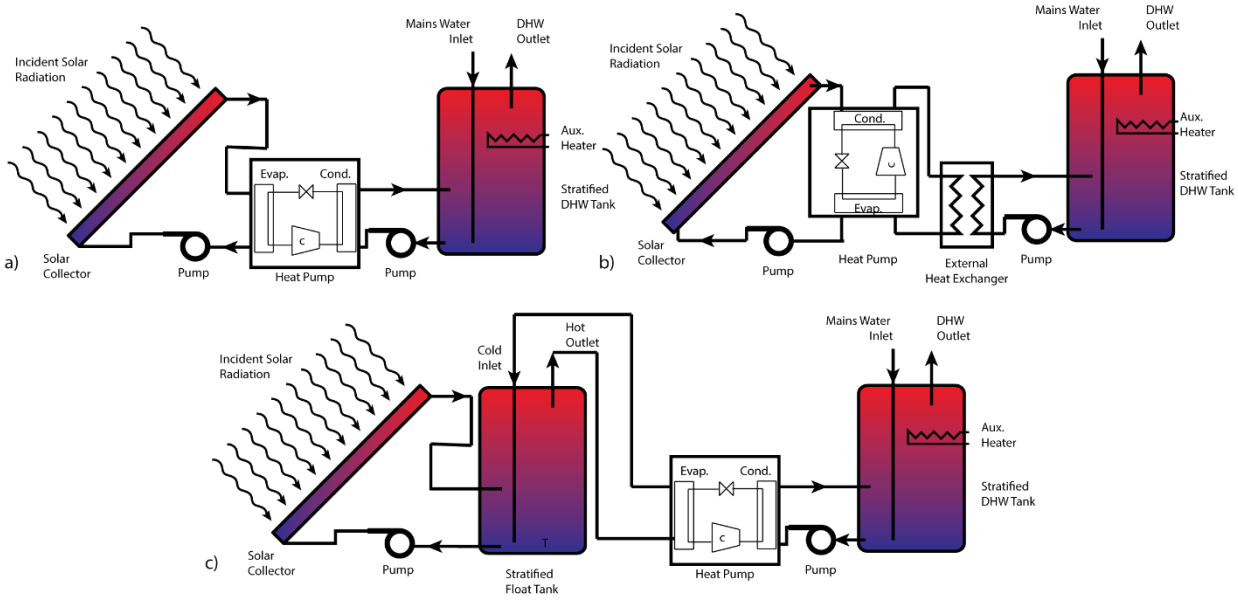


Figure 1-10: Simplified examples of indirect-expansion SAHP-DHW system configurations
a) Conventional HP as investigated by Bridgeman [17], and later Wagar [2];
b) Solar-Side (Parallel) HP as investigated by Sterling [16]; and
c) Dual-Tank System as investigated by Sterling [16] and Banister [3]

1.3 Research Objectives

1.3.1 Current System of Interest

Of interest in this work is a domestic hot water system arranged as pictured in Figure 1-11. The system makes use of a single storage tank having a capacity of 364 L (80 US Gal). This system incorporates flow diverters to switch between operating as a traditional SDHW system, or as a SAHP system when required. A novel aspect of this system is the variable HP, which can modulate its heat transfer capacity on demand.

The HP, which is described in more detail in future chapters, is a custom-built unit that incorporates a Variable Frequency Drive (VFD) to control the rotor speed of a 3-phase scroll compressor and in doing so, modifies the HP's mass flow rate of refrigerant which changes the rate of heat transfer from the source to the load side of the system. The need for this special application heat pump is made clear by calls from other researchers cited in the literature review chapter.

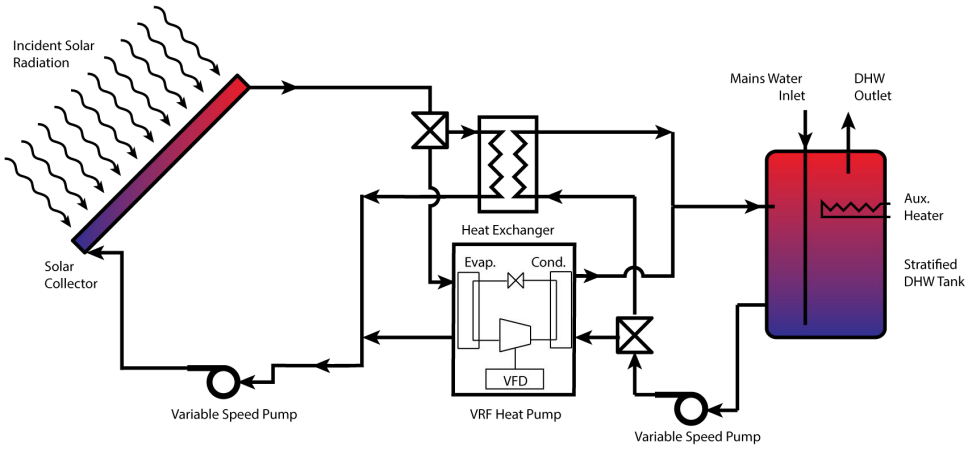


Figure 1-11: System layout of multi-mode SDHW/SAHP system under study

The system described here is a conceptual representation of a multi-mode variable capacity SAHP system that incorporates components sized for a typical SDHW installation in a single-family residence. The STC is specified as an array of 2 glazed, spectrally selective flat plate collectors connected in series with a gross array area of approximately 5 m^2. Further details about the physical representation of this system for experimental investigation, and a numerical-analytical model representation of this system for simulations will be provided in future chapters.

The first objective of the current work is to explore the possible performance benefits of incorporating a variable capacity HP in an SAHP system. This will be done through a mix of experimentation and simulation-based approaches.

1.3.2 System Control Improvements

As will be seen in the literature review chapter, many studies have based SDHW and SAHP system operation on temperature-based controllers (thermostats). Thermostat-based control is by far the most common method in traditional SDHW systems going back to their inception [14] and is the predominant control strategy for other conventional DHW storage systems as well. Various works have called for further controls development [3, 17, 18, 19, 20].

The current work aims to investigate the use of a predictive controller with iterative updating in a SAHP system. This predictive controller follows some principles of a typical Model Predictive Controller (MPC), which is a broad class of control methodology that incorporates the following elements [21]:

1. The explicit use of a model to predict process output out to a future time horizon.
2. Generation of a sequence of control signals that minimize an objective function.
3. Displacement of the time horizon into the future by the incorporation of control signals from previous time steps.

As a general class of control methodology, MPC has wide applications that reach into chemical process control (including energy systems), mechatronics, traffic management, electric circuits, etc. [22]. The key distinguishing elements between the various implementations of MPC are the model development and the objective function. The process model plays a critical role in the functioning of a MPC controller.

The control system employed for this work is distinguishable from MPC, however, because instead of using an analytical model to predict the output of the system and mathematically optimizing the control series, the process model employed is a numerical system model that is iteratively updated. An objective function is minimized numerically to arrive at the desired control series.

Because this control approach involves inherently minimizing an objective function over a forecasted timespan, it can be thought of as an optimizing controller, although care is taken to qualify the use of the word optimum here to be limited to the operational optimization identified by the controller and subject to the constraints applied to the system. This optimum is also considered a local optimum since it is arrived at through an iterative numerical approach beginning at the initial conditions of the system, as opposed to an analytical mathematical optimization that could be guaranteed to produce the global minimum of an objective function. Nevertheless, the implementation of this improved controller to a multi-mode SAHP system has promise to better utilize a given system's resources to better achieve the performance potential of the system.

A second objective of this research is to respond to calls in the literature for more detailed investigation of a system that employs a more advanced control methodology. The improved controller developed in this work is identified as having potential to significantly improve the performance of a SAHP system and with the flexibility to make best use of the additional degree of freedom that comes with the variable capacity HP.

Chapter 2
Literature Review

2.1 Recent Work at the Solar Thermal Research Lab (STRL)

In recent years, several investigations of SAHP systems have been conducted in the Solar Thermal Research Lab (STRL) at the University of Waterloo (UW). This section reviews the internal work in the area by graduate students up to the beginning of the current work. These studies inform and lend technical background to the current work.

In his master's thesis, Sterling [16], performed a simulation-based study on two SAHP-DHW system configurations and compared them to electric-only and traditional SDHW systems. The first configuration studied was a "solar-side" system where the heat pump runs in parallel with the solar collector to boost source-side temperatures in the heat exchanger. This first configuration was depicted as system b) in Figure 1-10. The second configuration studied was a two-tank system shown as system c) in Figure 1-10, where the solar collector was able to directly charge a "float tank". The float tank was allowed to have any natural temperature that weather and draw conditions created. Either a Heat Pump, or a Heat Exchanger was able to selectively transfer heat from the float tank into the main DHW tank, where water was maintained at a set point temperature, ready for use.

Sterling concluded that both SAHP systems showed favorable energy performance over the traditional systems they were compared against. It was noted that the solar-side system's operation featured longer solar collector runtimes in the summer, while the dual-tank system had longer runtimes in the winter. Longer runtimes generally equated to more energy collection in a day. The dual tank system was found to collect the most solar energy over the course of the simulation, however, the additional energy collected was offset by higher tank losses and greater electrical energy consumption.

Sterling's recommendations included continued investigation using computer simulation techniques to understand the impact of a variety of parameters on system performance. It was also recommended that working prototype models of systems be built to validate the simulations and inform the practical aspects of the design and implementation of these systems. Lastly, Sterling recommended further study of installations in various locations, with realistic draw profiles which were not considered in his study.

Wagar, in his master's thesis [2], performed simulation and experimental validation of a single-tank SAHP system that could alternately use a heat pump, or heat exchanger to move heat from the solar collector to charge the DHW tank. Wagar designed and constructed the first of a series of experimental test units (ETU) in the STRL, to validate the TRNSYS model that was developed to simulate the single-tank system.

A contribution of Wagar's was the development of a heat pump performance map for the single-stage water-to-water heat pump under study at the time. This performance map was achieved through an experimental campaign that spanned the full range of operating temperatures of the system but only explored a single water flow-rate. The result was a set of third-order polynomials, fit to the empirical performance data that described the HP operation.

With regards to validation of the TRNSYS model used in system simulation, DHW tank temperatures were cited as a potential area of concern. The built-in TRNSYS models for thermal storage in tanks were noted to not accurately reflect the stratification observed in the experimental tank. The tank node temperatures were used to inform control decisions of the system, and thus, inaccuracies in tank temperatures propagated through a myriad of other simulation inaccuracies. Wagar's recommendation in this area was the investigation or development of a more sophisticated tank model that could take into account the heated plume that develops around the auxiliary heater in the tank, as well as better handle stratification and mixing. With regards to the heat pump, Wagar noted that a major issue to be resolved is that of heat pump sizing. The heat pump that was studied was oversized for the system and therefore short-cycled and operated outside of its peak efficiency range. Wagar recommended investigation of the performance of a variable capacity HP using either a digital scroll compressor or a VFD.

Banister performed a simulation and experimental evaluation of a dual-tank SAHP system which was the subject of his PhD research thesis [3]. The system was capable of commuting between operating modes that delivered heat either via a heat pump or heat exchanger from the solar collector to the DHW tank, or first stored the heat in a "Float Tank", before using the heat exchanger or heat pump. His work involved modification to create the second of the series of ETUs in the STRL. A second tank was added as well as significant modification to the control infrastructure to facilitate validation of a more complex TRNSYS model that simulated multiple tanks and modes of operation.

Banister's contributions included an understanding of the interaction between low-temperature heat storage, high-temperature storage, and effectiveness of heat transfers between the two storage tanks. His research also included the development of a mode-switching control system for the ETU. Banister did not endeavor to optimize the control protocol used in selecting operating modes for the system that was studied, and recommended that future work be focused on parametric optimization of the controller in order to further improve system performance. It was concluded, however, that for a system with approximately 2.5 m^2 of collector area, an approximately 39% savings could be realized over a traditional DHW system, and the dual-tank system performed approximately 6% better in terms of energy cost than a traditional SAHP system.

Banister conducted the majority of simulations using a scaling factor of 0.6 to reduce the capacity of the heat pump that Wagar had mapped and modeled in TRNSYS. The nominal capacity had to be reduced to allow for steady-state heat removal from the solar collector loop under common irradiation conditions. This was necessary to avoid short-cycling and freezing out the collector loop.

Banister concluded that with an oversized HP, there wasn't an appreciable benefit to adding a HP to a properly-sized, traditional SDHW system. It was found, however, that significant benefit could be reaped from the addition of the second "Float" hot water storage tank. Oversizing the collection system in terms of the collector area also improved the energy performance of the system, but with diminishing returns beyond a second panel.

Banister recommended carefully sizing the heat pump for applications of SAHP, but also noted that in cases where an oversized heat pump is specified, trouble can be mitigated by avoiding the operating mode that pumps heat directly from the STC to the DHW tank (i.e. use a HX to transfer the heat to the float tank first, then a heat pump from the float to DHW tank).

Most-Recently, Chasse conducted master's research on the topic of controls development for a single-tank SAHP system [18]. The bulk of this work was focused on evaluating control protocols that sought to minimize energy consumption, maximize heat transfer rate, or deliver a heuristic balance of the two. This work was entirely simulation-based, and is left with the recommendation that experimental validation be performed in the future.

In Chasse's work the control architecture was heuristically developed with the goal of seeking the best operating mode based on current conditions such as tank temperatures and targeted recharging rates. The evaluation was performed by modeling the electrical power consumption and heat gains

from the solar collector to the storage system on an ongoing basis and then predicting the best operating mode at each time step. The controller decided between the following modes of operation: “Off”, “HX-only”, “HP-only”, “Series-only” and “Warmup”, a mode used to build up solar energy in the solar side loop.

The series (SRS) mode was developed as a way of tempering the source-side inlet temperature to the HP during periods of higher-than-normal solar irradiation. In this mode of operation, the heat transfer fluid passes through both the HX and the HP. Simulations showed that in the current implementation, this mode of operation was very rarely called for (less than 3% of the time compared to other operating modes) and did not prove to be the best mode of operation on any of the days tested in the study.

In addition to the previously stated recommendation that validation experiments be performed to match the TRNSYS model to actual system components, Chasse made further recommendations with regards to the development of 3-mode SRS-SAHP systems. It is recommended that a study be conducted with a modified controller that forces the use of SRS mode whenever it has a high heat-transfer potential, so that the other performance parameters, such as solar fraction and electricity consumption can be assessed.

Since the controller developed by Chasse involved a number of gain and set-point parameters, a parametric optimization study is recommended to evaluate the complex interplay between these different settings. The starting point, default settings were not guaranteed to be at or even near an optimum condition as no optimization was performed as a part of the study.

2.2 Other Recent Work at Canadian Institutions

Chu, Choi, Cruickshank, and Harrison [23] modeled an indirect SAHP system for use in the Team-Ontario entry into the 2013 United States Department of Energy Solar Decathlon. This innovative system consisted of two storage tanks, one hot and one cold. Heat was pumped from one to the other to meet domestic hot-water, as well as space heating and space cooling demands. Simulation showed that approximately 58.3% of the annual heating, cooling, and DHW loads of the house could be met by solar energy. The authors concluded that although single-parameter studies were useful in guiding the preliminary design of their system, more work was required to characterize the interactions of multiple design parameters.

Chu and Cruickshank [24] conducted a thorough review of the existing studies into SAHP. The purpose of the work was to analyze the key performance data from existing studies and to attempt to assess the merit of individual system configurations for residential buildings in the Canadian climate. The study concluded that it was difficult to determine a single optimal configuration, as the performance of the systems are dependent on many factors including building characteristics, control parameters, included system components and local climate. It was further noted that existing studies are not consistent in the performance criteria that they use to assess their systems. The authors call for a standardized set of performance indicators for future studies in order to aid in the cross-comparison of different system designs.

Bursill and Cruickshank [25] evaluated the performance of a commercially available air-source Heat Pump Water Heater (HPWH). The system consisted of a preheat, and a HPHW tank. The HPHW tank received heat input both from the integrated HP condenser, but also from a booster heater near the top of the tank. The study dealt first with characterizing the equipment, then modeling its performance in TRNSYS. Finally, the researchers assessed various control strategies based on the temperature set-points of both the HP and the booster heater. They concluded that the total energy consumption of the system was least when the heat pump was used the most. They also found that having the HP set point equal to the booster heat set point resulted in the absolute least energy consumption. The authors' suggestions for future work include multiparameter studies which may take into account the effects of thermal heating dead bands and tank insulation properties, as well as seasonal simulations to optimize the energy use of the system.

Baldwin and Cruickshank [19] studied a space heating and cooling system that included both hot and cold-storage as well as a standard geothermal heat pump. A goal of this research was to assess the peak load-shifting capabilities of such a system to reduce the peak demand on the electrical utility due to space cooling and heating loads. It was found that although the percentage of energy used during peak hours could be reduced by using thermal storage, the studied system used more annual energy and caused more GHG emissions than a heat pump-only system. This was attributed to the different source and load temperatures seen by the heat pump, causing a lower Coefficient of Performance (COP) during the system's operation. The authors called for more work on the particular system including more control optimization, as well as the selection of a more appropriate heat pump model in order to achieve their stated goal of both reducing peak loads, as well as reducing total energy consumption and GHG emission overall.

Harrison [20] reviewed the current state of SAHP for DHW including direct, indirect, series and parallel systems. He discussed collector selection and compared glazed and unglazed collectors, as well as hybrid collectors which include a back-vent to allow convective heat transfer to the collector from the ambient air when conditions are favorable. It is indicated that these hybrid collectors can be used quite effectively, however their applicability is sensitive to climate, and system operating conditions. With reference to the work of Sterling, Harrison states that further research into the solar-side system would be of value. It is also suggested that research into multi-mode, and combined systems is of value. Harrison concludes that issues related to system control have not yet been resolved, indicating that there would be some value to continued research into system control.

Berger and Harrison [26] compare and evaluate PV-DHW and PV-Thermal (PV/T) DHW systems for the Canadian climate. Year-long simulations were performed in TRNSYS with 3-minute time steps. Real system components were simulated using built-in TRNSYS "TYPE" models. Simulations showed that roughly double the array area of PV was required as compared to PV/T to achieve the same solar fraction. It was noted that the PV-DHW performance suffered because of mixing in the tank caused by the resistance heating element as opposed to the relatively stratified operation of the SDHW system. The authors note as well, that the base PV configuration is still too expensive to justify implementing in lieu of other solar energy options including combined PV-Thermal, and regular SDHW. They therefore recommend further research and development in the area to improve system performance and reduce costs.

Asaee et al. [27] performed a techno-economic investigation to determine the feasibility and benefit of retrofitting Canadian houses with the SAHP system proposed by Banister but extending its use to include both DHW and space heating thermal loads. A series of eligibility criteria were developed to determine which houses should be retrofitted. Approximately 37% of Canadian homes met the eligibility requirements. Simulations performed in ESP-r showed that GHG emissions due to water heating in Canadian homes could be reduced by about 20% if all eligible houses were retrofitted. The authors noted that the benefit of retrofitting homes with the proposed system was heavily dependent on climatic conditions, auxiliary energy sources and fuel mixture for electricity generation. They also concluded based on their economic study, that incentive programs would likely be required in order for property owners to justify the expense of retrofitting their homes.

Unrau and Lightstone [28] studied the numerical error associated with the temperature profiles of simulated water storage tanks using models available in TRNSYS. Simulation results were compared to an exact analytical solution for 1-D advection-diffusion. The authors note that in the simulated results, the models over-predict the size of the thermocline in the tank compared with the analytical solution. The results appear to show that the accuracy of the tank model is more sensitive to the number of nodes, than to the length of the time step. The numerical scheme over predicts the diffusion. It is noted that a larger number of nodes in the tank, approximately 1000 would reduce the error to an acceptable level, however that number of nodes is greater than the default setting TRNSYS would allow. The authors studied the effects of the numerical error in the tank models on the predicted solar fraction of various SDHW systems. They concluded that grid spacing had less of an impact on systems with undersized tanks, and more of an impact on larger tank systems. It can also be noted in the results presented that the result of an oversized thermocline caused by over predicted diffusion would lead to an under prediction of the solar fraction of an SDHW system.

2.3 Recent International Work

Buker and Riffat [29] perform a systematic review of SAHP systems including both direct and indirect systems. With regard to International Energy Association (IEA) Solar Heating and Cooling (SHC) Task 44, Subtask B, the authors attempt to consolidate the performance metrics used in the various studies using a tabulated presentation of many studies. There is large variation in the performance criteria and analysis techniques used in the various studies, which is problematic for comparative purposes. The review concludes that no single optimum solution can be identified in the literature, but prompts further investigation to prove both the economic viability, and development of a standardized assessment method for SAHP systems.

Carbonell et al. [30] studied an indirect SAHP system with combined ice storage, and Drain Water Heat Recovery (DWHR). The system was purposed for both DHW and space heating loads. The simulations were performed in TRNSYS with individually validated components. No whole-system experimental validation was performed. Ice storage in this system can be thought of as an extension of the dual-tank SAHP principle where the heat pump can boost low-grade heat that is stored in the ice storage. This system uses specialized heat exchangers and controls that allow the heat pump to draw from a source temperature that is below freezing, leveraging the latent heat of freezing water to store and retrieve more energy than would be possible from a sensible-only system. Since the storage

temperature is quite low, the system lends itself well to heat recovery from the drain water. The study concluded that an energy benefit of 2% can be achieved pairing this system with a flowing DWHR system, and from 8.5% to 22% when using a waste-water storage heat recovery system.

Fraga et al. [31] consider the case study of a housing complex of 4 buildings with high performance envelopes, housing 10 blocks of 8 flats. Both DHW and Space Heating (SH) loads are met by a combined SAHP system. The authors use the term "Direct" to mean a connection that bypasses the HP, from the system schematic, it is understood that this is an indirect-expansion style SAHP, connected to a system which is capable of operation either in SDHW or SAHP mode. The authors also prepared a numerical model to simulate and perform parametric sensitivity study of component sizing. The system under investigation produced a relatively low seasonal performance factor as a result of a heat distribution loop that is used for both DHW and SH with separate DHW storage in each residential unit, which forced higher temperatures in the loop, and did not allow for solar preheating of the DHW.

Poppi et al. [32] studied a combined DHW and SH system that used both a solar collection loop and an air source heat pump to meet the thermal loads of two test houses in Zurich and Carcassonne. They studied the effects of a heat pump cycle with vapor injection (vapor bypasses the evaporator, improving performance) compared to a cycle without vapor injection at the compressor. A variable speed compressor was also examined. The study found that in one of the test homes, in Zurich, the variable speed compressor offered a benefit, while in the other, there was less advantage in terms of electricity consumption. This difference was attributed to the different SH types, which require different heating loop temperatures.

Bellos et al. [33] compared performance of a PV-powered air-source heat pump to that of a PV/T and a flat plate collector source connected to an indirect-expansion HP system used to meet space heating demands in a building of 100 m^2 in Athens. In the PV-connected HP example, the study found that 20 m^2 was an acceptable collector area to sufficiently meet the heating demands (with a connection to the electricity grid available as a backup). For the flat plate collector study, it was found that a storage tank between 1 m^3 and 1.5m^3 offered an acceptable amount of storage, but a similar size of collector area, 20 m^2, was required to meet the indoor temperature profile required for thermal comfort. For the PV/T system, it was found that no decrease in collector area was possible due to the same thermal comfort requirements, however, the grid electricity consumption was significantly

reduced, and the COP of the heat pump was augmented with greater collector areas. Solar energy was able to meet approximately 90% of the heating load. The study concluded that for electricity prices between 0.2 €/kW h and 0.23 €/kW h (in CAD currency in 2017, $0.29 to $0.34 per kW h) a PV-connected air-source heat pump was economically favorable, while for higher electricity prices PV/T collectors and an indirect-water source HP made a better choice.

Youssef et al. [34] investigated an indirect expansion SAHP system that incorporated a Phase Change Material (PCM) storage tank as an additional input to the heat pump which was used to meet the DHW demands of a standard UK dwelling. The DHW load was 350 L per day. A fully-functional test rig was built including evacuated-tube solar collectors, DHW storage tank, HP, and the PCM heat exchanger. The study concluded that the PCM was a viable alternative to water-based storage, and allowed more energy to be stored with fewer losses, in a smaller volume of storage. The COP of the heat pump was shown to increase by a range of 6.1% on sunny days, to 14% on cloudy days over the performance of a system without PCM. This research included a specific focus on a control strategy with an objective of improving system efficiency and maintaining a constant hot water supply temperature.

Wang et al. [35] presented a review of SAHP for DHW research. This work presented and classified various types of systems, describing the theory that underlies each system, and summarizing the parametric data published by the principal investigators of the systems. The authors comment that future work in the area should focus on the following 4 items: First, the development of new energy-efficient systems that leverage and combine advanced technologies, while still remaining economically viable. Secondly, deployment of in-situ and onsite measurement systems for real test cases. Thirdly standardizing the sizing and selection of components and system architectures. And finally, investigating and developing energy policy and incentive programs to help encourage the expansion of the SAHP technology.

2.4 Author's Contributions

The literature presents a number of configurations for SAHP systems similar to those described in the introduction section. A complex thermal system could incorporate any combination of storage tanks, heat exchangers, circulator pumps and heat pump connections. This project will examine an indirect-expansion SAHP system that can convert between HP-only, HX-only, or HX-HP mixed-modes of operation. A novel addition to the system under study is the incorporation of a variable capacity HP

between the solar-side and the DHW side of the heat-transfer loop. The variable capacity HP and two variable speed water circulation pumps on both sides of the heat transfer loops interact to affect the energy efficiency and heating capacity of the system leaving open a number of research opportunities within the overall umbrella of determining an optimum configuration for this system through experimentation and simulation.

2.4.1 Characterizing and Modeling of a Variable-Capacity Heat Pump

Where a number of previous studies have included TRNSYS modeling of SAHP systems and called for either a smaller capacity or variable capacity HP, the models available in TRNSYS by default are limited in this area. Some previous studies have applied scaling factors to the fixed-capacity HP models, and called for further investigation as future work.

In Chapter 3, this work contributes an experimental characterization of a variable capacity water-to-water HP. The empirical performance map generated from this characterization experiment is built into a custom TRNSYS TYPE model presented in Chapter 4, which is validated using day-long solar simulation experiments. The contributed HP model improves on what was previously available in the following ways:

- Validated over the entire range of temperatures under-which the HP is permitted to operate (safety limits set by manufacturer).
- Modeled for a range of heat transfer fluid flow rates under balanced and unbalanced flow conditions.
- Model input representing the compressor driving speed as opposed to an arbitrary scaling factor applied to the rated capacity.

This work was shared with the broader research community in [36]. It will be useful to other researchers both within the STRL and externally, who wish to perform TRNSYS simulations of variable-HP systems.

2.4.2 Overall SAHP/SDHW System Performance Comparison

The literature is rich with studies of a variety of SAHP system topologies but inconsistent in the key performance indicators used to evaluate and compare those systems. Studies in the literature have called for greater consistency in Key Performance Indicators (KPI)s reported [24].

Also, in Chapter 4, this work contributes a validated system model of an SAHP-DHW system that can be controlled by interchangeable MATLAB scripts. When configured to operate in restricted modes, the model can behave as if it were an electric-resistance DHW system, an external HX SDHW system, a fixed capacity or variable-capacity SAHP system, or a mixed-mode system. Simulation results in this work are presented on the basis of relevant KPIs from the literature, increasing the likelihood that they may be compared against other studies who present similar KPIs, but also system performance can be compared between different systems discussed in this thesis like-for-like since the same validated model components were used to simulate each system, removing the confusion or uncertainty that comes from benchmarking against other studies' KPIs which are known to be variable in the literature.

Chapter 5 of this thesis presents a comparison of systems under a variety of system configurations which are analogous to various traditional system topologies. The comparison of these configurations using a consistent set of KPIs is useful in highlighting the relative strengths of the various systems, and also highlights the need for more advanced control of more advanced systems. This part of the work was shared with the research community in [37].

2.4.3 Predictive Control with Iterative Update for a Variable-Capacity SAHP System

Prominent studies in the literature have focused on fixed-capacity HP systems, and limited their control methodology to temperature-differential controllers with a fixed setpoint and deadband. A recurring call for more advanced controls development has been noted [3] [20].

Furthermore, the majority of SAHP simulation studies available in the literature make a disclaimer that their simulated system performance is based on an un-optimized controller. In Chapter 6 of this work, a contribution is made towards an improved control strategy of SAHP systems through the development, implementation and simulation of a predictive control algorithm that is capable of selecting the optimal operating mode for the system (best result over the simulation window) subject to a variety of imposed constraints.

Since the controller employed in this work can be thought of as an optimizing controller, the contribution stated above can equally be framed as a contribution towards a better understanding of the best-case or upper limit of system performance for a variable-speed SAHP system as configured in this study. This represents a significant improvement in clarity of the performance potential of these systems, as opposed to previous studies that have made statements to the effect of: *the system*

can perform at least this well, but has potential to perform better under a different control scheme [3, 18, 20, 27].

Another novel aspect of this work lies in the ability of the improved controller to detect and address fault conditions such as system stagnation. Generally, stagnation has been addressed in the literature through hardware by increasing tank storage capacity as a ratio to collector surface area or including separate water-to-air heat exchangers for heat rejection when DHW storage is at capacity. In this work, over-charging of the DHW tank and stagnation of the STC are avoided by the forward-looking control software with positive results.

2.4.4 Operation of a Working Prototype System

While a standalone prototype system has not been commissioned, the author has configured the ETU in the STRL to operate as the system under current study and in Chapter 7 of this thesis, a series of experimental trials are presented of daily operation of a variable-SAHP system under the presented improved controller. This is presented as a step towards a functional prototype that may be retrofitted and monitored in-situ in an occupied residence.

For the current contribution, the transformation of an experimental controller out of the numerical simulation study previously described into a physical experiment, controlling real hardware as a Hardware in the Loop (HIL) style simulation serves as a significant step towards the research goals outlined in the introduction chapter.

Chapter 3
Experimental Test Unit

3.1 Experimental Test Unit Description

All experimentation detailed in this thesis employed an experimental test unit which has been constructed in the STRL. The test unit is based on a construction in the STRL held over from previous studies. It is extensively described in the works of Wagar [2], Sterling [16], and Banister [3], and the reader is referred to those works for a full description of system components and reconfiguration options. The ETU has been modified so that it can emulate the system under study shown previously in Figure 1-11. The present discussion will focus only on the aspects of the ETU that relate to the current research. The most significant change to the ETU from previous work is the addition of a new custom-built heat pump with compressor modulating capabilities, this new HP is the subject of Section 3.9.

The ETU is designed to offer a great deal of flexibility in terms of flow paths through the various components and also in terms of the flow rate and heat source for the DHW system. Electrically actuated flow diverter valves allow for the selection of plumbing branches that bring key components into and out of the system on demand. In doing so, the ETU can simulate all of the operating modes of the system of interest from Chapter 1. Figure 3-1 is a photograph of the ETU from the operator's position.

The main components that can be brought in and out of the system are: Two insulated storage tanks, a heat exchanger, a variable-capacity heat pump, variable speed water circulator pumps, an immersion heater configured to simulate the heat input from a solar-thermal collector, and instrumentation to measure temperature, flow rate, and electrical power input to the various system components. Control of the system is achieved through a LabVIEW interface. Through the inputs and outputs of the test rig's Data Acquisition system (DAQ), components of the system are controlled and monitored.

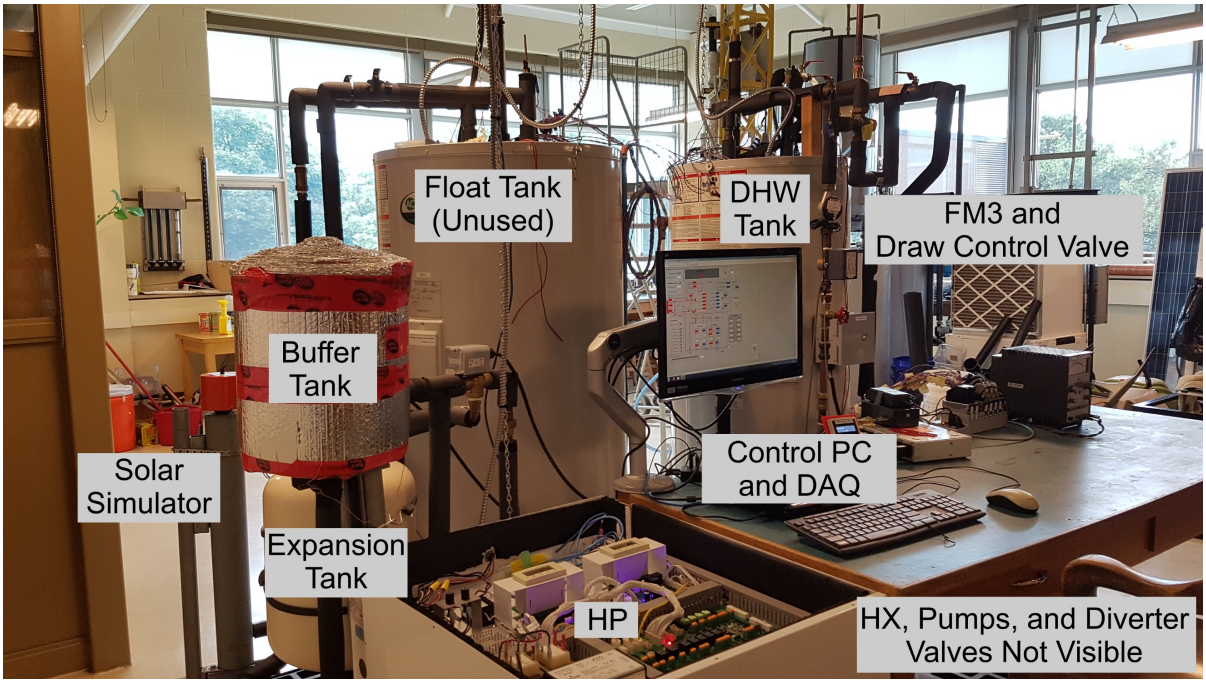


Figure 3-1: Photograph of the ETU

3.2 LabVIEW Control and Data Acquisition

Both input and output data from the ETU are managed through LabVIEW interfacing with a National Instruments DAQ system. Table 3-1 lists measurement equipment on the ETU and the uncertainty associated with the direct measurements and data acquisition. Sample uncertainty calculations for key derived quantities are included in Appendix A.

Table 3-1: Direct measurement uncertainty of specified instruments in ETU

Instrument:	Measurement Type (Symbol) [Units]:	Spec. Accuracy per Data Sheet	Nominal Measurement Value
NI cDAQ-9178 DAQ Chassis	-	-	-
	Digital interface for cDAQ expansion cards. No specific error associated with the DAQ Chassis.		
NI 9213 Thermocouple Card	Temperature (T) [°C]	$\pm 0.1°C$ *or* 0.5%	20°C
	2 expansion cards connected with 31 Type-T Thermocouples. Uncertainty based on spool-calibration with 4-wire platinum RTD and Omega 869C Digital thermometer.		
NI 9265 Analog I/O Card	Current (I) [mA]	± 0.5 mA *or* 2.6%	20 mA
	This card is used as an output to control pump and compressor speeds. .		
NI 9041 Digital I/O Card	Frequency (f) [Hz]	± 0 Hz	3.6 Hz
	Counter function used to read frequency of pulses generated by flowmeters. The nominal frequency of measurements is much smaller than time-base of the digital counter (100 MHz), so the uncertainty of the frequency measurement is taken to be negligible.		
NI 9205 Analog Input Card	Voltage (V) [V]	± 0.006 V *or* 0.6%	1 VAC
	Analog Voltage input used for HP Power, HP Current, DHW Power, and DHW Current sensing. Each quantity is read as an RMS AC voltage signal, then transformed within NI LabVIEW.		
Magnelab SPT-0375 Potential Transformer	Voltage - Voltage ($V_{out} = \frac{0.333}{230} V_{in}$) [V]	± 2.06V *or* $\pm$ 1%	206 VAC
	2 Units, one installed at the DHW Aux. Heater, and a second at the HP electrical input. Transforms mains voltage into a safe level to pass to the NI 9205 Analog Input card.		
Magnelab SCT-0750 AC Current Sensor	Current - Voltage ($V = \frac{0.333}{25}$I) [A → V]	± 0.1A → ± 0.0033V *or* $\pm$ 1%	10A → 0.3333V
	2 Units, one installed at the DHW Aux. Heater, and a second at the HP electrical input. Transforms AC Current into an analog voltage output at a safe level to pass to the NI 9205 Analog Input card.		
Omega FTB-4607 Hall-Effect Flow Meter	Pulse/Volume ($\dot{V}$) [GPM or L/min]	± 0.16 L/min $\pm$ 1.5%	10.75 L/min
	Three Flow Meters in system, one each for source-side, load-side, and DHW draw to sink. The meters have a pulse-rate of 20 Pulse/L.		

LabVIEW also communicates through digital IO and a multitude of relays to provide power to diverter valves that switch the flow of the system between the various flow paths. Circulator pumps are controlled by a 4 to 20 mA current output signal controlled by LabVIEW. The HP (described in a future section) requires a speed input, which is also generated by LabVIEW as a 4 to 20 mA current control signal. Flow rates, temperatures, voltages and current are processed through various custom

LabVIEW VIs both for system performance logging and to supply data to any control system that is implemented.

3.3 Relevant Flow Paths

To illustrate how the ETU physically models the system of interest described in Chapter 1, two relevant flow paths are highlighted in Figure 3-2 and Figure 3-3.

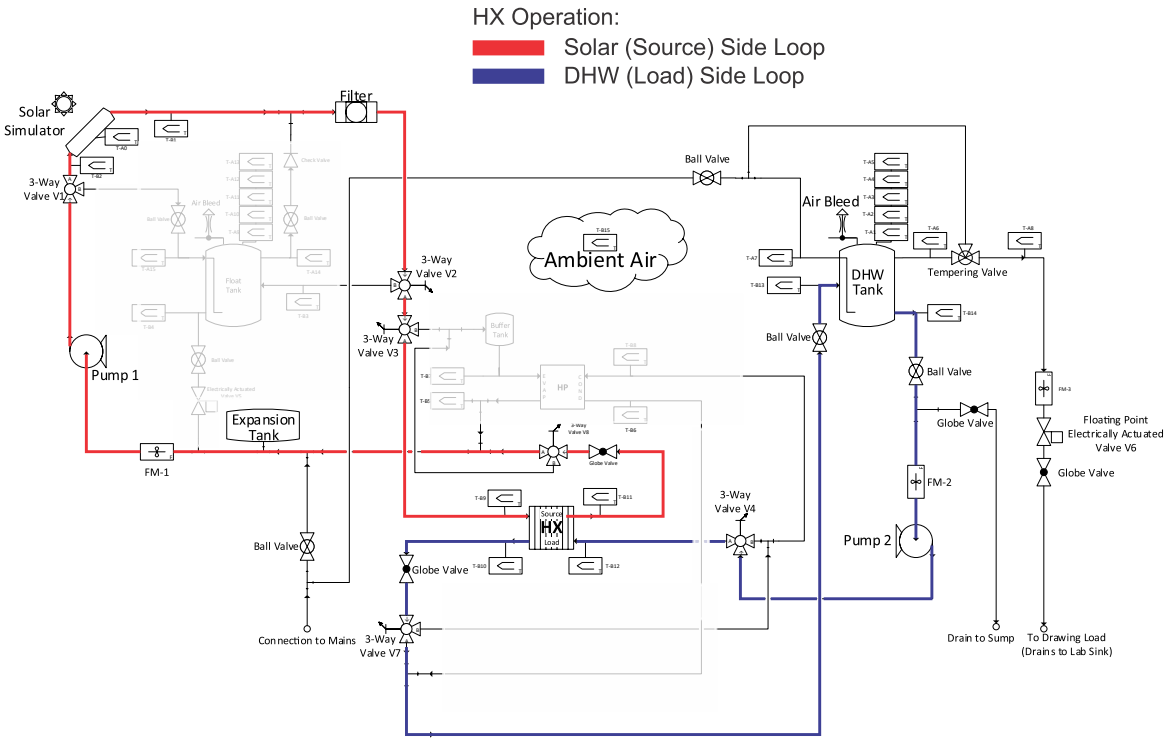


Figure 3-2: Flow path for heat exchanger operation, ETU behaves as an SDHW system

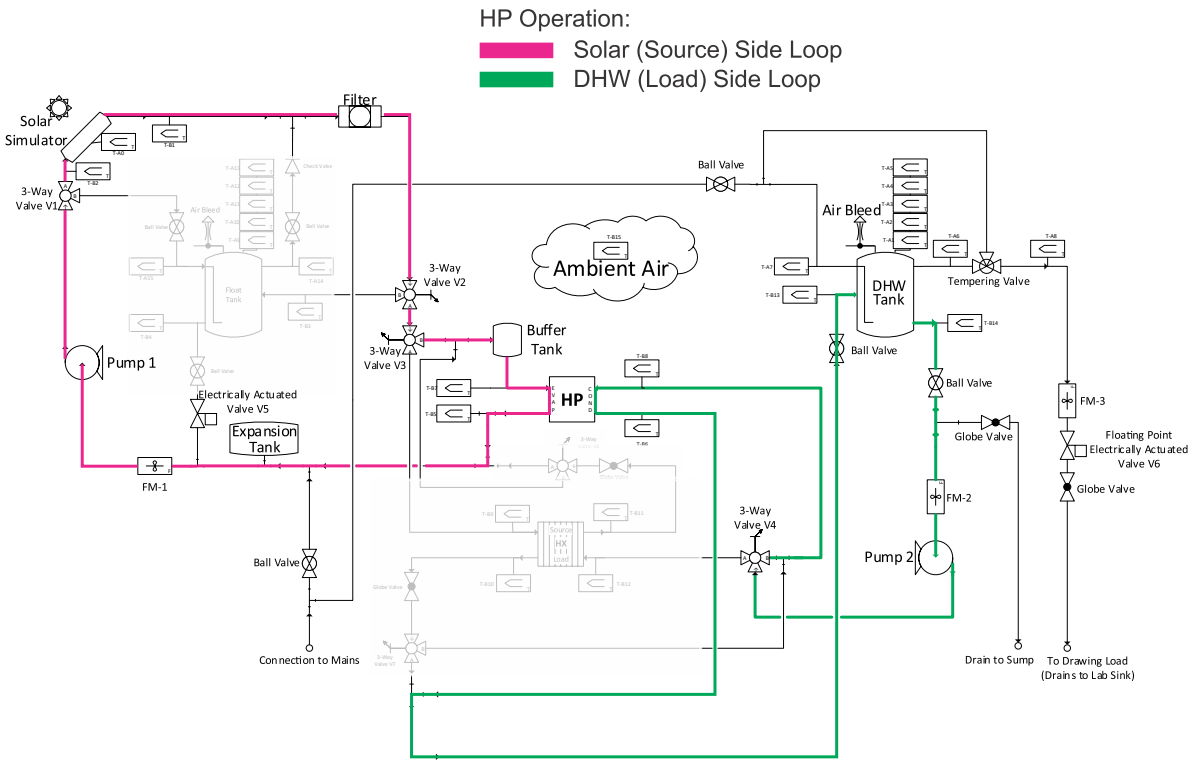


Figure 3-3: Flow path for heat pump operation, ETU behaves as an SAHP system

As seen above, the system can be divided into two plumbing loops. The loop to the left side of the flow schematic is referred to in this work as the "source" side or the "solar" side interchangeably. The plumbing loop on the right side of the diagram is referred to as the "load" side or the "DHW" side of the system. The difference between the flow paths illustrated above is the means of transferring heat from the solar side into the DHW side of the system. Alternately, the ETU can operate as an SDHW system using a HX to connect the source side to the load side, or as an SAHP system using a variable-capacity HP to move heat from the source to the load side.

A number of other flow branches exist in the ETU's plumbing system because future work will likely revisit other system topologies, including the 2-tank topology investigated by Banister [3], and the series-mode SAHP simulated by Chasse, but not experimentally verified [18]. These flow paths are preserved in the ETU, but were disabled during the current work.

3.4 Water Storage Tanks

The ETU is equipped with two water storage tanks manufactured by A.O. Smith in their "Solar Booster Tank" series. The smaller of the two tanks, labeled "DHW Tank" in the system schematics previously shown is a "Sun-80" model tank with a capacity of 302.8 L. A diagram of the DHW tank adapted from the manufacturer's specification sheet is shown below.

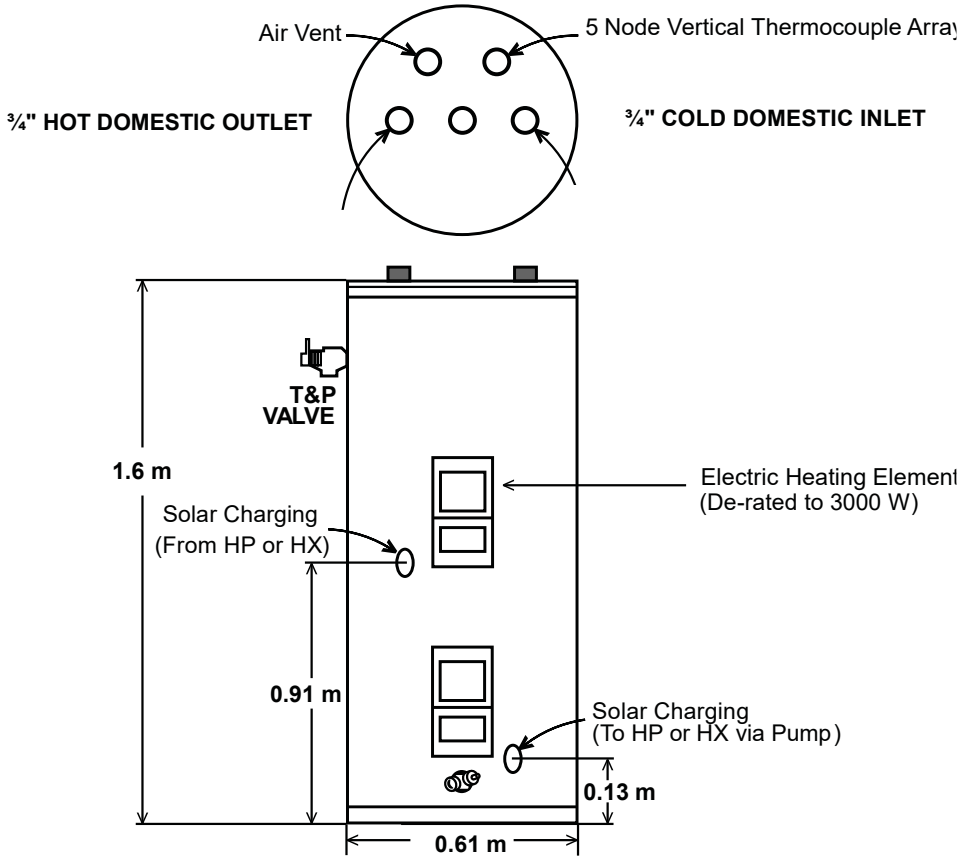


Figure 3-4: Domestic Hot Water Storage Tank Port Locations (Adapted from [38])

As seen in Figure 3-4 , the DHW tank has an overall height of 1.6 m with ports at the top, bottom, and mid-height on the tank. The tank also features an electric resistance heating element which is referred to in this work as the Auxiliary heater (AUX). In the current ETU, the AUX heating element is digitally controlled by a solid-state-relay connected to the LabVIEW DAQ system. The DHW tank has a manufacturer-stated insulation value of R16, which translates to an RSI value of approximately 2.82 m^2K/W [38].

A second tank labeled as “Float Tank” in the system schematics is a “Sun-120” model tank with a capacity of 454.25 L. The float tank remains installed in the ETU from previous investigations and may be incorporated into future work, but it does not form a part of the work currently being described.

3.5 Water Circulation Pumps

The ETU uses two TACO 008-IFC series circulator pumps, one on each side of the heat exchange mechanisms in the system. The performance of a pump is characterized by the amount of flow that it produces at a given head pressure. This relationship is commonly expressed as a pump performance curve, an example of which is provided in Figure 3-5.

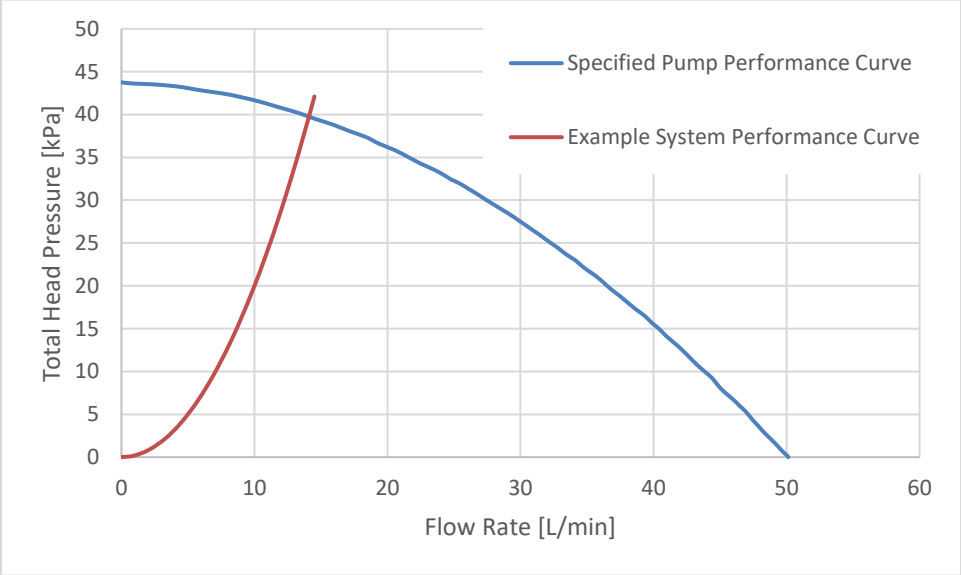


Figure 3-5: Pump performance curve for TACO 008-IFC cartridge circulator [39] and an example system performance curve to show the intersection operating point

The operating point for a pump is determined by the intersection of the pump performance curve with the system performance curve. From the energy equation, it can be demonstrated that the pressure-loss of a piping system is proportional to the velocity, U, of the flow squared [40]. System performance curves are expressed in the form:

$$\Delta P = k_{sys} \cdot U^2 \qquad (3-1)$$

where k_{sys} in the above equation represents a constant of proportionality. The constant, k_{sys}, reflects the flow restriction and frictional losses inherent in the plumbing system, which are a result of losses

in the pipes themselves, as well as losses internal to fittings and fixtures, such as valves, heat exchangers, etc. It can be seen therefore, that in a multi-mode system with various plumbing branches that are switched between in the operation of the system, a different system performance curve would exist for every mode.

The TACO brand pumps used in the system have a variable speed control which enables them to produce more or less flow depending on the control signal that they receive. This enables the system to produce a consistent flow rate in each of the multiple operating modes by shifting the pump performance curve up or down to keep the operating point at the desired flow-rate.

A consequence of the variable-speed pump operation, and the multiple system curves related to the multiple modes of operation, is that the pumps will consume less or more energy at different moments during operation. Power is input to the pumps in electrical form, and the output form is hydraulic mechanical power. The efficiency of this energy conversion can be described in 2-steps as: η_{motor}, the motor efficiency (converting from electrical to shaft power), and η_{pump}, the pump efficiency (converting from shaft power to hydraulic power). For packaged pumps with the motor directly coupled to the pump, it is convenient to define and model a single conversion efficiency: $\eta_{Wire-to-Water}$, which encompasses both of the energy conversion steps. This efficiency can be used to predict the electrical energy consumption of the pump for a given operating point, which will be useful in the simulation of the system.

3.6 Simulated Solar Input

The ETU is equipped with a WATLOW CBEC27J10 immersion heater in a plumbing branch identified as "Solar Simulator". Because the ETU is housed indoors with no connection to a rooftop test platform, the immersion heater stands in for an STC in the test rig. Through an accompanying PID controller, the immersion heater can be configured to either deliver water at a setpoint temperature, or to deliver a demanded heating rate in a range from 0 to 4.5 kW.

The energy collection of an STC is described by Equation 3-2 below, known as the Hottel-Whillier-Bliss (HWB) equation, named for its developers [13, 14].

$$Q_u = A_c F_R [S - U_L (T_i - T_a)] \tag{3-2}$$

In the above equation, Q_u is the rate of solar energy collection. The term A_c is the gross collector area. F_R is the heat removal factor for the collector. The heat removal factor depends on the

configuration of the collector and can be developed theoretically through heat transfer analysis, or can be experimentally determined. S is the amount of incident solar radiation absorbed at the collector plate. This quantity depends largely on the product of G_t, the quantity of solar irradiance transmitted through the Earth's atmosphere to the collector and α_s, the collector surface's absorptivity in the solar (shortwave) spectrum. For glazed flat-plate collectors or tube-collectors encapsulated in a glass envelope, the magnitude of S will also be affected by the transmissivity and internal reflections of the cover glass. A detailed explanation of the calculation to determine S is available in [13, 14].

U_L is a coefficient of thermal loss which is comprised of radiation, convection and conduction heat transfer from the collector plate to the ambient surroundings through all paths available. A common challenge with the application of Equation 3-2 is that U_L varies greatly with different ambient and collector inlet temperatures. As a result, iterative solutions to Equation 3-2 are often required to predict the performance of a collector [14].

The efficiency of an STC is defined as follows in Equation 3-3 [13, 14]:

$$\eta_{coll} = Q_u / A_c G_t \tag{3-3}$$

In the above equation, all of the variables have been previously defined. As seen in Equation 3-2, the rate of heat collection depends on the collector temperature, the ambient temperature, and the irradiation, so it is natural to discuss and plot collector performance in terms of $\frac{T_i - T_a}{G_t}$.

In LabVIEW, a VI called "Solar Collector" is programmed to ingest weather data from an input file and using a STC manufacturer's rated efficiency model, calculate Q_u for the current operating point, taking into account inlet flow rate and temperatures. The heating rate of the immersion heater is then controlled to match the required Q_u, physically simulating a connection to a flat-plate STC.

3.7 Simulated DHW Draws

The ETU is connected to a cold mains water supply which is ported to the bottom of the DHW tank as would be typical of a residential DHW installation. The hot outlet of the DHW tank is mixed through a thermostatic mixing valve for scald protection, set to approximately 55°C. Provided that the temperature of the water at the top of the DHW tank is above 55°C and that the mains temperature is below that temperature, the mixing valve produces a consistent stream at its setpoint by proportionally mixing the DHW and mains inlet water streams. In situations where the water

temperature at the top of the DHW tank falls below the setpoint of the mixing valve, it is expected to pass a stream of 100% DHW and the temperature exits unchanged.

DHW draws are controlled through the use of a solenoid valve for positive shutoff, and an electrically actuated floating diverter valve that provides flow throttling to control the flow rate of a water draw. A "draw profile" file is ingested by LabVIEW containing a time-series of flow rates representing draws throughout a test period. The throttling valve is opened to increase the flow rate, closed to decrease it, and rests when the measured flow rate is within a deadband of ± 0.15 L/min of the setpoint.

Fluctuations in supply pressure from the building mains plumbing affect the flow rates that can be achieved by the DHW draw system. The flow meters in the ETU are specified by the manufacturer to read a range from 0.83 L/min to 75.7 L/min, but the upper range of flow rates that the ETU can attain is often as low as 9 L/min.

Fidelity in the flow rate of a DHW draw is important because of its effects on plumes that develop within the DHW tank which drive tank mixing and the destruction of thermal stratification. The overall volume of water removed in a scheduled draw is also important because of the quantity of heat that is removed with that water. The DHW draw simulation controls the flow rate as close as possible to the scheduled flow rate within the possible range of draws that can be produced, but also counts flow meter pulses to directly track the volume of water that has flowed during a draw event. If the actual flow rate produced during an event is higher than the scheduled flow, then the draw is automatically terminated early when the correct volume of water has been drawn from the tank. If the actual draw flow rate has been lower than the scheduled rate, then the draw period will be extended, again so that the correct volume of water is removed.

3.8 Heat Exchanger

An external braised-plate heat exchanger is installed as a means of transferring heat into the DHW tank from the simulated solar source. The HX is described by the manufacturer as a construction of 25 stainless steel plates brazed with copper and having a heat transfer rating of 44 kW at its rated condition [41], which is for a temperature difference of 60 K at a flow rate of 56.8 L/min.

Heat exchangers can be defined in terms of an overall heat transfer coefficient, UA, which is physically analogous to a product of the thermal conductance of the heat transfer surface and the area

of that surface. A common and general approach to modeling the performance of compact heat exchangers is the *Effectiveness – Number of Transfer Units* method ($\varepsilon - NTU$) [42].

The effectiveness of a heat exchanger, ε, is defined as in Equation 3-4:

$$\varepsilon = {}^{q_{act}}/_{q_{max}} \tag{3-4}$$

where: $q_{max} = C_{min}(T_{H,i} - T_{C,i})$, and C_{min} is the lesser of $C_C = \dot{m}c_{p,c}$, and $C_H = \dot{m}c_{p,,\mathrm{H}}$.

The number of transfer units, *NTU*, is defined as in Equation 3-5:

$$NTU = {}^{UA}/_{C_{min}} \tag{3-5}$$

The effectiveness, *NTU*, and C_r, the ratio of C_{min} to C_{max} have been shown to be related, and for a simple counter-flow heat exchanger, for $C_r < 1$ that relationship is shown in Equation 3-6.

$$\varepsilon = \frac{1 - e^{[-NTU(1-C_r)]}}{1 - C_r e^{[-NTU(1-C_r)]}} \tag{3-6}$$

For a counterflow heat exchanger with the case of $C_r = 1$, the relationship is:

$$\varepsilon = \frac{NTU}{1 + NTU} \tag{3-7}$$

The actual operating point of the HX in the ETU is variable as different temperatures and flow rates will be passed through the unit. Previous students determined a model based on a constant *UA* product was a valid description of this HX's operation in the flow regimes experienced in the ETU. An overall heat transfer coefficient of 5000 kJ/h K was experimentally assigned to the HX in previous student's work [3] [2] and continues to be used in the current study.

3.9 Variable Capacity Heat Pump

This section details an experimental campaign to characterize the performance of a new variable capacity HP. Text and figures in this chapter are reproduced with permission from the author's original work "Experimental Characterization of a Variable Refrigerant Flow Heat Pump for Solar-Domestic-Hot-Water Applications," 2021 [36]. A letter of permission from the publisher can be viewed in the Letter of Copyright Permission section at the end of this document.

As indicated in Section 3.1, a main contribution of this work in terms of the ETU is addition of a previously uncharacterized variable capacity HP. The new HP is a W2W090-VF-UW134A model built custom for the STRL by Ecologix in Cambridge ON. This HP incorporates a variable frequency

drive to control the speed of the HP compressor and in doing so, the capacity of the HP is modulated. The performance of a HP is characterized by its Coefficient of Performance, (COP), the heat transfer rate of each of the heat exchangers, ($\dot{Q}_{cond}$, and $\dot{Q}_{evap}$), as well as its electric power consumption, ($\dot{W}_{el}$). From a 1st law analysis, the following can be shown for steady-state heat pump operation [15]:

$$\dot{Q}_{cond} = \dot{Q}_{evap} + \dot{W}_{el} - \dot{Q}_{loss} \quad (3-8)$$

It can be seen, therefore, that for systems with low losses, the HP is capable of delivering more heat to the load than it consumes in the form of electrical work. The Coefficient of Performance for Heating (COP_h) is an expression used to relate the rate of heating that the device supplies to the amount of power that it consumes. A similar definition can be made for COP_C, which would describe the rate of cooling. COP_h is defined as follows:

$$COP_h = \frac{\dot{Q}_{cond}}{\dot{W}_{el}} \quad (3-9)$$

The theoretical upper limit for the COP_h is defined by the operation of a reversible Carnot Cycle between the condensation and evaporation temperatures in the system [15]. This is defined as follows:

$$COP_{h,Carnot} = \frac{T_h}{T_h - T_c} \quad (3-10)$$

In practice, due to stray heat transfer and internal irreversibility, heat pumps often only achieve 50% to 70% of their theoretical maximum COP, but for a range of useful temperatures for water heating, this still commonly results in COP values in the range of 2.5 to 4.5.

Due to the temperature dependence of the COP, as well as the real thermo-physical properties of the refrigerant used in the system, R-134a, it cannot be assumed that the heat pump being studied would operate at any constant COP or $\dot{W}_{el}$ through the range of operating conditions required to charge a DHW tank using a solar input. As a result, the heat pump must be characterized over the range of possible operating conditions. The experimental characterization of this HP is detailed in the next several subsections.

3.9.1 Method

The independent variables that affect the performance of the heat pump can be thought of as any data that define the state of the inlet fluids on the source and load sides of the heat pump, as well as the compressor control signal, which dictates the speed at which the VFD drives the compressor. Therefore, 5 independent input variables are defined as follows in Table 3-2

Table 3-2: Independent input variables for HP performance mapping

Name	Symbol	Expected range	Range Limited by
Evaporator-Side (Source) Mass Flow Rate	$\dot{m}_{Source}$	6 to 10.75 [kg/min]	ETU water pumps
Condenser-Side (Load) Mass Flow Rate	$\dot{m}_{Load}$	6 to 10.75 [kg/min]	
Source Temperature Inlet	$T_{s,i}$	10 to 60 [°C]	Freeze and scald
Load Temperature Inlet	$T_{l,i}$	10 to 60 [°C]	protection
Compressor Speed Signal	N_{comp}	50% to 100%	Compressor stall

As discussed in the previous subsection, the performance of the heat pump is described by the heat transfer rates at the evaporator and condenser, and the electrical power consumption, $\dot{Q}_{evap}$, $\dot{Q}_{cond}$, and $\dot{W}_{el}$. The coefficient of performance and stray heat losses, COP and $\dot{Q}_{loss}$, can simply be calculated from the first three quantities using Equations 3-9 and 3-8.

The ETU is capable of controlling and holding relatively constant 4 of the 5 independent variables in Table 3-2. The flow rates are controlled by a proportional signal sent from LabVIEW through the analog control card to each of the pumps. Using the flow-meters installed in series with the pumps on each side of the heat transfer loop (source side and load side), flow rates are numerically selectable and automatically maintained within approximately ± 0.5 kg/min of the commanded value.

The temperature of the fluid inlet to the source-side (evaporator) of the heat pump is heated by the immersion heater. In practice, the immersion heater is able to maintain a steady input temperature that typically deviates less than $0.1°C$ from the set point temperature during steady operation.

The compressor speed signal is a current signal from 4 to 20 mA that is linearly mapped from 0% to 100% of line frequency. The maximum commanded compressor speed of 3600 RPM corresponds to the 20 mA signal and a driving frequency of 60 Hz. This control signal is a set point within the LabVIEW Control Panel and is generated by the DAQ analog output card.

The final input parameter is the load-side (condenser) fluid temperature, which is not easily controlled in the ETU. The temperature would naturally range from 10°C to 60°C in a typical DHW system, where the lower temperature represents the mains water input temperature, and the upper temperature represents the maximum storage set point temperature expected in such a system. When operating the ETU, the load-side fluid temperature is dictated by the temperature of the water strata stored in the DHW tank. The tank outlet that feeds the heat pump is at the bottom of the tank, so it is typically the coolest water in the tank that passes to the inlet on the load side. Because the load-side temperature cannot be directly controlled, the experiment was designed to run through the entire range of expected temperatures by gradually heating the DHW tank, while running the heat pump and recording performance data. The rate of change of the DHW tank's bottom outlet temperature is slow enough that this performance can be regarded as quasi-steady, even though this independent variable is not directly controlled.

It is desired to capture the individual effects of each input variable, as well as the combined effects that changes of multiple input variables at a time would have on the modeled performance of the heat pump. As such, a modified Fractional Factorial design was selected, as described in Antony's reference text on Design of Experiments [43]. A modification had to be made to the experimental design because of the inability of the ETU to directly control the load-side inlet temperature to the heat pump. The treatment levels (tested values) of each of the input variables are listed below in Table 3-3.

A series of 50 unique trials were conducted with every permutation of the compressor speed and temperature treatment levels. Duplicate trials were conducted for each condition to ensure repeatability and redundancy of the data in case of interrupted tests or corrupted data files.

Table 3-3: Treatment levels for input variables to heat pump mapping experiment

Symbol	Unit	Treatment Level					
		1	2	3	4	5	6
$\dot{m}_{Source}$	$[kg/min]$	6.000	8.375	10.75	-	-	-
$\dot{m}_{Load}$	$[kg/min]$	6.000	8.375	10.75	-	-	-
$T_{s,i}$	$[°C]$	10	20	30	40	50	60
$T_{l,i}$	$[°C]$	Continuously Variable from 10 to 60 $°C$.					
N_{comp}	[%]	100.0	87.5	75.0	62.5	50.0	-

A further 54 unique trials were conducted at $N_{comp} = 100\%$ to cover all of the possible permutations of temperature set points and flow rate set points. Duplicate trials were once again conducted for the same reason as above. An analysis of the results of these trials is discussed in the following section.

3.9.2 Results

Results of the heat pump characterization trials were analyzed using Rave, a data visualization and model fitting toolbox that was developed for MATLAB and made available from Georgia Tech Research [44].

The result of any one set of trial conditions is a set of 2-dimensional performance curves that traverse the load-side inlet temperature and the electric work, source heat transfer, and load heat transfer rate spaces. Using Equation 3-9, the COP of the heat pump is also calculated and plotted. Figure 3-6 below shows, as an example, the performance curves generated for trials with a source inlet temperature of $40°C$, source and load flow rates of 10.75 kg/min each, and 100% compressor speed.

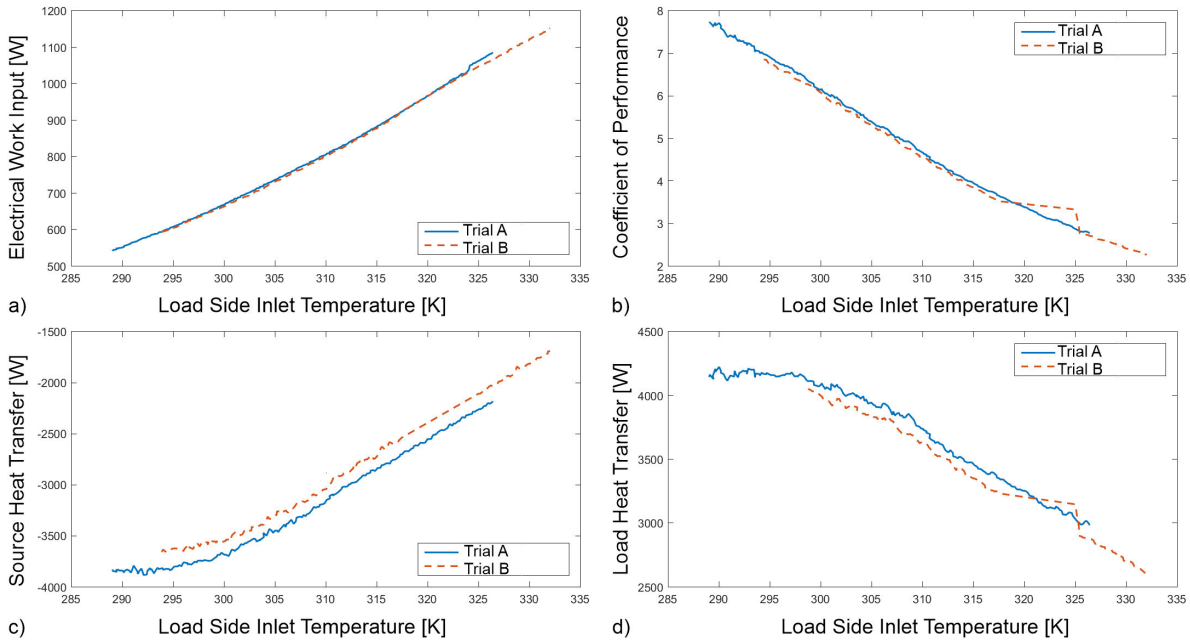


Figure 3-6: HP mapping trials at N_c=100%, $T_{S,i}$=40°C and $\dot{m}_S$=$\dot{m}_L$= 10.75 kg/min showing: a) $\dot{W}_{el}$ vs. $T_{l,i}$, b) *COP* vs. $T_{l,i}$, c) $\dot{Q}_s$ vs. $T_{l,i}$, d) Q_L vs. $T_{l,i}$

The repeatability of the experiment was assessed by comparing duplicate trials. The following values are expressed first as an average of the absolute differences in the natural units recorded during the experiment followed by an average of the percentage-differences of each observation, computed as follows:

$$\text{Average Absolute Difference} = \frac{1}{n}\sum_{i=1}^{n}|x_{1i} - x_{2i}| \qquad (3-11)$$

$$\text{Average Percent Difference} = \frac{1}{n}\sum_{i=1}^{n}\frac{|x_{1i} - x_{2i}|}{(x_{1i} + x_{2i})/2} \times 100\% \qquad (3-12)$$

The repeatability of the tests is seen to be quite good in terms of the electrical work input pictured in Figure 3-6 a) and COP pictured in b), which differ from duplicate trial-to-trial by an average power of ±8.9 W or approximately ±1% and an average COP of approximately ±0.1 or ±2%. For the heat transfer rates, Source-Side (Evaporator) pictured in Figure 3-6 c) and Load-Side (Condenser) pictured in d), an average difference between trials is observed to be ±138.0 W or ±4% and ±98.6 W or ±3% respectively. The agreement between the duplicate trials displayed above is characteristic of the various trials that were described in the methodology section, indicating that this experiment is sufficiently repeatable to be used to produce a reliable model of the heat pump's performance.

3.9.3 Analysis

Combining all of the trial data for a single compressor-speed and flow-rate, 3-dimensional performance surfaces can be generated to illustrate the performance of the heat pump throughout the source-side and load-side inlet temperature spaces. Such surfaces are plotted in Figure 3-7.

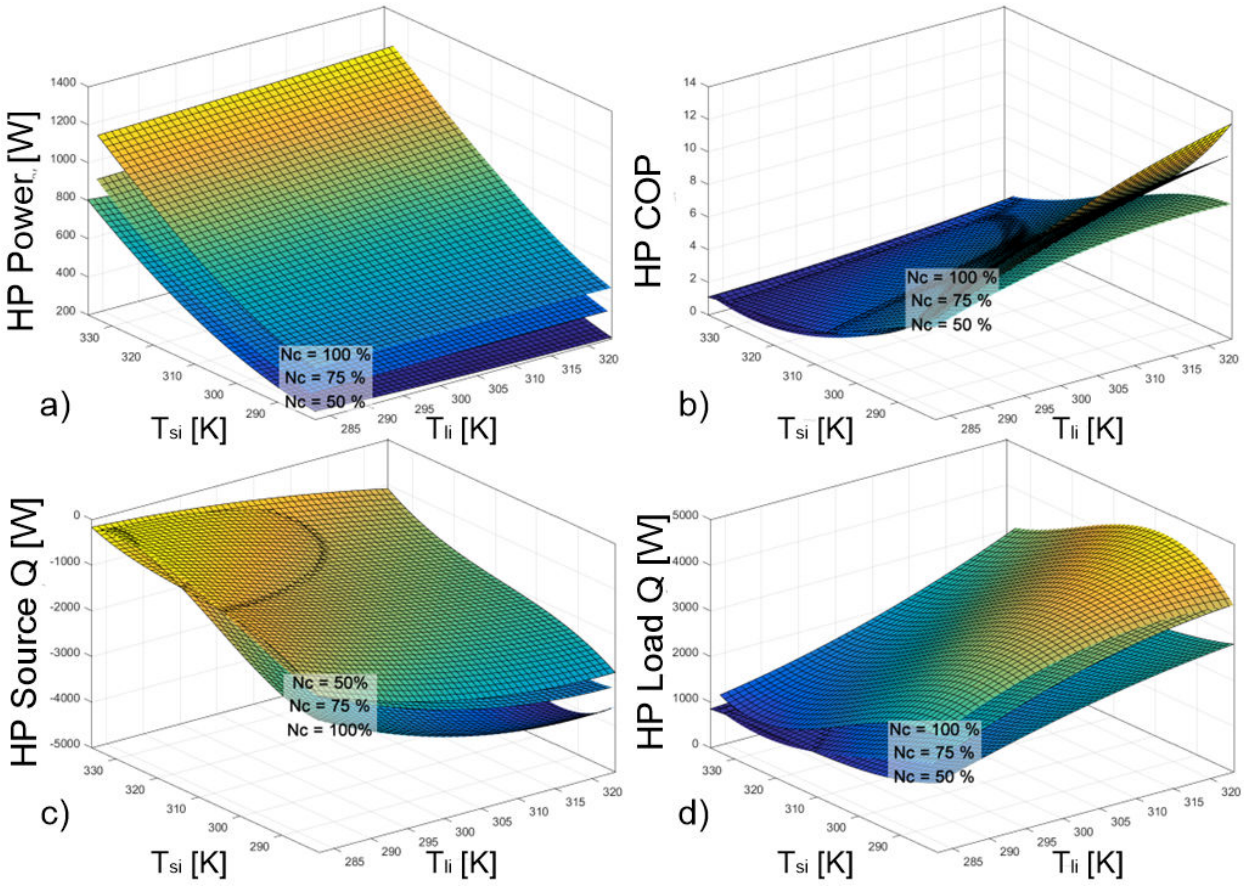


Figure 3-7: Plotted results of aggregated trials across three compressor speed set-points, N_c = 100%, 75%, and 50%, showing: a) $\dot{W}_{el}$ vs. $T_{l,i}$ vs. $T_{s,i}$, b) COP vs. $T_{l,i}$ vs. $T_{s,i}$, c) $\dot{Q}_s$ vs. $T_{l,i}$ vs. $T_{s,i}$, d) $\dot{Q}_L$ vs. $T_{l,i}$ vs. $T_{s,i}$

In Figure 3-7 a) it can be seen that the power consumption of the HP is strongly dependent on the load-side inlet temperature and the commanded compressor speed, but much less sensitive to the source-side temperature. This is very likely due to the high refrigerant pressure that corresponds with having a hot condenser, causing the compressor work rate to increase.

In Figure 3-7 b) it can be seen that the minimum COP of 1 is achieved in conditions of a very high temperature difference between the source and load sides. This observation is consistent with the expected behavior predicted by the temperature dependence of the COP of the Carnot HP Cycle defined in Equation 3-10. For the majority of the range of COP plotted in b), the three surfaces are largely overlapping. This demonstrates a weak dependence of COP on the commanded compressor speed. However, in conditions of high source-side inlet-temperature and low load-side inlet temperature, the three surfaces diverge with the $N_c = 100\%$ surface producing the largest COP, and $N_c = 50\%$ producing the smallest. This suggests that in these high temperature-difference conditions, the limiting factor in the cycle's ability to move heat is the mass flow rate of the refrigerant, rather than the rate at which heat can be transferred to that refrigerant within the condenser and the evaporator, leading to a higher dependence on the compressor rotor speed.

Figure 3-7 c) and d) show the heat transfer rate at the source (evaporator) and load (condenser) sides respectively. The surfaces are nearly mirror reflections of one another about the horizontal plane, which demonstrates agreement with the 1st law energy balance described in Equation 3-8. The response of the system to changes in commanded compressor speed is seen as a higher absolute-value rate of heat transfer on both the source-side and the load-side when the compressor is commanded at a higher speed.

The lowest possible heat transfer rate in both cases is 0 W, which appears to be approached asymptotically from both the positive and the negative side as the load-side temperature exceeds $330\ K$. The refrigerant used in the HP under study, R134a, has a critical temperature of approximately $374\ K$ [45]. As the condensation and evaporation temperatures approach this critical temperature, the amount of heat transfer that can take place as the refrigerant changes phase within the condenser and the evaporator is significantly reduced. When the refrigerant is near its critical point, the compressor works very hard to effect any phase-change at all. Under these conditions, the COP of the HP is very low, and the majority of the heat transfer observed at the condenser is

attributed to compressor-heating, rather than the removal of heat from the source fluid at the evaporator.

Aggregating all of the HP characterization trials at the various compressor speeds, water flow rates, and inlet temperatures, the performance of the HP is modeled according to the 5 independent variables listed in Table 3-2. Three multivariate polynomial equations were generated to calculate $\dot{W}_{el}$, $\dot{Q}_S$, and $\dot{Q}_L$. The COP and Q_L terms are later calculated using Equations 3-9 and 3-8.

The three polynomials have full-factorial terms up to order 3, as described in [43]. The input variables to the model are $\dot{m}_1, \dot{m}_2, T_{si}, T_{li}, and\ N_{comp}$, the mass flow rates on the source and load side, the temperatures on the source and load side inlets, and the commanded compressor speed, respectively. A separate set of coefficients correspond to each of the 3 above performance parameters. The listing of coefficients, as well as a full-form of the characteristic polynomial equation are expressed in Equation B-1 and table B-1 in Appendix B to this document.

The polynomial surfaces are evaluated in terms of the coefficient of determination (R^2), the Root-Mean-Squared Error (RMSE), and the Mean Error. The models were validated and tested with data kept separate from the data used to train the model. The fit statistics are detailed in the Table 3-4 which shows both the fit-data and validation-data sets in adjacent columns. The quality of the model-fits are both plotted and discussed on the following pages in Figure 3-8, Figure 3-9, Figure 3-10, and their accompanying text.

Table 3-4: Fit Statistics for HP Power, Source Heat Transfer, and Load Heat Transfer Polynomial Surfaces

	HP Power ($\dot{W}_{el}$)		**Source Heat Xfer ($\dot{Q}_s$)**		**Load Heat Xfer ($\dot{Q}_l$)**	
Data Set	Fit	Validation	Fit	Validation	Fit	Validation
R^2	0.998	0.998	0.992	0.99186	0.988	0.988
$RMSE^*$	13.83	13.72	137.3	137.7	154.7	154.1
$Mean\ Error^*$	6.75×10^{-5}	-0.0188	-0.00124	0.500	0.00100	-0.286

* Units for RMSE and Mean Error: [W]

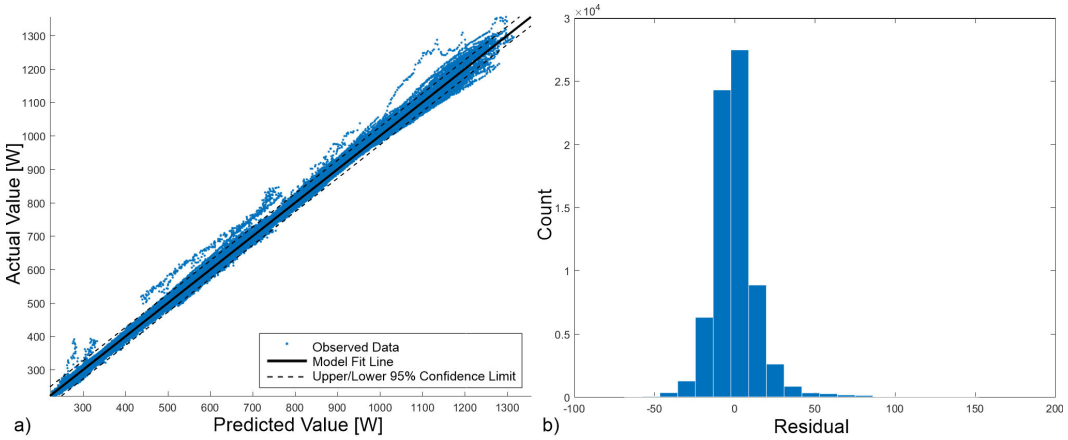


Figure 3-8: HP Power Model, a) Observed vs. Predicted Values; b) Residuals Histogram

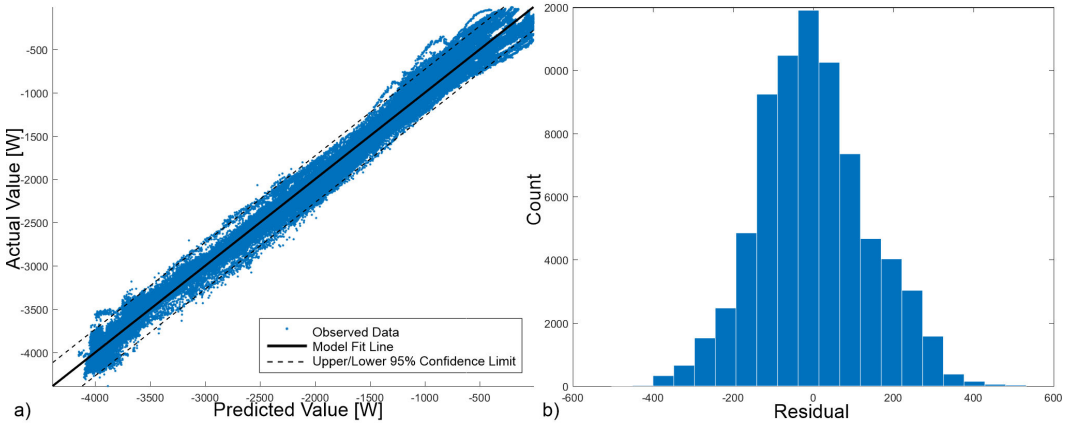


Figure 3-9: HP Heat Q_s Model, a) Observed vs. Predicted Values; b) Residuals Histogram

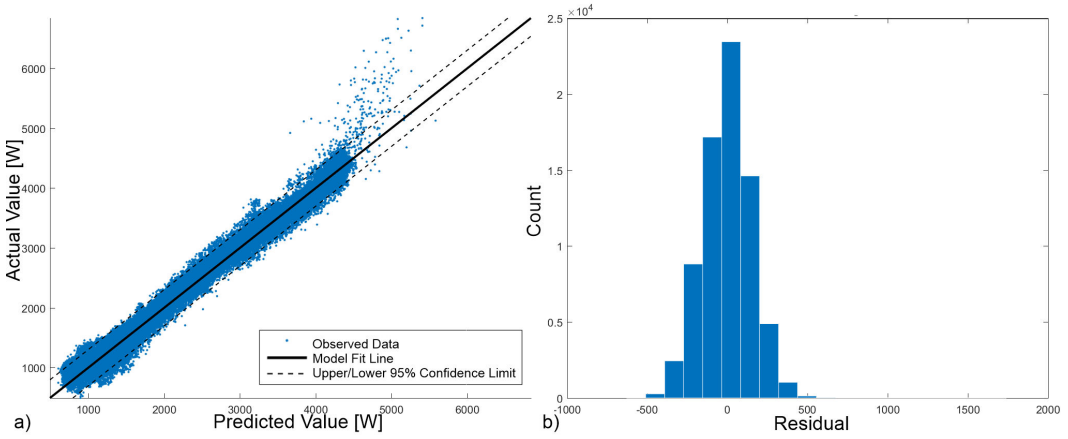


Figure 3-10: HP Q_l Model, a) Observed vs. Predicted Values; b) Residuals Histogram

Analysis of the residuals histograms in the above figures shows a tight grouping, relatively little skew, and an approximately normal distribution of the fitting errors, which suggests that the 3rd order of the polynomial surfaces is sufficiently high to capture the effects of the input variables.

A separate validation data set was used to evaluate whether the degree of the model functions is too high and thus whether overfitting has occurred. For all 3 models, the coefficients of determination were nearly identical between the fitting and the validation sets. The RMSE of each model also agreed, each with a difference of less than 1% between fitting and validation data sets.

The upper and lower limits of prediction error at 95% confidence for the HP Power Consumption model ($\dot{W}_{el}$) are approximately $\pm 27.02\ W$ which represents an error of 12% at the minimum expected power consumption and 0.19% at the maximum expected power. In the average range of operation, the error expected due to statistical uncertainty would be less than 5%.

For the HP Source-Side Heat Transfer Model ($\dot{Q}_s$), the 95% confidence limits on model prediction would be approximately $\pm 269.5\ W$, which would represent a large range of errors from above 100% down to 6%. Over the average range of system operation, the error is expected to be less than 15%. It should be noted that the higher end of the error expected in this model corresponds to operating conditions where the actual value of heat removed from the source loop by the HP is very near to $0\ W$. In these conditions, where the HP moves very little heat, it would not be desirable to operate the heat pump and system controllers would be programmed to avoid operating the heat pump under these conditions. For that reason, the high relative error at low heat transfer rates is not expected to have a significant impact on simulation results that use this model.

Finally, for the load-side heat transfer rate, a very similar statistical uncertainty of approximately $\pm 302.9\ W$ exists. Again, in the context of percent error, the maximum expected error, which would occur when the heat pump is moving very little heat at the load-side, is 61%. The minimum percent error, which would correspond to when the heat pump is producing its maximum observed heat transfer rate is approximately 4.4%. Over the average range of operation of the model, the statistical uncertainty would be expected to contribute an error of less than 13%.

It should also be noted that in all three models discussed above, the uncertainties describe the variation of individual operating points from the mean response. Again, observing the residuals, by far the most common residual is within the first bin around 0. In addition, the balance between

positive and negative residuals suggests that errors will be self-canceling if the model is operated over a wide enough range of conditions with a large enough set of samples.

3.9.4 Conclusions

In summary, it can be seen above that a campaign of data acquisition and model-fitting has been conducted to generate a characteristic model of the operation of a custom-built variable speed HP.

Trends in heat transfer rates and electrical power consumption were explored and it was shown that there is a strong dependence of performance on the source-side inlet temperature. The HP compressor speed control was shown to have a strong influence on power consumption and heat transfer rates, but because these two quantities scaled together, the compressor speed does not strongly influence the *COP* except at the far limits of the domain of temperatures investigated.

Further investigation into the performance of variable speed HPs would be advised to isolate the effects of each individual control variable. It would be prudent to examine these effects across multiple models and capacities of variable HP to determine whether general modeling equations can be developed.

The results of this particular study are three multivariate polynomial response hypersurfaces that describe the unit's $\dot{W}_{el}, \dot{Q}_s, and\ \dot{Q}_l$. These hypersurface fits can be used to model the performance of the variable HP within 5% for power consumption, 15% for source-side heat transfer, and 13% for the load-side heat transfer rates. Although these uncertainties are significant, they are small enough to permit the implementation of the model in a predictive controller or for preliminary simulation studies, provided that the models are validated over the range of conditions being simulated. This would be consistent with the approaches of other researchers [3, 2]. It is also worth noting that the uncertainty in model predictions described above are of a similar magnitude to the uncertainties predicted by others in analytical work relating to boiling and evaporative heat transfer with refrigerants [46].

The final product of this HP characterization experiment is the incorporation of the model presented above into a custom TRNSYS "TYPE" model that is detailed in the next chapter's Section 4.3. This custom TRNSYS TYPE is then incorporated into an overall system simulation model described in the following chapter and used to generate the results discussed throughout the remainder of this thesis.

Chapter 4
System Simulation Model

This chapter describes an overall system model developed in TRNSYS, a transient system simulation tool. The model described here is used in subsequent chapters to predict system behavior under various modes of operation and control schemes. The model and its validation are treated in this chapter. Text and figures in Sections 4.1, 4.2, and 4.4 are reproduced with permission from the author's original work, "Annual performance of a simulated multi-mode SDHW/SAHP system with various control options," 2023 [37]. A letter of permission is included at the end of this document.

4.1 Simulated System

It was desired to create a simulation model that represents the system of interest described briefly in Section 1.3.1 of the introduction chapter. For clarity, the figure from that section is repeated below. This system is physically embodied in the ETU, which lends itself to short-scale experiments, but a numerical-analytical model will prove useful for prediction of the system's performance over longer timeframes and for exploration of alternative control strategies.

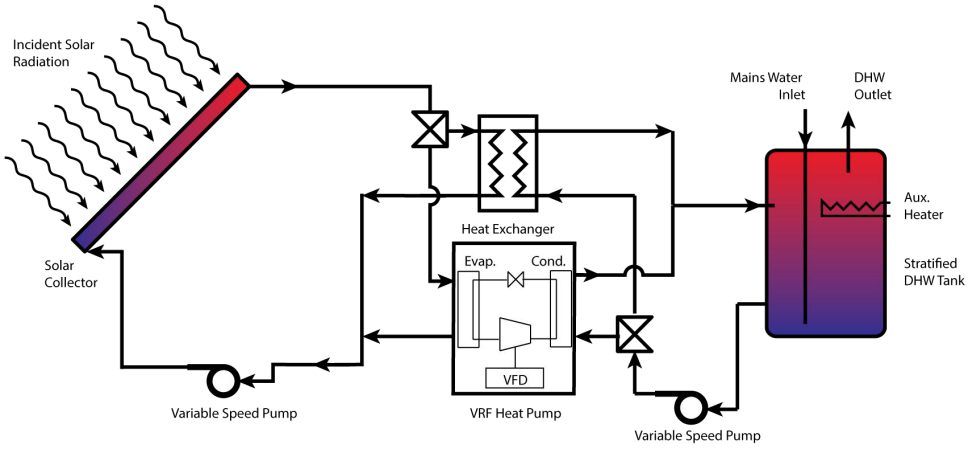


Figure 4-1: System of interest (Figure 1-11 repeated)

In the figure above, the 3-way valves can be arranged so that the following operating modes are available, mimicking the behavior of the ETU:

1. Solar domestic water heater with an external heat exchanger (HX Mode, 1)

2. Indirect-expansion-solar-source-heat pump water heater (HP Mode, 2)

3. Activating the circulation pumps with flow diverters set to the HP's evaporator and condenser without calling for the HP compressor can be considered a "warmup" mode where fluid is circulated to the collector, gaining heat in the solar-side loop prior to HP or HX activation (Warmup Mode, 3).

In addition to the topological layout of components for the system, a controller is required to provide inputs to the diverter valves, circulator pump, heat pump, and the separately controlled AUX Heating element. The system model is developed to accept controller input from a variety of system controllers which can be substituted for different purposes.

4.2 TRNSYS Model

Simulation was performed in TRNSYS using the overall simulation model pictured in Figure 4-2. Blue solid lines in the figure below show water flow paths, while orange dashed lines show data links between the various model components.

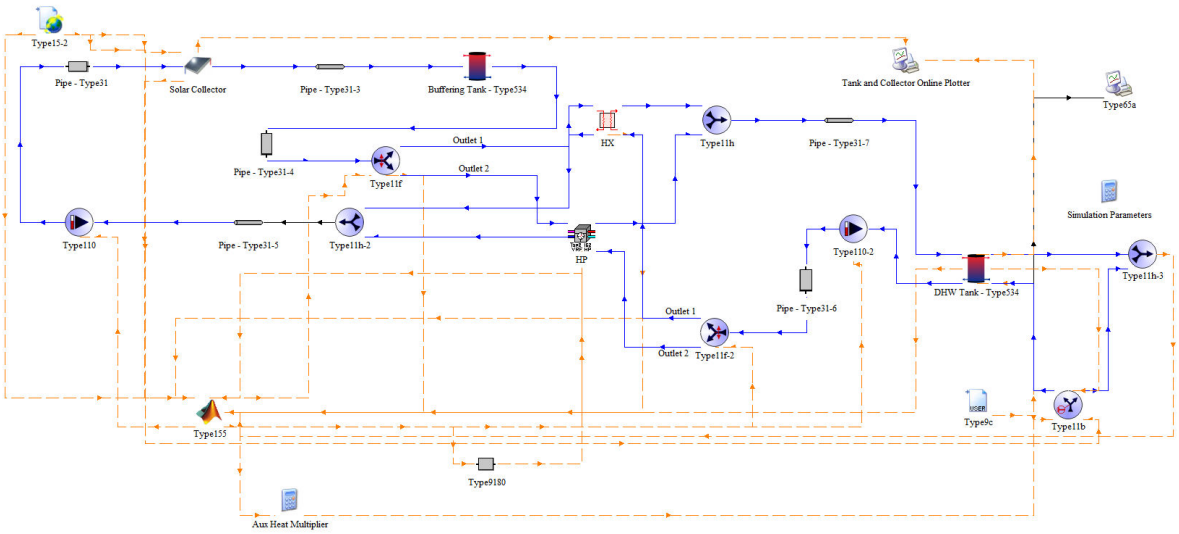


Figure 4-2: TRNSYS model constructed to represent the system being simulated

The individual system components are simulated by TRNSYS "TYPE" models. The TYPE models and important parameters are summarized in Table 4-1. Fluid properties are for water with a constant specific heat of c_p =4.19 kJ/kgK and a density of $\rho = 998$ kg/m^3. Unless otherwise specified, any additional properties for water are taken at room temperature and 101.325 kPa pressure. Components not listed in the table are configured using default TRNSYS parameters.

Table 4-1: TRNSYS Types used and corresponding simulation parameters

Component	TRNSYS TYPE	Simulation Parameters
Circulation Pumps	TYPE 110	Rated Flow Rate: 634 kg/h
		Rated power: 76.13 W
Storage Tank	TYPE 534	Volume: 364 L (80 gal)
		Nodes: 20
Solar Thermal Collector	TYPE 1	Number in series: 2
		Collector area: 4.988 m^2
		Fluid specific heat: 4.19 kJ/kg K
		Efficiency mode: 1
		Tested flow rate: 72.17 kg/hr m^2
		Intercept efficiency: 0.769
		Efficiency slope: 13.0104 kJ/hr.m^2 K
		Efficiency curvature: 0.048888 kJ/hr m^2K^2
		Optical mode: 2
		1st-order IAM: 0.32
		2nd-order IAM: 0
Heat Exchanger	TYPE 5b	Heat Exchange Coefficient: 138.8 kW/K
MATLAB Controller	TYPE 155	Described in section 4.4.
Heat Pump	Custom TYPE 1162	Described in section 4.3.
Weather Reader	TYPE 15	Weather File: CA-ON-Ottawa-716280.tm2

4.3 Type 1162 Custom HP Model

The overall system simulation relies on a custom HP model of the Ecologix W2W090-VF-UW134A unit equipped with a 3-phase scroll compressor and a VFD described in Section 3.9. This was implemented in TRNSYS as a custom "TYPE 1162" model written in the TRNSYS default programming language FORTRAN.

The model uses a multivariate polynomial surface which is fit to experimental performance data to generate the predicted performance outputs listed in Table 4-3 based on the inputs and parameters listed in Table 4-2.

Table 4-2: Type 1162 Required Inputs and Parameters

Variable Name	Units	Description
$c_{p,source}$	kJ/kg.K	Specific heat of the source-side fluid
$c_{p,load}$	kJ/kg.K	Specific heat of the load-side fluid
$HP\ Call$	Boolean	Command the HP on = 1
N_{comp}	Dimensionless	Linear map of rotor speed (0 =0 rpm to 1=3600 rpm)
$Source\ Flow\ Rate$	kg/hr	Mass Flow Rate of the fluid flowing from the source (e.g. from a solar collector)
$Load\ Flow\ Rate$	kg/hr	Mass Flow Rate of the fluid flowing to the load (e.g. Storage Tank)
$T_{Source,In}$	°C	Temperature at the source-side inlet
$T_{Load,In}$	°C	Temperature at the load-side inlet

Table 4-3: Type 1162 Calculated Outputs

Variable Name	Units	Description
$\dot{W}_{el}$	W	Input Power
$\dot{Q}_{source}$	W	Source Heat Removal Rate
$\dot{Q}_{load}$	W	Load Heat Addition Rate
COP_h	dimensionless	Coefficient of Performance for the HP system
$T_{Source,Out}$	°C	Temperature at the source-side outlet
$T_{Load,Out}$	°C	Temperature at the load-side outlet

While running, the HP model also performs checks to ensure that simulations remain within the range of input values tested during the HP characterization experiments. These checks are as follows:

- If input temperatures or flow rates are outside of the characterized range, the value for that input is temporarily coerced to the nearest in-range input and the HP model generates a prediction for $\dot{W}_{el}$, $\dot{Q}_{source}$, and $\dot{Q}_{load}$ based on the temporary input value. Using the original out of range input value, along with the predicted heat transfer rates, a new outlet temperature can be calculated as if the input variable had been in range
- If any temperature in the inputs or outputs of the model approaches the boiling or freezing point of water, the model switches the HP compressor off and passes outputs as if $\dot{W}_{el}$, $\dot{Q}_{source}$, and $\dot{Q}_{load}$ are all equal to 0. Source code for the Type 1162 Variable Speed Water-to-Water Heat Pump is included as Appendix C to this document.

4.4 TYPE 155 MATLAB Custom Controller

TRNSYS is able to call MATLAB through a COM interface using TYPE 155 in the TRNSYS model and a custom MATLAB script, the output of which sends commands to the TRNSYS TYPE for the flow diverters, circulation pumps, HP compressor, and the AUX heater.

A great deal of flexibility is achieved using the MATLAB controller rather than another custom TYPE or pre-packaged controller option available in TRNSYS. These advantages include:

- Access to a wide array of built-in MATLAB functions and libraries, rather than coding every method from scratch in TRNSYS's default language, FORTRAN
- Avoids the need to re-compile code after every edit.
- Dynamic memory allocation, allows intermediate variables to be declared on the fly. Where necessary, arrays can change size easily to match the requirements of the parent simulation (time-step, simulation bounds, etc.).

A series of MATLAB controllers were written over the course of the work. Ultimately two branches of the custom controllers were utilized in simulations presented in future chapters. They are a Temperature-Differential based controller used in developing benchmark performance data and a Predictive-Controller based on draw-energy used to explore what performance improvements can be realized through controls development. The control algorithms are described in detail in subsequent chapters. Published MATLAB code can be viewed in Appendix D.

4.5 Forcing Functions

The simulation model is forced by two data inputs represented in TRNSYS by a TYPE 15 data reader which handles weather and solar irradiation inputs and a TYPE 9c data reader for DHW draw flows.

Inputs sourced from the TYPE 15 data reader include: ambient air temperature, incident radiation (horizontal, diffuse, and tilted), ground reflectance (albedo), solar radiation angle of incidence, and mains water temperature. These data are extracted from the Typical Meteorological Year (TMY) file for Ottawa, ON, which is constructed from a statistical treatment of climatological data to produce average weather conditions for each day of the year.

The flow rate data are supplied at 5-minute resolution in a CSV input file. Flow rates are read into the simulation and interpolated to match the by TRNSYS time-step by a TYPE 9c data reader. By using an input data file for DHW draws, the model is adaptable to a variety of draw schedules. In preliminary work a draw profile from the Canadian Standards Association (CSA) standard CAN/CSA-F379 Series-09 [47] was employed to characterize the operation of a system on a sample day of the year. Later as work expanded in the time domain to simulation of a full year. It was desirable to use a realistic and non-repeating draw schedule as an input to the DHW flow rate forcing function, so a draw schedule from Edwards [48] was employed. More details on the draw schedules will be shown adjacent to simulation results for which they were used.

4.6 Initial Conditions

It was desired to produce a starting condition for the model that could be approximately reproduced in physical experiments. The component of the TRNSYS model with the largest thermal capacity, and therefore highest importance of initial condition is the DHW storage tank, modeled by a TRNSYS TYPE 534 component configured to have 20 vertical tank nodes. Preliminary investigations showed that a common condition for the DHW tank was a partially charged, stratified condition. This condition could be experimentally replicated by charging the ETU's DHW tank to a mixed temperature, then drawing off a percentage of the tank volume from the top of the tank and replacing it with mains-temperature water at the bottom.

Beginning with a mixed tank at 50°C and drawing half of the tank volume (150 L) produces the temperature distribution shown in Figure 4-3 which is used as initial conditions for the DHW tank in the system model in all simulations unless otherwise stated.

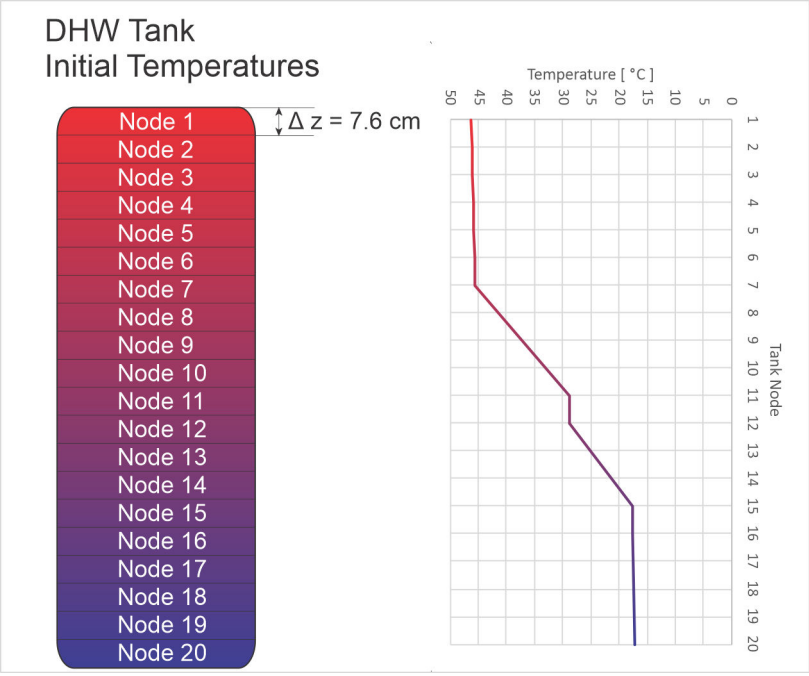


Figure 4-3: Initial temperature profile of DHW tank in TRNSYS simulations

The initial conditions for all other components in the system were chosen to best represent a system that had been inactive and come to thermal equilibrium with its surroundings. Therefore, the initial temperature condition for other components of the system is 20°C, the initial flow rate for all components is 0 kg/hr and initial control signals are equivalent to the power being switched off and components being in an inactive state.

4.7 TRNSYS Model Validation

This section describes steps taken to validate the overall system TRNSYS model and verify the preliminary simulation results obtained. These steps included characterization of individual model components, analysis of the simulation's time step sensitivity, and validation experiments comparing the simulated output to results obtained from the ETU.

4.7.1 Characterization of Components

Components in the overall system model were sized and configured to match as closely to their analogous real-world components built into the ETU in the lab. Many components, including the storage tank, circulator pumps, and HX were characterized and modeled as part of previous student's

work in the lab [2, 3]. Other components, such as the diverter valves and AUX heaters were configured according to the manufacturer's specifications.

Single-component validation tests were conducted to ensure that the performance of previously characterized equipment had not deteriorated or changed appreciably. These miniature validations included a multiple-day temperature decay test on the DHW tank. The circulator pumps were verified by random sampling of various speeds with flow rates being measured by the ETU flow meters, and electrical power input to the pumps measured by the ETU power metering transformers.

Since the DHW tank model is a large component in the system model, some discussion of the model's performance is warranted. Results from the temperature decay test are seen in Figure 4-4 and Figure 4-5. The experiment was conducted by charging and mixing the DHW tank on the ETU to 51°C then switching all pumps and heaters off and activating the DAQ to record over the course of 48 hours. A matching simulation was conducted with the tank starting at a well-mixed 51°C, the ambient conditions were set to match the room temperature observed during the experiment.

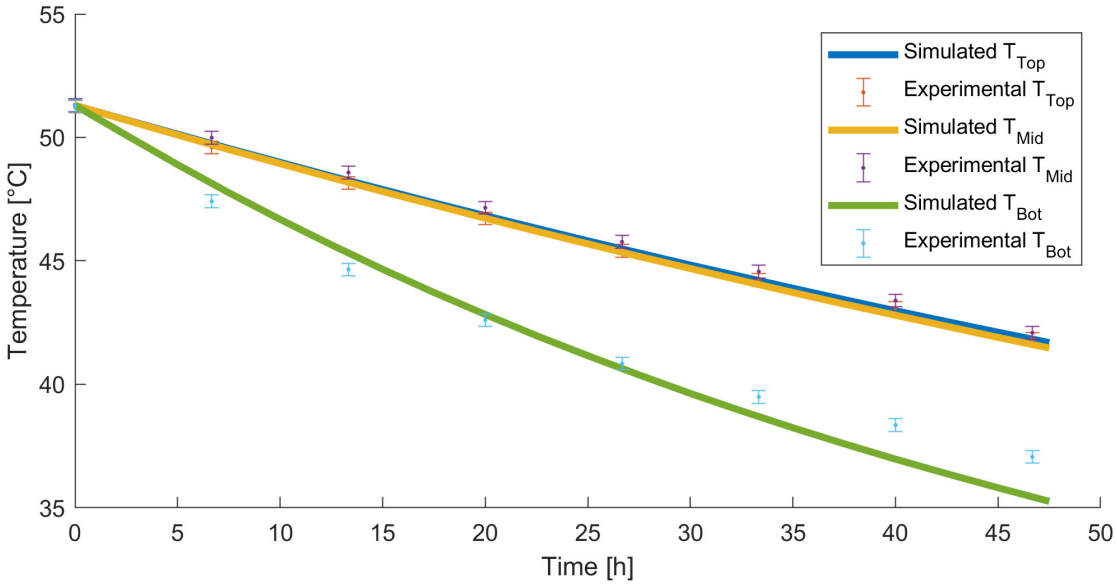


Figure 4-4: Simulated and Experimental DHW Tank Temperatures (Top, Middle and Bottom)

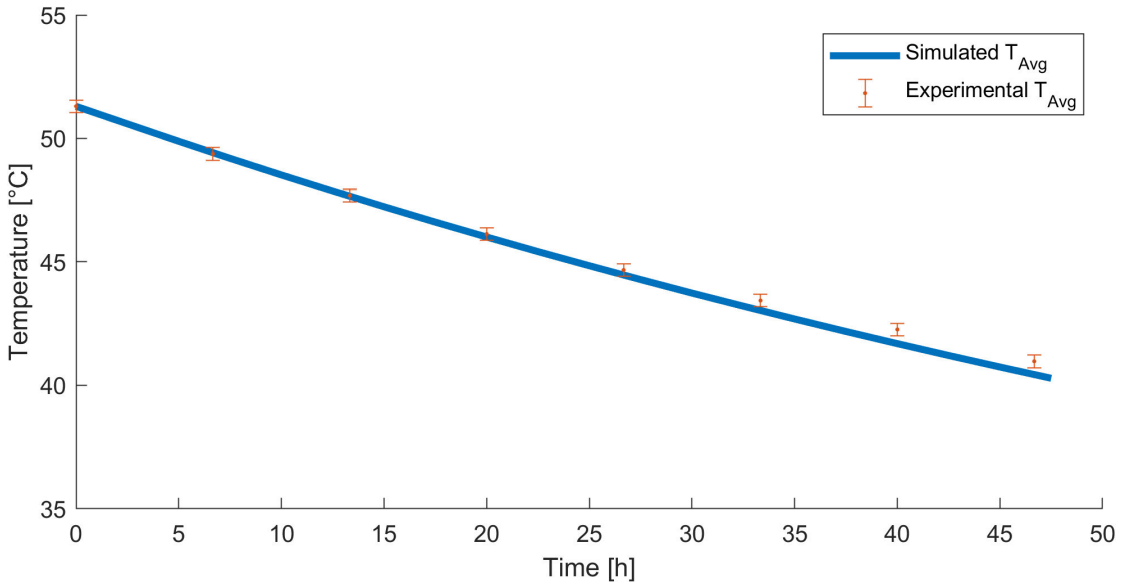


Figure 4-5: Simulated and Experimental DHW Average Tank Temperature

The validation trial of the DHW tank shows agreement within the measurement uncertainty for top and middle temperatures of the tank over the course of the 48-hour test. The bottom node of the tank shows a very similar trend, but not perfect agreement with the experiment. It is noted that between 0 and 33 hours of simulation, the bottom tank node temperatures are within $1°C$ of one another, approximately a 2% error. At end of the trial at 48 hours, the difference is approximately 1.5°C, which is approximately 5% error. The average tank temperatures agree within the experimental uncertainty until 33 hours, after which the average tank temperature error reaches a maximum of 0.7°C at 48 hours.

It should be noted that the DHW tank model is one of the limiting factors in terms of simulation fidelity to long-term experiments. This has been captured in previous reports [2, 28]. As mentioned in the literature review section, the tank model is known to over-estimate tank mixing. Fortunately, tank charging and discharging cycles in the current work are largely diurnal with a small emphasis in later chapters on storage of energy to the next day but seldom over a period of greater than 48 hours. Therefore, this work continues with the assertion that the tank model performs adequately for the purposes required.

4.7.2 Time Step Independence

A study was conducted to assess the sensitivity of the overall system TRNSYS model to the size of the simulation timestep. Possible time-steps range from sub 1 second to a maximum of 300 seconds.

The lower-bound of possible time step size is set practically by computation time and storage capacity. Also, measurements made using the ETU are logged on a time-base of 1 second, so it was deemed unnecessary to explore significantly smaller time step sizes. The upper bound of possible time step size is set by the frequency of draw data. One main source of draw data used in later chapters is Edwards [48] where draw data are presented on a 5-minute interval. This limits the time step to 300 seconds unless additional pre-processing is performed to accumulate and average the draw data, which was judged to be counterproductive as it would lead to a loss in fidelity.

The following are plots of the percent change in key simulation output variables with increasing time step size. Both the energy collection and energy delivery side of the system are considered.

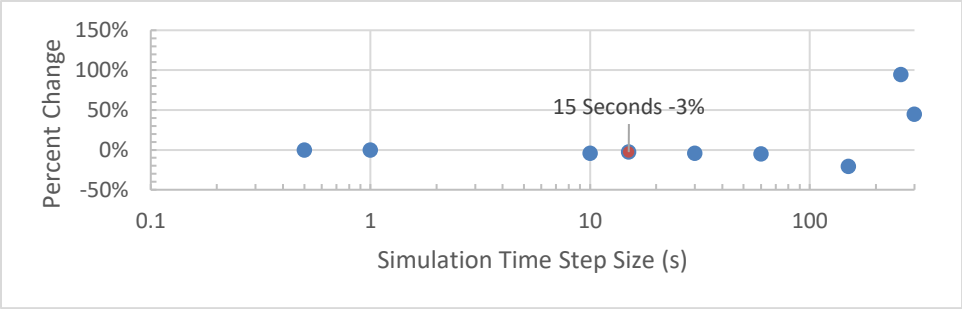


Figure 4-6: Change in DHW Tank Solar Charge Energy (HX+HP) vs. Time Step Size

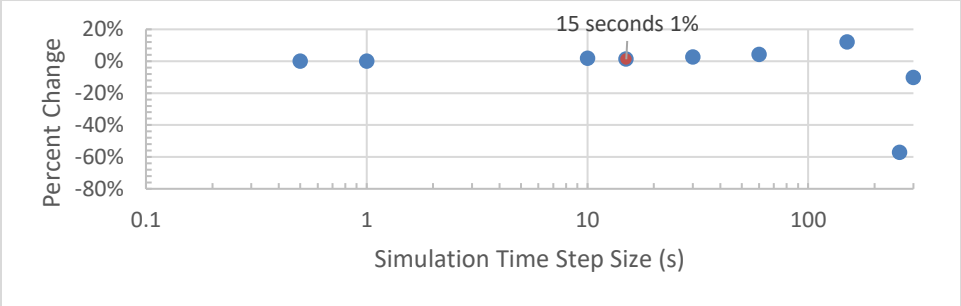


Figure 4-7: Change in DHW Tank Auxiliary Charge Energy vs. Time Step Size

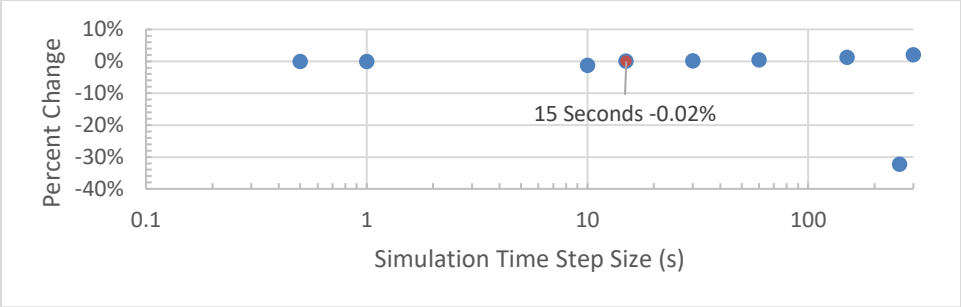


Figure 4-8: Change in Draw Energy Delivered vs. Time Step Size

In the three previous figures, a trend towards convergence is seen as the time step size reduces. Small amounts of numerical error, less than $\pm 5\%$ in terms of energy quantities are seen on the collection side of the system for time steps of 60 seconds and smaller. The variation between 0.5 seconds and 60 seconds is attributable to the discretization of incident radiation on the collector panel. For larger time steps, high and low peaks in the incident radiation are averaged and combined.

Similarly on the storage side of the model, tank nodal temperatures and the amount of thermal diffusion within the tank are dependent on time-step size. The TRNSYS tank models report and store the average nodal temperature for each time step. This is a known limitation of the TRNSYS tank models, and in situations where a high degree of fidelity in the tank's thermal profile is desired, it is not advisable to have particularly large time steps. Since STC, HX, and HP operation are all dependent on inlet temperatures from the DHW tank, care was taken to keep the time step size reasonably small to limit over-predicting of thermal diffusion within the tank.

Larger variation occurs for timesteps sized 150 seconds and greater due to similar effects, and also owing to a limitation of TRNSYS' modeling of fluid flow through model components having a capacitance. Pipes and storage tank nodes in TRNSYS have a limitation relating the capacitance of the node to the flow rate passing through it. TRNSYS is not equipped to handle flows that exceed the mass contained within a component or node in any given time-step. Time steps larger than 60 seconds exceed this limit and therefore simulations conducted with these large timesteps have so-called "run-away solutions" that cannot be trusted.

Another contribution to the variation of simulation results with the size of the timestep is the functioning of a system controller. The method of simulation in TRNSYS by linking multiple component models together through and iterative kernel requires that a controller have a defined state for the entirety of a time-step. There is no provision for part-loading or scaling the control function to represent a percentage of a timestep. Therefore, with larger time step sizes, the controller may be stuck in a certain mode for a period of time (within the same timestep) beyond the logical condition that it has been programmed to trigger a mode-change. This effect would be amplified for very large time steps (greater than 5 minutes), however those step sizes are already avoided due to the input data frequency discussed earlier.

A final consideration in the study of time step size is the computation time required for the simulation. The following plot shows the estimated simulation runtime for an annual simulation at

various time step sizes. Note that the axes in Figure 4-9 are log-log. There is a marked penalty in terms of simulation runtime for timesteps smaller than 10 seconds.

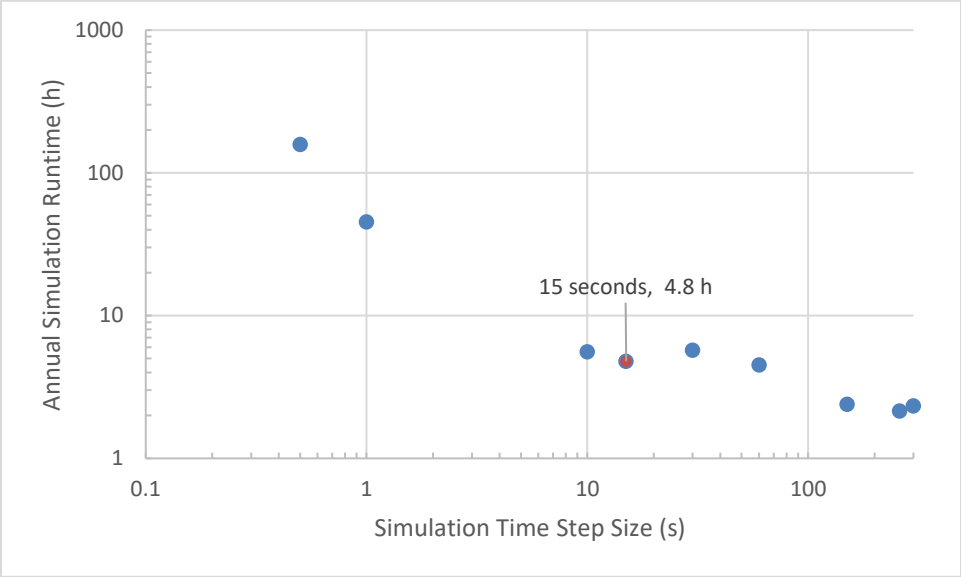


Figure 4-9 Annual Simulation Computation Time vs Time Step Size

A timestep of 15 seconds was selected for the overall system simulation because it strikes an appropriate balance between maintaining numerical precision and expending computational resources and time. The plots in this section show that a 15 second time step leads to less than 3% contribution of numerical error in collected solar energy compared to the smallest timestep tested. Using 15 second time steps, an annual simulation can be performed in approximately 4.8 hours real time.

4.7.3 System Model Validation

A series of simulations were conducted with matching validation experiments to establish confidence in the overall TRNSYS model. With matching inputs between the ETU tests and the TRNSYS simulation, a high degree of agreement was observed, however there are some discrepancies between the simulated and measured results that will be discussed below. The results presented below are from a representative day based on weather inputs for July 3rd in the Ottawa TMY file and draws from a CSA standard [47]. Error quantities reported below are based on the mean absolute error and the mean percent error calculated using Equations 3-11 and 3-12 as seen in the HP characterization Section 3.9. Figure 4-10 and Figure 4-11 show model inputs for the following validation trial.

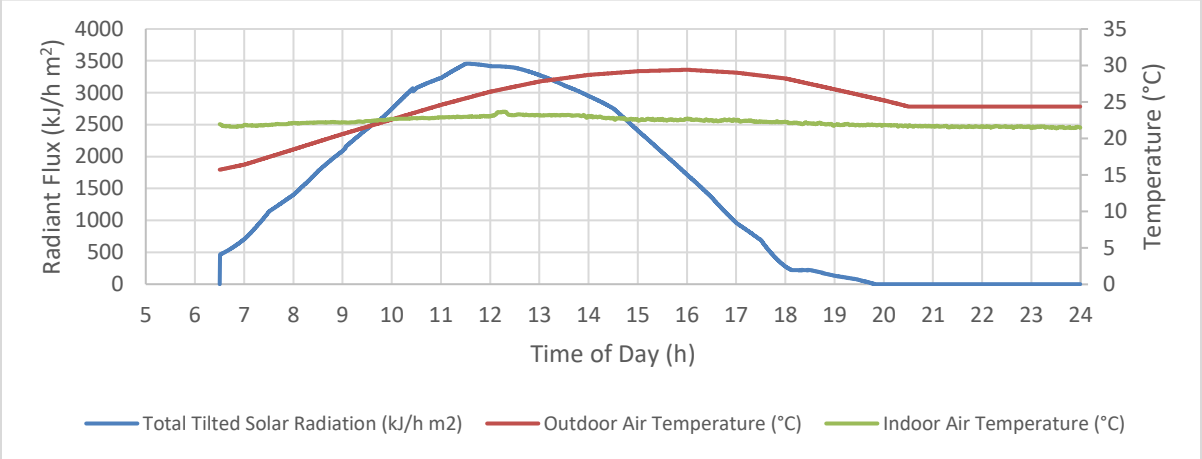


Figure 4-10: Solar radiation and ambient temperature inputs to the model for July 3rd validation trial

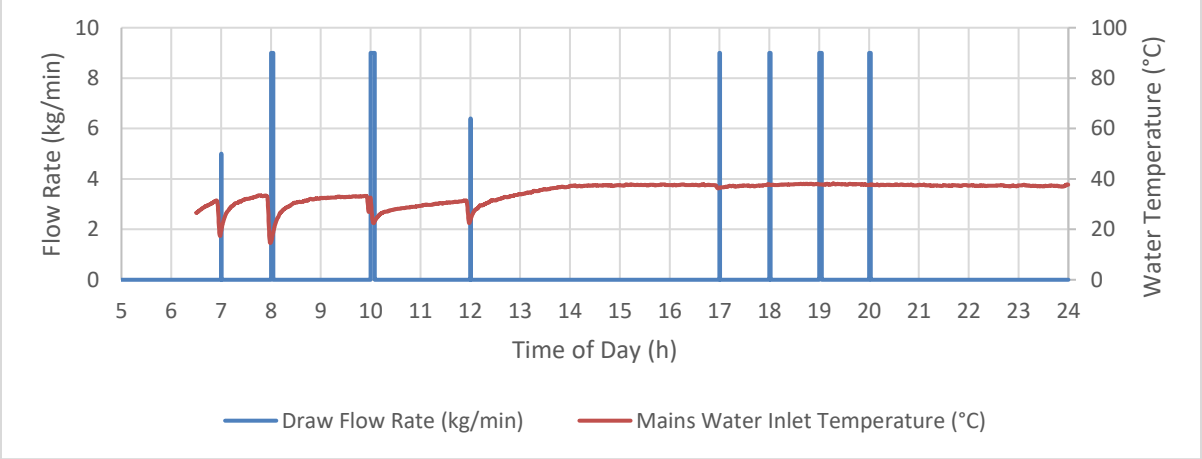


Figure 4-11: Draw flow rate and mains inlet temperature inputs to model for July 3rd validation trial

The TRNSYS model and ETU were configured to use the same inputs for the validation trials presented in this chapter. Figure 4-10 shows the total radiant flux incident on a tilted collector along with the ambient outdoor air temperature extracted from the Ottawa TMY file. Also plotted is the indoor ambient temperature that was fed to the TRNSYS model as the temperature of the surroundings for components other than the STC. In terms of insolation, July 3rd can be described as a relatively clear-sky day. Since it is July, there is no reason to believe that there would be snow-cover affecting ground reflectance or shading the STC.

For the DHW draw forcing function, a draw profile adapted from the CSA standard for packaged solar water heaters [47] was employed. The draw schedule is plotted in Figure 4-11 along with the

cold mains inlet temperature that was used as an input to the TRNSYS model. While the TMY file provides an estimated mains water temperature, it was found in preliminary work that the temperature of the cold-water supply available in STRL varies throughout the day and is sensitive to draws. During periods when there is no draw occurring, the mains inlet thermocouple reads temperatures closer to the storage temperature of water in the DHW tank, suggesting some conduction of heat out of the tank and into the supply plumbing. During periods of draws, the inlet temperature quickly drops to around or below room temperature. For the model validation, the measured mains inlet temperature from the experimental trial was used as the input to the TRNSYS simulation.

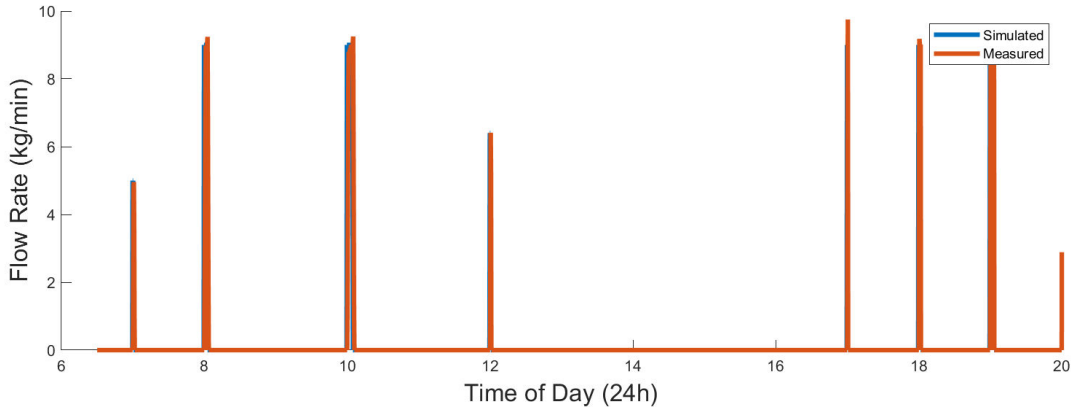


Figure 4-12: Draw flow rates in July 3rd representative day trial

Figure 4-12 above compares the simulated (TRNSYS Model) and measured (ETU Trial) draw throughout the July 3rd representative day trial. It can be seen that the simulation tracks well with the measured draws, which is expected because both results are forced from the same input data. The mean error between simulated and measured results for draw flow rates is 0.65 kg/min, or 1.8% error.

Small differences in both magnitude and the timing of the draws can be seen in Figure 4-12. The timing error can be attributed in part to experimental sampling by the DAQ being slightly out of phase with the simulation timestep, also the method of controlling draw flows on the ETU is with an electromechanically actuated valve which has some inherent time-delay in opening. These small differences in timing are less than 1 minute, and have little impact on the overall simulation spanning several hours.

Figure 4-13 shows the flow rates produced by the circulator pumps in the source-side and load-side plumbing loops. The flow rates from 7h to 12h represent flow during HX operation and are in near-perfect agreement. From 12h to 14h, there is also near-perfect agreement for the main portion of the

day under HP operation. At 14h, the simulation flow rate goes to zero 14.8 minutes earlier than when the experimental measurements go to zero. It can also be seen that there are three short periods of flow at 15.3h, 15.75h, and 16.5h. These discrepancies will be discussed alongside the plots of HP operation later. The mean error in system flow rates over the entire day was 0.422 kg/min or approximately 8%. For the period from 7h to 14h, the error was only 0.047 kg/min or 1%.

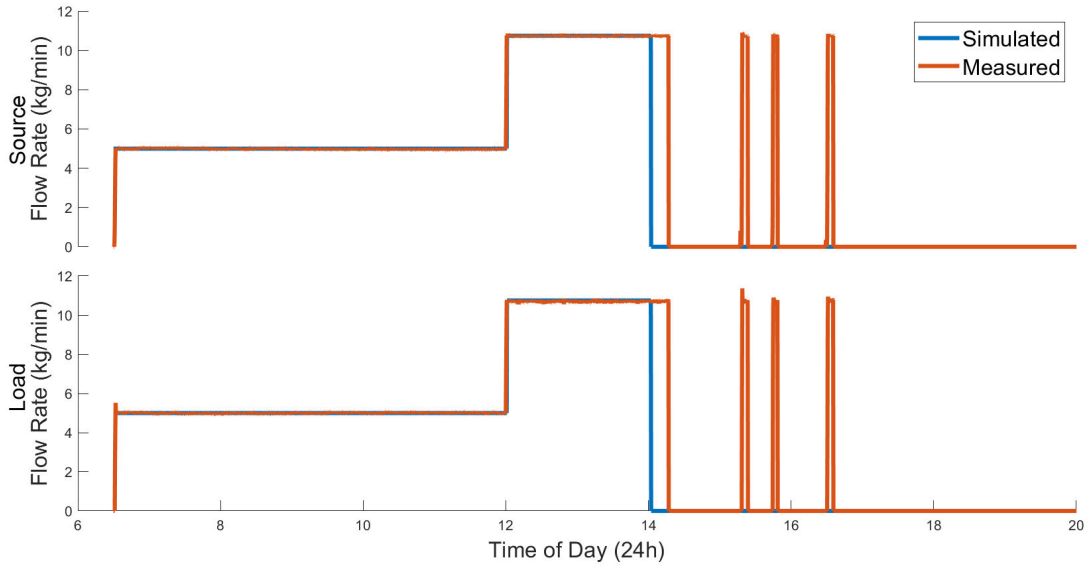


Figure 4-13: Flow rates within the Source and Load side in July 3rd representative day trial

Figure 4-14 below shows the operation of the HX from 7h to 12h in the representative day trial. Overall, the simulated and measured performance of the HX are in agreement, showing similar rates of energy transfer throughout the day. The simulation appears to respond slightly faster to transients in the temperature inputs. This is likely owing to the damping effect of the mass of water contained in the real HX, which is not modeled in TRNSYS. Thermal capacitance in TRNSYS is modeled in the pipe elements on either side of the HX, which means that temperature variations at the inlet and outlet are smoothed out in real measured data but aren't smoothed in the simulation.

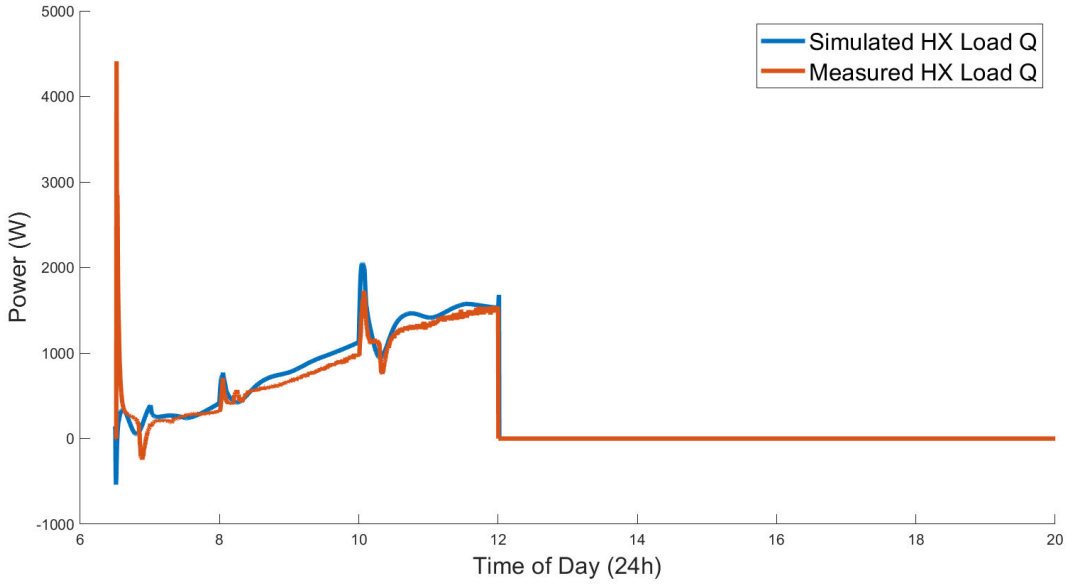


Figure 4-14: HX Energy transfer rate in July 3rd representative day trial

There is, however, a rather large discrepancy between simulated and experimental results during the first 10 minutes of the trial. This is attributed to non-matching starting conditions between the ETU and the simulation. While care was taken to match the simulation initial conditions as closely to the experimental starting conditions as possible, some mis-match was unavoidable. In the verification trials, to obtain the DHW tank starting condition described in Section 4.6, the tank was pre-charged using the ETU's immersion heater and the HX. This left a plug of high temperature water within the source-side loop that was present at the beginning of the verification trials. While the simulation started with an expected source-side temperature that would have settled to the outdoor ambient condition overnight with the sun down at 12.7°C, the experiment started with a pre-heated source temperature of approximately 32°C. After the first 10 minutes of the trial, the temperature discrepancy has largely worked its way out through an elevated rate of heat transfer in the HX and operation begins to stabilize. Including the spurious transient at the beginning of the trial, mean error in HX operation over the day is 54.6 W. Analyzing the period after 10 minutes of simulation, the mean error is 41.0 W or 5.2%.

Figure 4-15 shows the operation of the HP in terms of input power and energy transferred from the source (evaporator) side to the load (condenser) side. There is near perfect agreement for input power but some discrepancies in the start-up transient seen in the heat transfer rate. In this case, the startup

transient of the simulation over-predicts the measured performance unlike with the HX discussed previously. This disagreement is attributed once again to a mis-match in starting conditions, and the way that the change-over between HX and HP are simulated in TRNSYS.

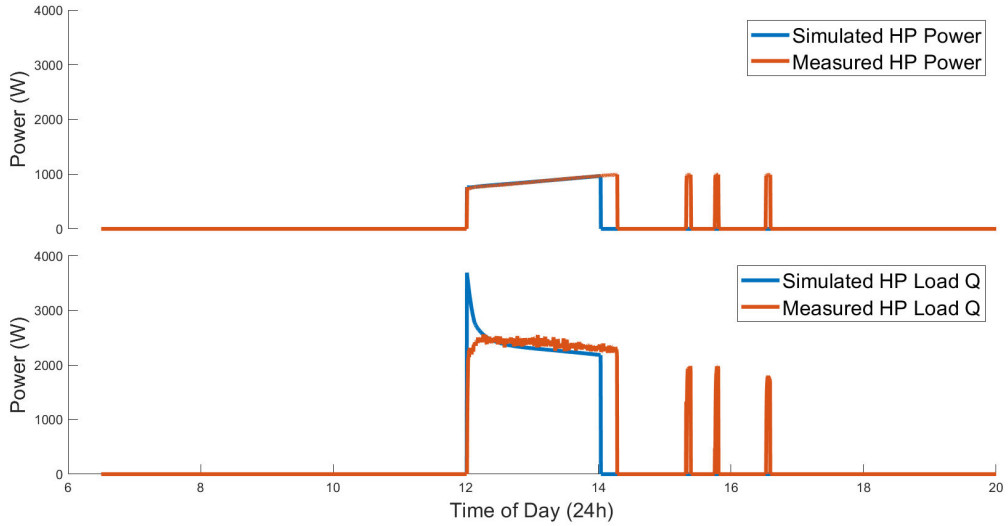


Figure 4-15: HP input power and energy transfer rate in July 3rd representative day trial

The measured HP source side inlet temperature at the time of the changeover is 32.6°C, which is discontinuous from the measured HX inlet temperature at the moment before changeover at 41.2°C. On the other hand, the simulated HP inlet source temperature at the time of changeover is 40.2°C, which matches exactly with the temperature that TRNSYS had been sending to the HX the moment before changeover. Evidently, there is a plug of lower temperature water held within the HP evaporator and its supply tubing that starts at an uncontrolled temperature that is not represented within TRNSYS. The result of this temperature mis-match is that the HP in the simulation operates at an artificially higher COP for approximately 7.5 minutes until the HP source inlet temperatures between the simulation and the experiment converge.

Another disagreement mentioned during the discussion of the circulator flow rates results from three short activations of the HP at 15.3h, 15.75h, and 16.5h in the experimental results. These activations do not occur in the simulation. The reason for this difference is owing to the HP's safety lockout system. In reality, and in the experimental setup, the HP protects against freezing the source loop by monitoring the source-side outlet temperature and forces the compressor off when temperatures approach 0°C. In the simulation, while developing the TRNSYS TYPE 1162 HP model, it was found that basing the HP safety lockout on the model's own output led to a circular-reference that caused

the HP to oscillate on and off. This was solved by instead setting a conservative threshold on the HP's operation that holds the compressor off whenever source side temperatures are predicted to fall below 5°C. It is therefore a limitation of this system's TRNSYS model, that the real HP can be operated in conditions slightly beyond the limits in the simulation. It should be noted, however, that the short periods of HP activation in this trial occur late in the day when there is little insolation. In a traditional SDHW system, heat at this hour would be provided by the auxiliary elements, not the STC. In future chapters, this issue will be mitigated by control systems that are aware of when daylight hours are, and/or the level of insolation.

One final discrepancy that must be noted is the difference in end-time of the main HP operation between 12 h and around 14 h. The simulation charges the tank from 12 h to 14 h, reaching 50°C at the top of the tank and then deactivating the HP. In the experiment, the HP is also activated at 12 h, but it continues to charge the tank until 14.29 h at which point the DHW tank reaches its 50°C setpoint. An explanation for the delay in deactivation of the HP in the experimental results is due to resolution of temperature sensing within the DHW tank. In the simulation, the top node temperature is represented by the average temperature of the top 7.6 cm of the tank (top node of 20 nodes). In the experimental setup, the tank top temperature is measured at a point approximately 26.7 cm from the top of the tank (top node of 5 nodes). It takes a certain amount more time to see the higher temperature register on the thermocouple located lower in the tank than the simulated node located right at the top. The average absolute difference over the entire test day including the mis-matched end-time and the three late-day activations for HP input power is 34.5 W or 7.2% and for the HP load Q was 92.9 W or 7.8%. Limiting the analysis to the period between 7h and 14h, the error in HP power was 1.67 W or 0.21% and for HP load Q the mean error was 46.4 W or 1.9%.

The plots in Figure 4-16 show the temperature profiles of the simulated and measured DHW tanks during the July 3 representative day trial. Overall, the plots show a strong agreement in system behavior between the simulation and the experiment. As noted earlier in the discussion of HP operation, there is a certain amount of error in tank temperatures attributable to the location and method of temperature measurement for the experiment and temperature representation for the simulation. Temperatures in the simulation represent both a spatially averaged temperature over the height of the tank node, and a time averaged temperature over the timespan of the simulation time step. Experimental temperatures are point measurements both in time and location.

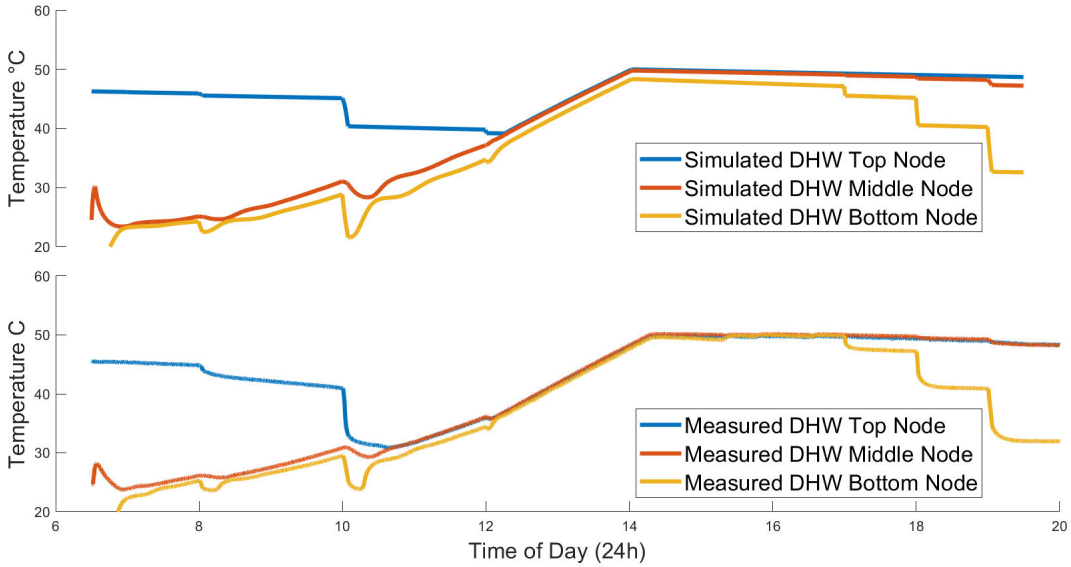


Figure 4-16: DHW tank temperature profiles for July 3rd representative day trial

As seen in Figure 4-17, the thermocouple measurements used to report experimental results do not lie in exactly the same location as node-centers in the simulation. The expected result of this would be a slight over-prediction of top-node temperatures and a slight under-prediction of bottom node temperatures by the simulation. This result is observed in the plotted temperature profiles in Figure 4-16.

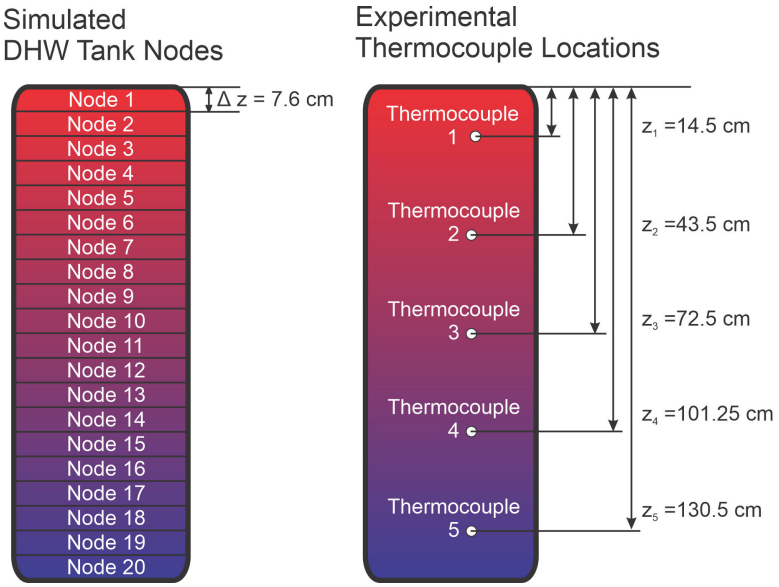


Figure 4-17: Simulation and experimental temperature locations

Another trend worth noting for its agreement between the simulated and measured results plotted in Figure 4-16 is the amount of tank stratification throughout the trial. In the early hours, the high degree of separation between the top and bottom temperatures of the DHW tank indicates a highly stratified starting condition. Towards the end of the HX operation at 12 h, though the bottom two nodes of the tank have gained in temperature, the top node of the tank is seen in both the simulated and measured results to have decreased in temperature, representing a loss in stratification of the tank. This trend continues after 12 h to 14 h as the HP operation utilizes a higher flow rate of water, resulting in a mixed tank. This is seen in the temperature profile as the three node temperatures converge towards one another from around 12.5 to 14h. Towards the end of the day, the experimental temperatures remain more homogenous than the simulated temperature profile because the HP operates longer and later into the day as mentioned previously.

Figure 4-18 shows the average temperature of the DHW tank throughout the July 3rd trial. The average temperature of the tank is a simulation output that trends with the total energy of the tank. Despite some discrepancies in the temperature profile discussed earlier, it can be seen that the average tank temperature output by the simulation has a maximum error of less than 2°C compared with the analogous experimental result. The mean absolute error for average tank temperature is 0.97°C, or 2.4%.

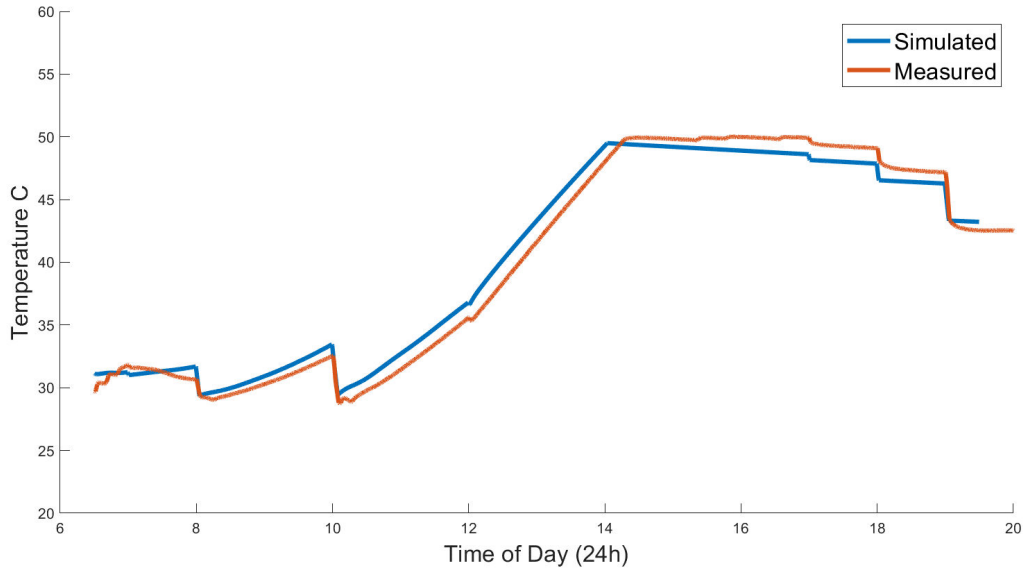


Figure 4-18: Average DHW tank temperature for July 3rd representative day trial

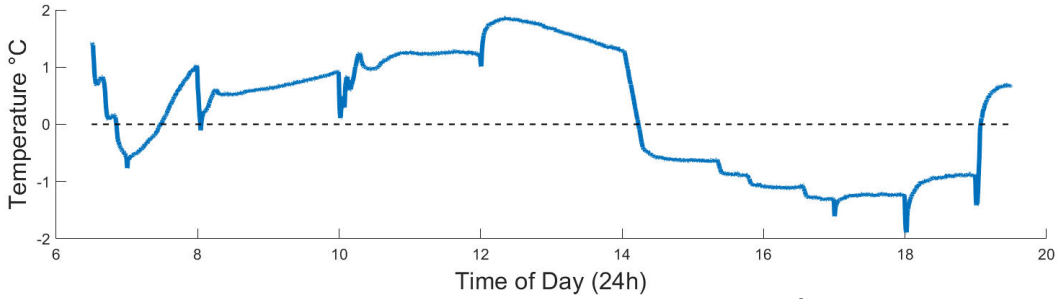


Figure 4-19: Average DHW tank temperature residuals for July 3rd representative day trial

The average tank temperature residuals are plotted in Figure 4-19, where it can be seen that the tank temperature is slightly over-predicted during the charging phase between 7 and 14 h, and then slightly under-predicted in the later afternoon/evening. It is likely that the over-prediction during charging indicates that there are some thermal losses to the surroundings in the experimental setup that exceed what is modeled in TRNSYS. It bears mentioning as well, that the entire trial begins with a residual of approximately 1.5°C, which is an accumulated error over the starting temperature profile of the DHW tank. The tank is highly stratified at the beginning of the test and with only 5 experimental temperature measurements, the starting conditions in the TRNSYS model could easily be ±0.5°C from their true values at each measurement point. Considering what has been discussed in this section, the agreement between the simulation and measurement is satisfactory.

In this chapter, the overall system TRNSYS model was presented and steps taken to validate this model have been discussed. This model is the basis for further investigations that appear in Chapters 5 and 6.

Chapter 5
Baseline System Performance Simulations

This chapter details a simulation-based study into the operation of SAHP systems with components arranged as described in earlier chapters. Of interest in this study was the effect of various control options on system performance, as well as establishing baseline performance of this system and simpler single mode SDHW and SAHP systems under conventional temperature-based control schemes. Text and figures in this chapter are reproduced with permission from the author's original works, "3-Stage Control of A Variable-Refrigerant-Flow Heat Pump For Solar Domestic Hot Water Applications," 2022 [49] "Annual performance of a simulated multi-mode SDHW/SAHP system with various control options," 2023 [37]. A letter of permission is included at the end of this document.

5.1 Method

Simulations were conducted using the TRNSYS model described in Chapter 4. Preliminary daily simulations were conducted to assess the benefit of employing the HP's modulating capability against systems having either no HP, or a fixed-capacity HP equivalent to the variable HP operating at 100%.

Annual simulations were conducted to compare the operation of a multi-mode SDHW/SAHP system under 3 stage modulating control against the same system's operation under on/off control. As reference cases, single mode SDHW and SAHP systems were also simulated, as well as an electric resistance DHW system. A custom MATLAB controller was written to implement the various configurations that were of interest in this study. The operation of the controller is described in Section 5.2.

5.2 MATLAB Controller

As explained in Section 4.4, TRNSYS is able to operate using control signals from an interchangeable controller written in MATLAB. The principle of the controller used in this chapters' simulations is to prioritize the operation of the least expensive heat source first in terms of purchased energy, the HX (Mode 1), followed by the next least expensive source, the HP (Mode 2), and lastly the controller implements a "Warmup mode" (Mode 3) when neither the HX or HP can operate. The HX and HP are restricted to operating within daylight hours only and the AUX heating element is restricted to operating outside of daylight hours, such that charging capacity in the storage tank is not prematurely

taken up by electrical resistance heating when a more economical source of heat could be used in the daytime.

The operating principle of the controller is shown in the flowchart in Figure 5-1. Table 5-1 explains the logical tests and simulation variables set by the various blocks in the flow chart.

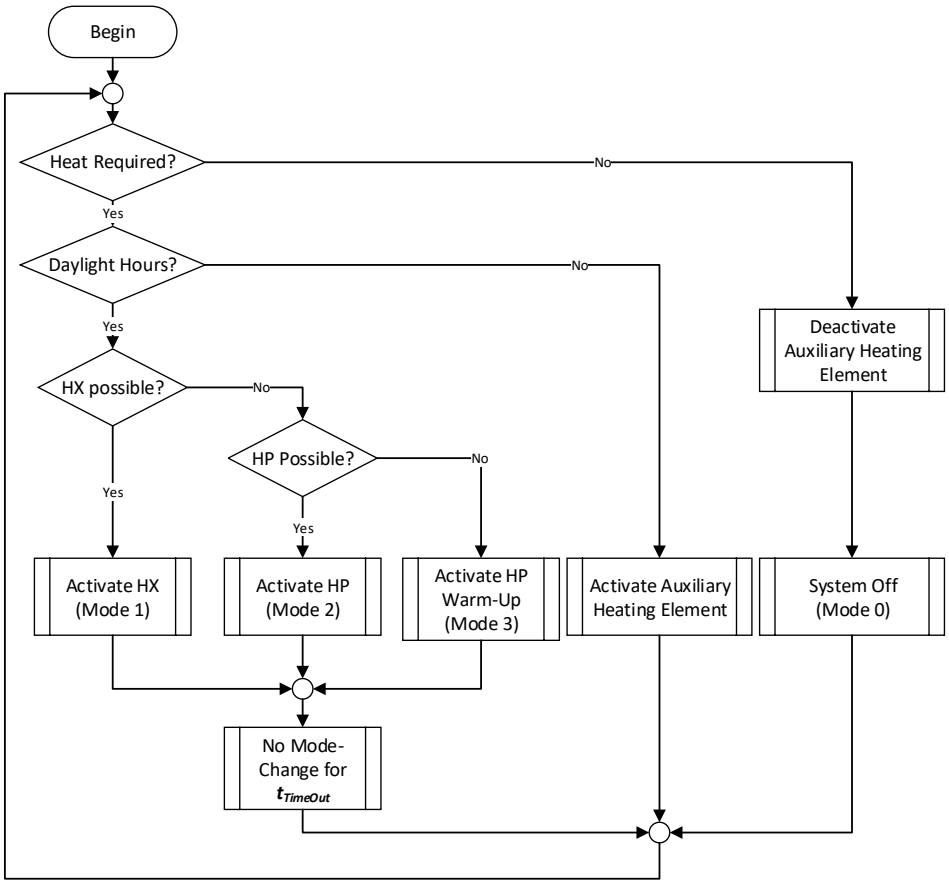


Figure 5-1: Multi-mode SDHW/SAHP Controller Operating Principle

Table 5-1: Explanation of flow chart blocks and logic

Flowchart Block	Test or Subroutine
Heat Required?	$T_{DHW} < T_{set} + T_{Deadband}$ Where T_{DHW} is the temperature of the DHW storage tank at the height of the thermostat, $T_{set} = 50°C$, and $T_{deadband} = 2°C$
Daylight Hours?	$t > SunriseTime$ & $t < SunsetTime$ Were *t* is the simulation time in h, and $SunriseTime$ and $SunsetTime$ are calculated dawn and dusk hours of each day.
HX Possible?	$T_{o,col}$-$T_{DHW,Bot} \geq 2°C$ Where $T_{o,col}$ is the collector outlet temperature, and $T_{DHW,Bot}$ is the temperature at the bottom of the DHW storage tank.
HP Possible?	$T_{o,col} \geq T_{HP,cutin}$ Where $T_{HP,cutin}$ is set to $10°C$ to prevent icing in the evaporator.
Activate HX	Mode = 1 Activate circulation pumps and set diverters towards the HX.
Activate HP	Mode = 2 Activate circulation pumps and set diverters towards the HP. Activate HP Compressor.
Activate HP Warmup	Mode = 3 Activate circulation pumps and set diverters towards the HP. Deactivate HP Compressor.
Activate/Deactivate AUX Heating Element	Toggle Aux Element Heating rate: Off=0 kJ/hr, On= 10735 kJ/h
System Off	Mode = 0 Everything off.

The MATLAB control script has two additional features beyond the control logic explained in Table 5-1. First the script was written so that the available operating modes can be restricted. In doing so the same TRNSYS model can be used to simulate an SDHW system (HX Mode 1 Only), and SAHP system (HP Mode 2 Only), a multi-mode SDHW/SAHP System (all modes available), also an electric resistance DHW system (disable modes 1, 2, and 3) by only allowing the AUX element. This controller function is implemented through a control parameter *AvailableModes* which translates as follows: 0 → All Modes available, 1 → HX Only, 2 → HP Only, 3 → AUX Only.

The other additional function of the control script was to control modulation of the HP compressor. A 3-stage modulation can be used to affect the speed of the HP compressor and therefore vary the heat transfer capacity of the HP. This controller function is implemented by a control parameter *ModulateCompressor* which translates as follows: 0 → Compressor runs full speed whenever HP is called, 1→ Compressor runs using 3-stages of capacity.

When HP modulation is enabled, the compressor speed is set based on the temperature of the source-side inlet fluid (T_{si}). The HP runs at 62.5% (2250 RPM) compressor speed when T_{si} <10°C, it runs at 75% speed (2700 RPM) when $10°C < T_{si} \leq 20°C$, and the compressor was run at 100% speed (3600 RPM) when $T_{si} > 20°C$.

The minimum rotor speed of 62.5% (2250 RPM) was selected to prevent stalling of the compressor, which could lead to damage in a real HP. The temperature control point of $10°C$ was selected to prevent the heat transfer fluid (water in this simulation) from approaching the freezing point. Real HPs will often have a safety limit that powers off the compressor if the heat transfer fluid is in danger of freezing.

The combination of these controller functions results in the following 6 configurations:

1) All Modes, Non Modulating
2) All Modes, Modulating
3) HP Only, Non Modulating
4) HP Only, Modulating
5) HX Only
6) AUX Only

5.3 Results and Analysis

A single day's operation of configurations 3, 4, and 5 under the controller described in Section 5.2 is presented in 5.3.1 followed by annual simulations that were conducted for each of the 6 operating configurations. The performance of each system's annual simulation is reported in terms of Solar Fraction (defined in Section 5.3.3), Quantity of Solar Energy collected, and the Quantity of Purchased Energy Consumed by each simulated system. In addition, the adequacy of the system is tested by keeping a count of the number of draw events and the associated mass of water that experienced a temperature below $35°C$ as will be explained in Section 5.3.6.

5.3.1 Preliminary Daily Simulation

A single day trial was conducted to demonstrate the effect and operation of systems operating in configurations 3) HP Only, Non Modulating, 4) HP Only, Modulating, and 5) HX only. Initial conditions for the DHW tank in these tests were the same for all 3 systems, but instead of the initial DHW temperature presented in Section 4.6, these simulations began with a mixed DHW tank at 7°C. There were no draws in these preliminary simulations. The input weather and solar conditions for these three daily simulations are plotted below in Figure 5-2.

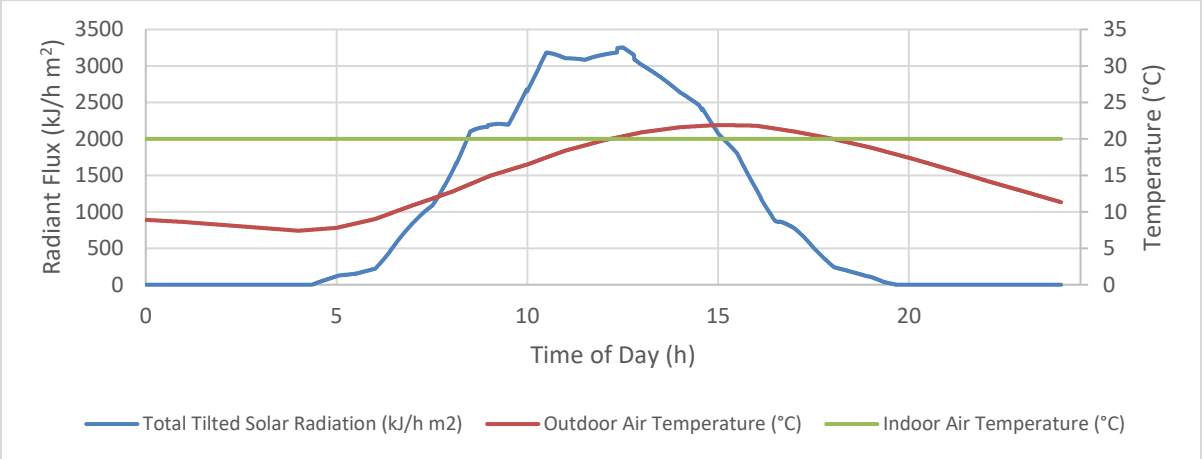


Figure 5-2: Solar radiation and ambient temperature inputs to the model for June 5th preliminary simulations

Figure 5-3 shows the energy gained in one-day's operation of the system as calculated by integrating the energy storage rate reported by the "TYPE 534" tank model which represents the DHW tank.

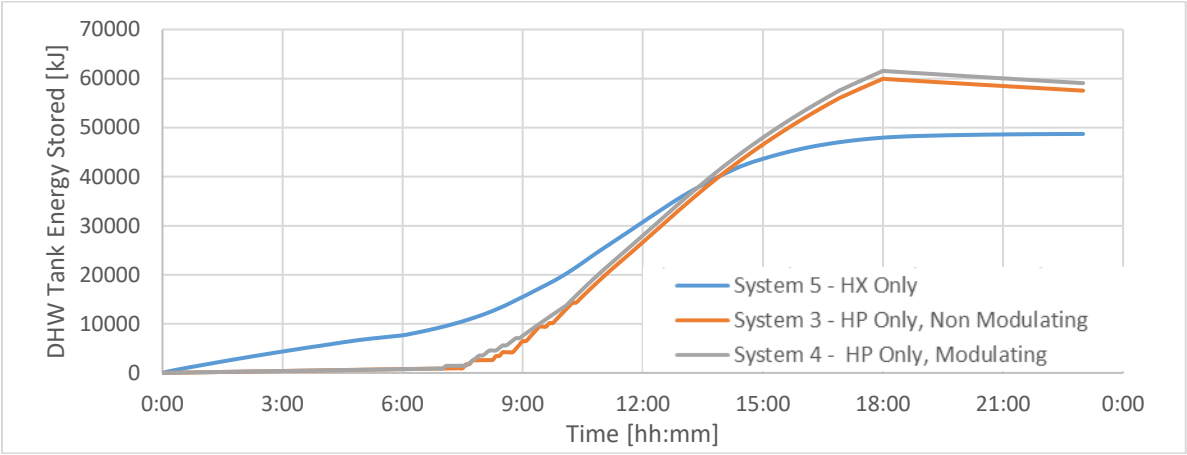


Figure 5-3: Energy Stored in DHW Tank vs. Time for three comparable SDHW and SAHP Systems

Separate periods of time can be highlighted in the above figure to understand and compare the performance of the systems. From 0 h to approximately 7 h, the HX only system is powered on and although there is no solar radiation incident on the collector surface during these hours, the system is collecting some energy through convective heat transfer with relatively warm ambient air. At this point in the simulation, the DHW tank is filled with water that is colder than the surrounding atmosphere.

From 7 h to 12 h all three systems are active but the HP systems are not yet receiving enough solar radiation to continuously run the HP without short cycling. The short-cycling of the HP will be discussed on the following page. During this period, it can be seen that the slope of the traces representing energy stored increases as time progresses. This increase in the rate of energy collection coincides with the increasing intensity of solar irradiation (due to the sun rising). The slopes continue to increase until approximately 12 h.

After 12 h and until approximately 19 h, all three systems remain active, but the rate of energy collection decreases as expected while the amount of solar irradiation subsides throughout the day. Of note, however, is that between 13.5 h and 14 h the energy storage rates for the HP systems intersect with that of the HX-only, indicating that for a runtime longer than 14 hours, the SAHP systems would collect more energy than the HX-based SDHW system.

Although the HX-only system is more effective during the early hours of the morning, its energy collection is outpaced by the two systems with HPs from the time that they are activated to the time that they are turned off. The higher rate of heat storage is attributed both to the higher collection efficiency

attained at the STC as a result of cooling the collector's heat transfer fluid, as well as the heat addition to the system through compressor work at the HP.

It can also be observed in Figure 5-3, that the HP with modulating compressor outperforms its fixed-capacity counterpart throughout the day and ultimately collects slightly more energy. It is worth highlighting here though, that the most important benefit of the speed-modulated HP might not be in terms of increased energy collection, but instead in reduced compressor wear and increased HP lifespan by avoiding short-cycle operation during periods of low solar irradiation.

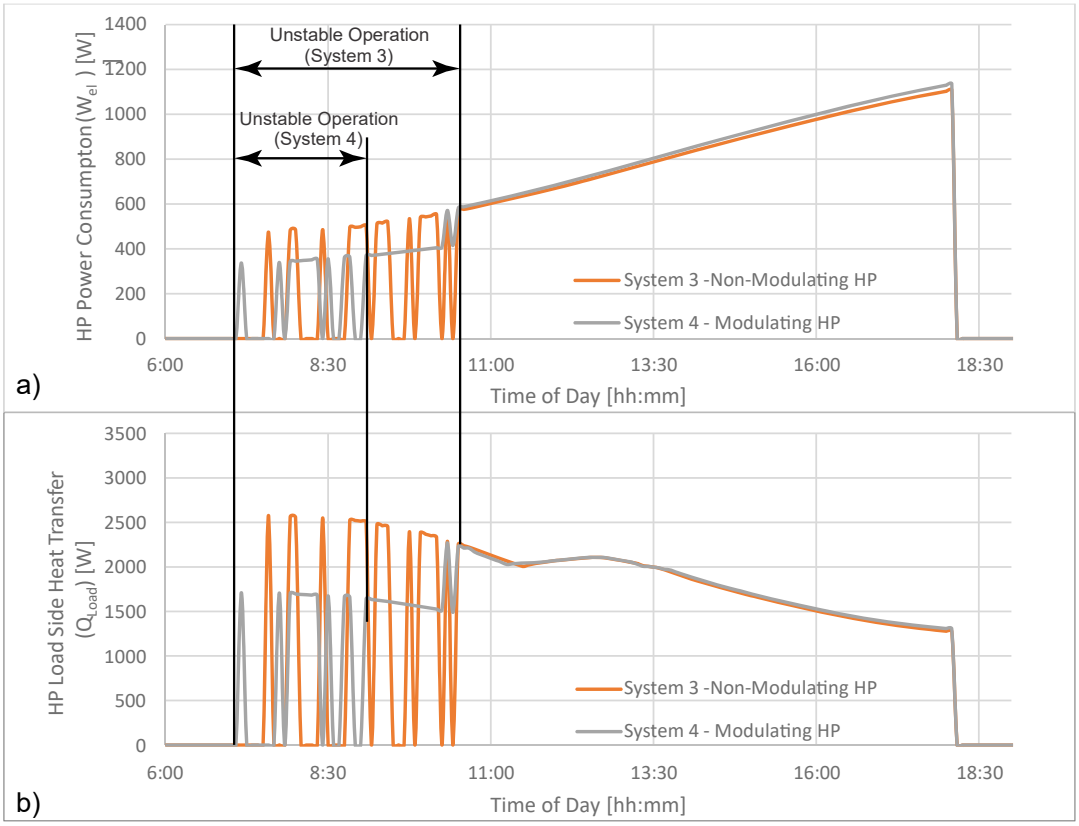


Figure 5-4: Plotted HP Performance data for SAHP systems with and without modulation

The plots in Figure 5-4 show the benefits of modulating the speed of the compressor to match the available energy to transfer. The features that are of greatest significance in the above plots are the oscillations at the beginning of the operation period of the HP systems. Between 7 h and 11 h, the HP can be seen to switch itself on and off. This unstable operation is not an artifact of the simulation, but occurs in real-world heat-pump operation. It is a result of the HP having a heat transfer capacity that exceeds the amount of solar energy that the STC can supply under that hour's weather conditions. The short-cycling

condition results from the HP spooling up, then removing too much heat from the source-side loop, before shutting down on a safety limit to prevent freezing the collector fluid and obstructing flow in the evaporator. Frequent short-cycles not-only reduce the amount of time that the HP can effectively heat water on the load-side, they can also lead to damage to the compressor or can otherwise shorten the service life of the HP. For this reason, HP manufacturers and researchers have implemented strategies to minimize short-cycling of the HP.

Previous mitigation strategies for the HP short-cycling during mornings and cloudy periods included the addition of a larger fluid capacity on the source-side of the system [3, 2], as well as control strategies that had extended periods of running the circulation pump on the source-side without calling for the HP to run [18]. It has also been attempted to simply avoid HP operation during periods of time where cycling on-and-off are likely to occur or mandate a rest-period for the equipment every time it switches off. All of these strategies limit the amount of time that a HP can run in any operating window and therefore reduce the amount of energy that could be stored by a system such as this.

Observing the relative duration of unstable operation of the systems in Figure 5-4, it can be seen that the modulating HP system is able to reduce unstable operation that would be expected by more than 33%. This means that the HP can be safely used over a wider range of solar conditions and over a longer period of collection each day.

5.3.2 Annual Simulation Model Inputs

Longer term simulations were also completed using the validated model described in Chapter 4. For these simulations, the initial conditions for the model were as described in Section 4.6 and model inputs were as described in Section 4.5.

For the draw profile, it was desired to simulate a realistic residential draw profile with non-repeating daily draws. Edwards et al. provide time-series data of the DHW flowrates measured in a sample of Canadian homes [48]. For this work, "House 38" was selected from their published data. This house is characterized as a 3-person household consuming on average 176 L of DHW per day. According to Edwards, this household falls near the mean consumption level of the houses in their study.

While there is little value in plotting the entire year's draws from Edwards [48], an example day (July 28th is show in Figure 5-5 on the following page to illustrate the variability of the draws included in the set. For concision, no plot of weather input conditions is provided for the annual simulations, but

the weather inputs are similar to those presented in the previous sub-section, with seasonal as well as daily variation of both temperature and irradiation input. Weather data was again taken from the TMY weather file for Ottawa, Ontario.

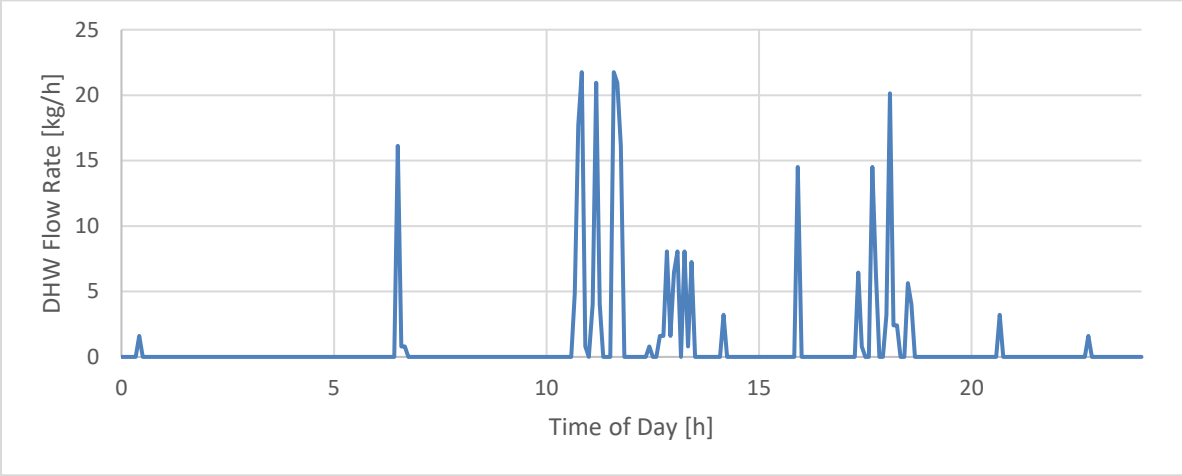


Figure 5-5: Sample 24 hours of DHW draw profile from House 38 on July 28th [48]

5.3.3 Solar Fraction

A common performance metric for traditional SDHW systems is the solar fraction which bears the symbol f_i on a monthly basis, and $\mathcal{F}$ when reported on an annual basis. Duffie and Beckman [16] define the solar fraction as the ratio of the solar contribution to the load, divided by the load. It is defined as:

$$f_i = \frac{L_i - L_{A,i}}{L_i} = \frac{L_{S,i}}{L_i} \qquad (5-1)$$

$$\mathcal{F} = \frac{L - L_A}{L} = \frac{L_S}{L} = \frac{\sum_{yr} f_i L_i}{\sum_{yr} L_i} \qquad (5-2)$$

In the above expressions *i* represents the month in question, L represents the total load energy delivered, L_A is the energy that would be consumed by the auxiliary heaters and L_S represents the solar energy collected by the system. It is assumed for the purpose of calculating solar fraction, that $L = L_A + L_S$, which is to say that any solar energy collected directly offsets the amount of auxiliary heat required to meet the load. This approach neglects losses and parasitic energy draws from ancillary equipment, which is a limitation of the metric for comparison across systems with varying components. Nevertheless, for studies such as the current one, where system components are similar, solar fraction provides a useful

metric for comparison between different operating modes or control strategies. The monthly solar fraction of each simulated system is plotted below in Figure 5-6, while annual values of the solar fraction are shown in Table 5-2.

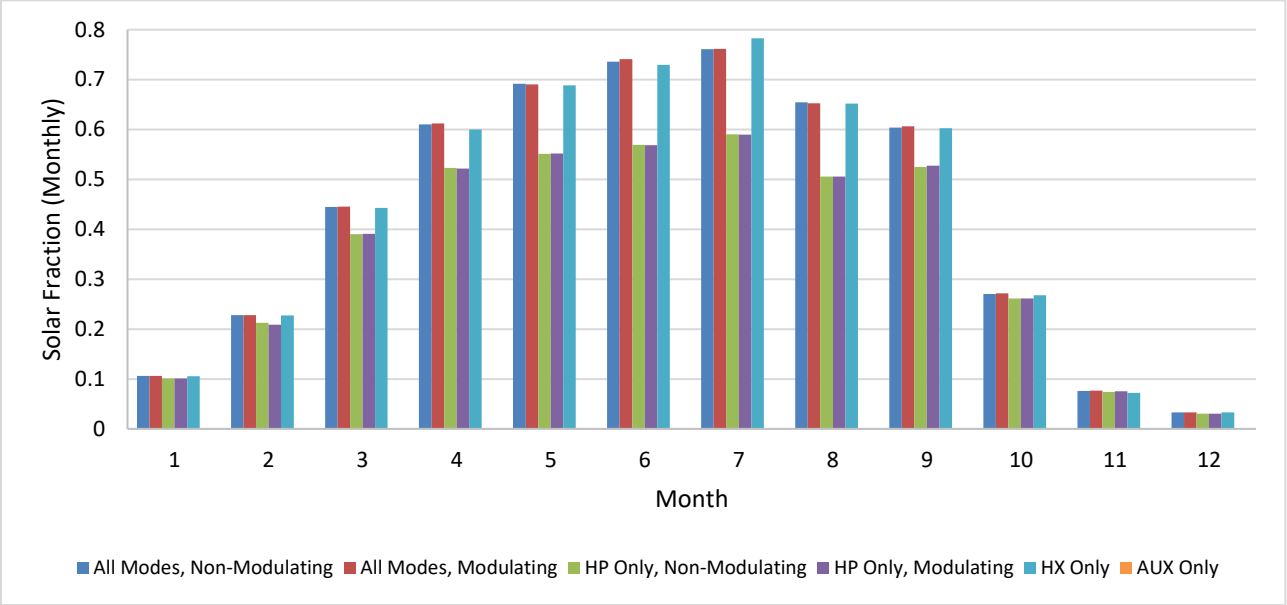


Figure 5-6: Monthly Solar Fraction of simulated SDHW/SAHP Systems

Table 5-2: Annual Solar Fraction of simulated SDHW/SAHP Systems

	All Modes		HP Only		HX Only	AUX Only
	Non-Modulating	Modulating	Non-Modulating	Modulating		
Annual Solar Fraction	0.4368	0.4374	0.3640	0.3638	0.4340	0

On the basis of solar fraction, the highest performing system was the "All Modes" system, which could switch between HP and HX operation with no restriction. Both the modulating and non-modulating versions of this system had annual solar fractions of approximately 0.44. The HX only system performed significantly better than the HP only system, which is to be expected because energy purchased to run the HP compressor detracts from the system's solar fraction whenever the HP is called.

5.3.4 Solar Energy Collected

The total quantity of solar energy collected is also of interest in characterizing the performance of these systems. This quantity is taken from the energy balances carried out by TRNSYS on the storage tank and

the HP. The storage tank reports $\dot{E}_{DHW\ charge}$, the rate at which it is being charged based on the mass flow rate and temperature difference across the ports that are attached to the HX and HP. For the HX only system, the integral of this value is the solar energy collected over that period, E_{Solar}. For systems that incorporate a HP, it is necessary to also consider that the electric work on the compressor $\dot{W}_{el}$ increases the energy delivered to the load side $\dot{Q}_{Load}$. Neglecting stray heat transfers, an energy balance on the HP yields:

$$\frac{dE}{dt} = \dot{Q}_{Source} + \dot{W}_{el} - \dot{Q}_{load} \qquad (5-3)$$

The source side of the HP in these simulations is connected to the solar thermal collector, and the load side is connected to the storage tank through flow diverters. Therefore, $\dot{Q}_{source} = \dot{E}_{Solar}$ and $\dot{Q}_{Load} = \dot{E}_{DHW\ Charge}$. Substituting and integrating the above, it can be seen that for time steps where the HP operates,

$$E_{Solar} = E_{DHW\ Charge} - W_{el} \qquad (5-4)$$

The monthly solar energy collected by each simulated system follows a similar trend to the solar fraction and is not plotted here for concision; annual values are shown in Table 5-3.

Table 5-3 Annual Solar Energy Collected by simulated SDHW/SAHP Systems

	All Modes		HP Only		HX Only	AUX Only
	Non-Modulating	Modulating	Non-Modulating	Modulating		
Annual Solar Energy Collected [MJ]	5420.3	5426.4	4564.2	4562.2	5355.2	0

5.3.5 Purchased Energy

The purchased energy reported in this section is from two tracked values: the auxiliary electric resistance heater and the HP compressor's electric power consumption. Other power consumption due to ancillary devices in the system is excluded because they consume a small fraction of the power compared to the AUX heaters and the HP (e.g. pumps consume 76 W while the AUX heater consumes 3000 W). The monthly purchased energy consumed by each simulated system is plotted below in Figure 5-7, while annual values are shown in Table 5-4.

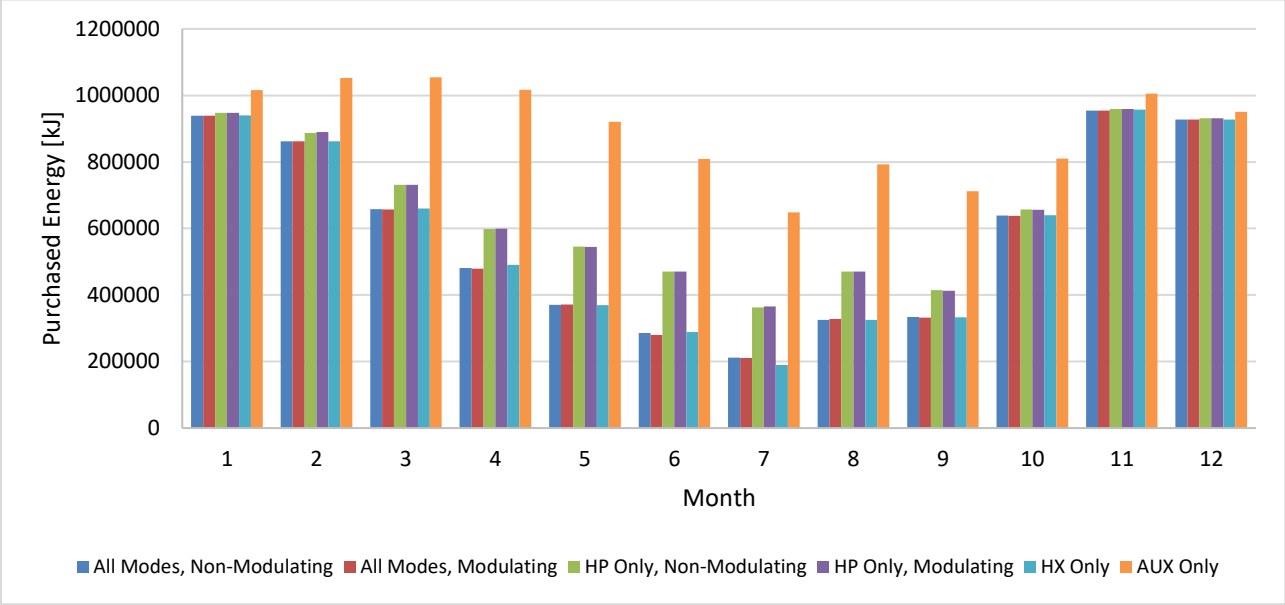


Figure 5-7: Monthly Purchased Energy for simulated SDHW/SAHP Systems

Table 5-4: Annual purchased energy consumed by simulated SDHW/SAHP Systems

	All Modes		HP Only		HX Only	AUX Only
	Non-Modulating	Modulating	Non-Modulating	Modulating		
Annual Purchased Energy [MJ]	6989.0	6980.9	7976.4	7979.3	6983.5	10792.0

5.3.6 System Adequacy and Other Metrics

The metrics presented in the previous sections compare the configurations based on energy considerations only. In this section an attempt is made to compare other aspects of the systems including their ability to deliver DHW at an appropriate temperature when required. Also considered in this section is the impact of modulating the HP compressor's speed on the number of compressor short cycles, and the duration of those cycles.

In the product standard for packaged SDHW (CAN/CSA-F379 Series-09) [47], the capacity test for a system involves measuring the volume of water that can be drawn from a charged tank before the outlet temperature falls below $35°C$. For this work, $35°C$ is used as a threshold temperature for a counter to keep track of the number and mass of DHW draws that were not sufficiently hot to meet demand. Table 5-5 shows this metric for the systems simulated.

Table 5-5: Annual DHW draws whose temperature fell below the required 35°C threshold

	All Modes		HP Only		HX Only	AUX Only
	Non-Modulating	Modulating	Non-Modulating	Modulating		
# of draws	130	130	111	113	150	351
Mass [kg]	3296.2	3304.5	3188.13	3200.5	3621.8	7025.3

As discussed in Section 5.3.1, the number of compressor cycles, and the duration of those cycles are expected to impact the lifespan and maintenance cost of a SAHP system. It was observed in shorter time-scale simulations that using a modulating HP compressor would decrease the number of times that the SAHP system short-cycles. The simulation output was post-processed to count the number of times during the annual simulation where the compressor was activated and stayed on for less than 5 minutes before turning off due to temperatures too high at the condenser or too low at the evaporator. For the "All Modes" and "HP Only" system, the total number of compressor cycles, and the number of compressor short-cycles are displayed in Table 5-6.

Table 5-6: Compressor cycles and short-cycles with and without modulating compressor control

	All Modes			HP Only		
	Non-Modulating	Modulating	% Reduction	Non-Modulating	Modulating	% Reduction
Total Number of Compressor Cycles	1374	1232	10.3%	2543	2382	6.3%
Number of Short Cycles	1237	1063	14.1%	2045	1913	6.45%

5.4 Conclusions and Recommendations

In this chapter, a set of 3 short time-span simulations were presented to compare the performance of SAHP systems incorporating modulating and non-modulating compressors, followed by annual simulations comparing similar SAHP and SDHW systems under different control regimes.

In the short time-scale simulations, both system configurations that incorporated a HP outperformed the traditional SDHW using a HX only, while the variable capacity HP outperformed the fixed capacity SAHP system. This is explained by the Variable-SAHP system operating more stably under low

irradiation conditions and experiencing fewer safety limits where the HP is forced to switch off. The Variable-SAHP is therefore a promising development in the area of SAHP-DHW systems.

Annual simulations compared the performance of multi-mode SDHW/SAHP systems using the solar fraction, solar energy collected, amount of purchased energy and other less common metrics. On the basis of solar fraction, there is a slight performance advantage to the "All modes" system compared to HP only and HX only systems. Ranking of the systems on the basis of the amount of solar energy collected follows a similar trend to that of solar fraction. The "All modes" system performed best by this metric, followed by the "HX Only" configuration, followed by the "HP Only" configuration.

On the basis of system adequacy, the "HP only" configuration had the lowest number of draws and the lowest mass of water drawn below $35°C$. The "All Modes" configuration had 3% more mass of water drawn below the temperature threshold, while the "HX Only" configuration drew 13% more water below the 35°C cutoff. Interestingly, this highlights that the total quantity of solar energy, and the solar fraction, are incomplete metrics for assessing a SDHW or SAHP system's performance. This is because the state of charge of a DHW tank varies based on the degree of stratification within it and a tank with a relatively high energy content still may not have sufficient hot water to meet demand, while a tank with a relatively low energy content but a higher degree of stratification might more readily supply a DHW load.

The difference in solar energy collected between non-modulating and modulating configurations of the "All Modes" system was 6.1 MJ, representing merely 0.1%. For the "HP Only" configuration, the difference in energy collection between a non-modulating and modulating compressor control scheme was 2 MJ which is approximately 0.05%. Similarly small differences in performance can be seen on all energy-based metrics when comparing non-modulating and modulating HP systems. The conclusion that can be drawn from these very small changes is that there is no performance penalty and no additional energy cost associated with having a modulating HP compressor.

Considering non-energy-based performance indicators, there is a significant increase in the performance of the HP system itself when a modulating compressor control scheme is employed. The total number of compressor cycles was seen to decrease by up to 10% when a modulating compressor was simulated for the "All-Modes" configuration. Furthermore, the number of short-cycles was reduced by 14%, which could significantly increase the lifetime of the system by reducing wear on the compressor and its starting components. Coupled with the finding that there is no energy-based penalty for enabling the 3-stage

modulating compressor control, it can be recommended that when such an option is available on a SAHP system, it should be employed.

Summarizing all of the above, it can be seen that a hierarchy of performance between SDHW, SAHP, and multimode systems can be established based on each of the metrics presented, but that those rankings do not necessarily agree when the metrics aren't solely based in the amount of collected solar energy. It is therefore recommended that future work undertake to develop a value-based metric for multi-mode SAHP-DHW systems to form a more robust criteria for system optimization, selection, and sizing.

Finally, whereas previous work has suggested that heat pumps can offer a performance benefit to solar-thermal systems but this simulation study did not sufficiently demonstrate that performance benefit, it is evident that there is a need to design and optimize a system controller that better utilizes the HP in the system to maximize solar collection potential and boost the capacity/adequacy of the system to meet DHW draw loads. The following Chapter details progress towards this goal through the introduction of a predictive-control approach.

Chapter 6
Controls Driven Performance Improvement

Previous work in this thesis and many studies reviewed have based SDHW and SAHP system operation on temperature-based controllers with setpoints and deadbands. This control scheme is by far the most common method in traditional SDHW systems going back to their inception [14]. Various works have called for further controls development [3, 17, 18, 19, 20].

While other researchers have suggested "optimizing" their control algorithms by performing parametric studies while varying temperature setpoints, this work applies a different control strategy as explained in Section 1.3.2 under Research Objectives. The methodology selected for this work is a predictive control with iterative update algorithm, which is implemented as described in the following sections.

The control improvements are based on principles similar to Model Predictive Control, although instead of the rigorous system identification and analytical modeling that would be required for that style of control, in the current work, a numerical model that has already been developed for the SAHP-DHW system under study is iteratively called to determine the forward-looking control series. The focus of this chapter is the use of the validated system model described previously in an iterative simulation that generates a control series through forward-looking predictions of system state over an emerging time-window.

6.1 Method

6.1.1 System Description

Although many thermal systems would be considered linear systems, there are three obvious non-linearities in the system of interest. First, the STC, which relies on radiative heat transfer from the sun has a 4^{th} order temperature response. Second, the HP, which was the subject of a previous chapter is theoretically governed by the Carnot cycle efficiency, which has a non-linear response to inlet temperatures. Third, the stratified DHW storage tank is a non-linear component as its state transformation from one time-step to another in simulation is governed by a non-linear system of ODEs that relate node temperatures in the tank to one another and the surrounding ambient

conditions. As a result, the system of interest can be considered a non-linear system for the purpose of controls development.

From a controls perspective, the state of the system of interest can be described by the variables listed in Table 6-1. The process variable of most significant interest (output) is the DHW draw temperature T_{draw}, which is the result of energy storage integrated in time from the initial condition, and complex mixing interactions with the DHW storage tank.

Table 6-1: State variable description for the system of interest

Variable (Units)	Description
$\boldsymbol{T_{DHW}}$ (°C)	Array of 20 temperatures representing the temperature profile of water in the DHW storage tank.
$\boldsymbol{T_{HX,Source\ In}}$ (°C)	Temperature of water flowing into HX Source side. This is the outlet temperature of the STC during HX Operation.
$\boldsymbol{T_{HX,Source\ Out}}$ (°C)	Temperature of water flowing out of the HX Source side. This is the inlet temperature of the STC during HX Operation.
$\boldsymbol{T_{HP,Source\ In}}$ (°C)	Temperature of water flowing into HP Source side. This is the outlet temperature of the STC during HP Operation.
$\boldsymbol{T_{HP,Source\ Out}}$ (°C)	Temperature of water flowing out of the HP Source side. This is the inlet temperature of the STC during HP Operation.
$\boldsymbol{T_{HX,Load\ In}}$ (°C)	Temperature of water flowing into HX Load side. This is the outlet temperature of the DHW tank charging port during HX Operation.
$\boldsymbol{T_{HX,Load\ Out}}$ (°C)	Temperature of water flowing out of the HX Load side. This is the inlet temperature of the DHW charging port during HX Operation.
$\boldsymbol{T_{HP,Load\ In}}$ (°C)	Temperature of water flowing into HP Load side. This is the outlet temperature of the DHW charging port during HP Operation.
$\boldsymbol{T_{HP,Load\ Out}}$ (°C)	Temperature of water flowing out of the HP Load side. This is the inlet temperature of the DHW charging port during HP Operation.

The input variables to the system of interest are listed below in Table 6-2. None of these inputs can be considered constant during the operation of the system.

Table 6-2: System input variable description for the system of interest.

Variable (Units)	Description
$T_{Amb,indoor}$ (°C)	The air temperature surrounding components housed indoor. Drives the loss functions for the DHW tank, and other indoor components.
$T_{Amb,outdoor}$ (°C)	The air temperature surrounding components housed outdoors. Drives the loss function for the STC.
T_{Mains} (°C)	The temperature of mains water inlet to the system. When DHW is drawn, it is replaced with mains water at the bottom of the DHW tank.
$\dot{m}_{Draw}$ (kg/min)	The mass flow rate of DHW draws removed from the system. This also represents the mass flow rate of mains water flowing into the system. This variable is forced by an external schedule.
V_{wind} (m/s)	Outdoor weather wind speed. Drives the loss function for the STC.
$G_{t,t}$(w/m^2)	Total flux rate of solar radiation incident on tilted plane.
$G_{d,t}$(w/m^2)	Flux rate of diffuse solar radiation incident on tilted plane.
θ_z	Solar zenith angle, the angle between the vertical and the location of the sun. Impacts the collection efficiency and the amount of energy incident on the STC.
γ_s	Solar azimuth angle, the angle in the horizontal plane between true north, and a point directly below the sun. Impacts the collection efficiency and the amount of energy incident on the STC.

The control variables for the system of interest are listed in Table 6-3 on the following page. There are a mix of discrete mode, on/off, and continuous variables that the controller can use to influence the operation of the system of interest.

Table 6-3: Control variable description for the system of interest.

Variable (Units)	Description
Controller Mode	Discrete mode of operation to select heat transfer component: Off, HX, HP, Warmup.
HP Call	Boolean variable representing whether the HP compressor is on or off.
HP Speed	Continuous variable between 0% = 0 RPM and 100% = 3600 RPM.
Pump 1 Flow Rate	Continuous variable representing the flow rate of water on the source-side.
Pump 2 Flow Rate	Continuous variable representing the flow rate of water on the load-side.
AUX Heat Call	Boolean variable representing whether the AUX Heat is on or off.

6.1.2 Simulation Architecture

A system of interacting simulations was developed to link multiple instances of TRNSYS through custom MATLAB scripts. The simulation architecture utilized in this study is diagramed in Figure 6-1. Source code of the MATLAB scripts and functions used can be viewed in Appendix D.

As will be shown, the simulation and control architecture for this system are divided between a top-level "Main simulation", and iterative "Sub-simulations" used to make control decisions. For clarity, the author wishes to distinguish in this chapter the terminology used while referring to data that are read into the two levels of simulation. The terms "Weather File" and "Draw Profile File" represent the forcing functions on the top-level simulation that stands in for real-world weather and loads demanded of the DHW system by building occupants. The terms "Weather Forecast" and "Draw Predictions" represent data that are supplied at intermediate sub-simulation steps, and used as forcing functions by the predictive controller. In reality, the weather forecasts and draw predictions would be known with much less certainty. For the work presented in this chapter, these data come from the same sources (TMY weather for Ottawa and Edwards "House 38" [48]) as the main simulation inputs but, as will be discussed later with the conclusions and recommendations, an important next-step to improve the realism of simulation results employing the architecture presented here will involve

replacing the data supplied to the sub-simulations with either statistical models or data sourced from a weather forecasting agency to better represent the effect of uncertainty in the inputs propagating through the controller's process model and impacting system performance.

The top level of this simulation environment is a TRNSYS instance of the system model described in Chapter 4. This operates in place of a physical system. The SAHP-DHW system is modeled for long simulation runs up to 1 year. As in the annual simulations presented in Chapter 5, the weather file is the TMY file for Ottawa, Ontario, and the DHW draw profile file is from Edwards "House 38" [48]. At every timestep, TRNSYS calls the main MATLAB control script, passing to it the current system state (Temperatures, Flow Rates, Current operating Mode, etc.), and the current weather conditions and time of the simulation. At every time step, TRNSYS receives back from MATLAB a control vector consisting of the operating mode to apply for the next time steps, circulator control signals, diverter valve control signals, auxiliary heat control signals, etc.

The main MATLAB control script receives its inputs from the main TRNSYS model. It also reads in data about the draw predictions and the weather forecast. The main control script maintains a history data structure used for simulation output. It makes its control decisions by calling subordinate TRNSYS simulations through MATLAB sub-simulation functions. The sub-simulations run over a shorter simulation period and are called at regular intervals by the main simulation. The MATLAB control script passes the current system state, weather and draw forecasts to the sub-simulations.

From the MATLAB sub-simulation script, TRNSYS sub-simulations are iteratively called and a system model executes various candidate control series. Between each sub-simulation, faults are detected and system performance is analyzed. The sub-simulation functions alter the candidate control series with each iteration attempting to minimize the projected energy deficit of the DHW tank with respect to the anticipated DHW draws.

Once the sub-simulations converge (i.e. the candidate control series isn't changing from one iteration to the next), the sub-simulation functions return a time series of control signals back to the main simulation, which operates the system using those signals until the next sub-simulation is triggered.

Predictive Controller Simulation Architecture:

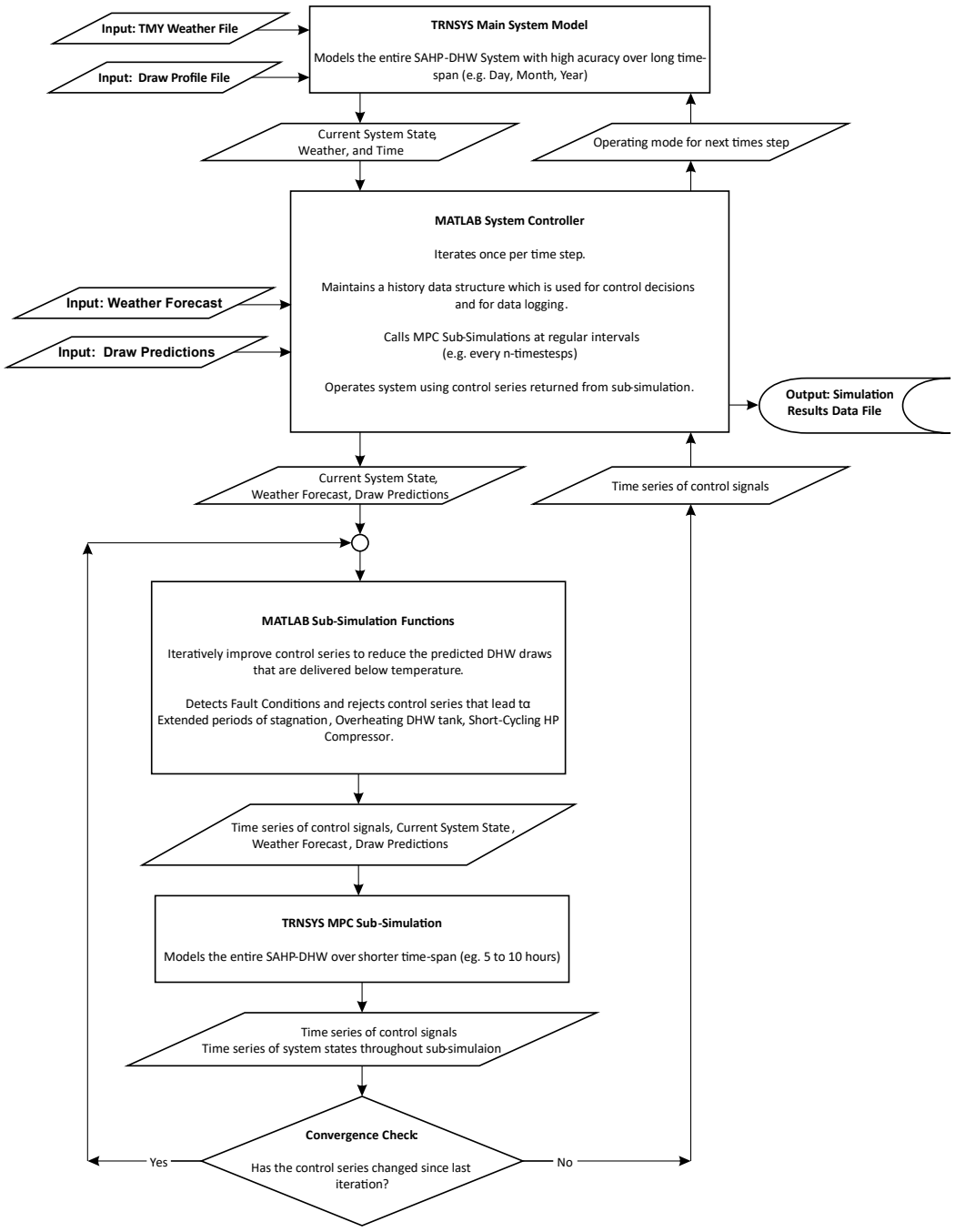


Figure 6-1: Simulation Architecture for Predictive Controller of TRNSYS SAHP-DHW Model

6.1.3 Fault Detection

There are several possible faults that could occur in the system being studied. In addition to satisfying the objective function described in the next section, the control system must avoid operating control sequences that lead to these fault conditions. Below are descriptions of the fault conditions considered in this work.

A well-known fault in solar-thermal systems is system *stagnation*, which occurs when the flow rate of heat transfer fluid in the STC is zero during a period when there is significant radiation incident on the STC. In this situation, the temperature of the heat transfer fluid climbs rapidly, which can result in one of the following undesirable outcomes [14]:

- Boiling of the heat transfer liquid leading to escape of the fluid through relief valves or potential pipe-burst in plastic plumbing systems
- Denaturing of any antifreeze solution in the heat transfer fluid, in the case of typical Glycol solutions, leading to loss of freeze-protection and the circulation of a corrosive byproduct throughout the system.
- Overheating of spectrally selective coatings on collector surfaces, causing damage and permanent degradation in performance.

There are limits to the range of temperatures under which the HP can operate. As demonstrated in Chapter 5, low source-side inlet temperatures can lead to undesirable HP cycling. In this study, *HP under-temp* is treated as a fault condition that can be identified within sub-simulation results.

A final kind of fault, which would be noticed quite readily by a user of the DHW system being studied, is the delivery of under-temperature water. Failing to meet the load requirements results in DHW being drawn from the tank below the required temperature. As with the system capacity metric presented in the previous chapter's Section 5.3.6, 35°C is used as a threshold *for DHW under-temperature* fault detection.

After each iteration of the TRNSYS sub-simulation, a MATLAB function parses the simulation output to find instances of the faults described above. In concert with the controller's attempt to minimize the objective function described in the following section, control sequences that lead to fault conditions are flagged by the sub-simulation functions so that operating modes that lead to fault conditions in the sub-simulation time-frame can be avoided.

6.1.4 Objective Function

The end-purpose of a SAHP-DHW system is to supply water of a sufficient temperature to building occupants. Components in the SAHP-DHW system affect the temperature of water drawn from the DHW tank by additions of energy to the tank. The objective function to be minimized through the iterative sub-simulations is the "Draw Energy Deficit" which is calculated after first identifying DHW draw events where the temperature was below the required temperature (set at 35°C in this work), then summing the energy content of those under temperature draws and taking the difference between that energy content and the energy content of a draw at a constant 35°C.

$$E_{deficit} = \sum_{T_{Draw}<35°C} E_{35°C\ Draw} - \sum_{T_{Draw}<35°C} E_{Undertemp\ Draw} \qquad (6-1)$$

A draw energy deficit of zero indicates that all draws within the simulation window met the minimum temperature of 35°C. Any positive value of draw energy deficit indicates that more heat was required in the time preceding that draw event. Although it is possible for the system to be in a "draw energy surplus" condition, this condition is not represented as a negative value in $E_{deficit}$ because draws where $T \geq 35°C$ are excluded from the summation in Equation 6-1. Therefore, the minimum value of $E_{deficit}$ is zero. As will be discussed in the next section, this variable is the signal to the controller to activate various modes (HX, HP, and/or AUX) to reduce or eliminate this deficit.

Closely related to the draw energy deficit is an additional performance metric "available energy", which represents the energy content of water in the DHW tank held above the temperature threshold. It is defined as follows:

$$E_{Available} = \rho\ A\ c_p \int_{z_{threshold}}^{z_{max}} T\ dz \qquad (6-2)$$

The complement of the available energy metric is the "unavailable energy" metric, which would represent the energy content of water in the tank below the temperature threshold.

$$E_{Unavailable} = E_{DHW\ Tank} - E_{Available} \qquad (6-3)$$

$$E_{DHW\ Tank} = \rho\ A\ c_p \int_{z_{min}}^{z_{max}} T\ dz \qquad (6-4)$$

In the equations above, Energy E is calculated in kJ, the density of water ρ is in kg/m^2, the specific heat capacity of water c_p is entered in kJ/kg °C, z is in m, and T is in °C. The reference point of

$E_{Available} = 0$ occurs when there is no water in the DHW tank above the threshold temperature (e.g. 35°C). $E_{Unavailable} = 0$ could occur either if the entire contents of the DHW tank was at 0°C, or when all of the water in the DHW tank is at or above the threshold temperature. $E_{DHW\ Tank} = 0$ can only occur if the entire tank contents are at 0°C, which is outside of the range of temperatures simulated as freezing would lead to discontinuous heat capacity and density of water.

Not all energy added to the DHW tank from HX, HP and AUX directly affects the temperature of DHW draw water. Due to the stratified nature of the DHW tank, there is a non-linear relationship between energy additions to the tank and the DHW draw outlet temperature. This is demonstrated in Figure 6-4 and Figure 6-5 showing available and unavailable energy contained in the DHW tank and accompanying DHW tank charge rate and controller mode over a sample 24 hours. These simulation results are extracted from a larger annual simulation using the model inputs specified in Section 4.5. The example day presented below uses model input for July 13th in the Ottawa TMY file and Edwards House 38. Solar irradiation and ambient temperatures are plotted below in Figure 6-2. The draw profile and mains inlet temperature for the day follows in Figure 6-3.

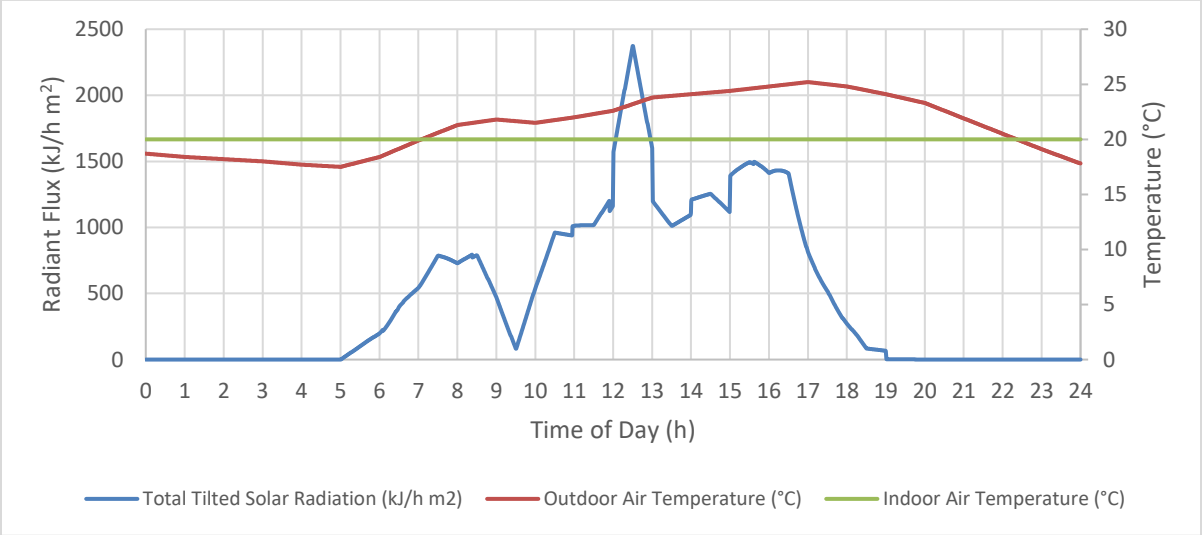


Figure 6-2: Solar radiation and ambient temperature inputs to the model for July 13th example simulation

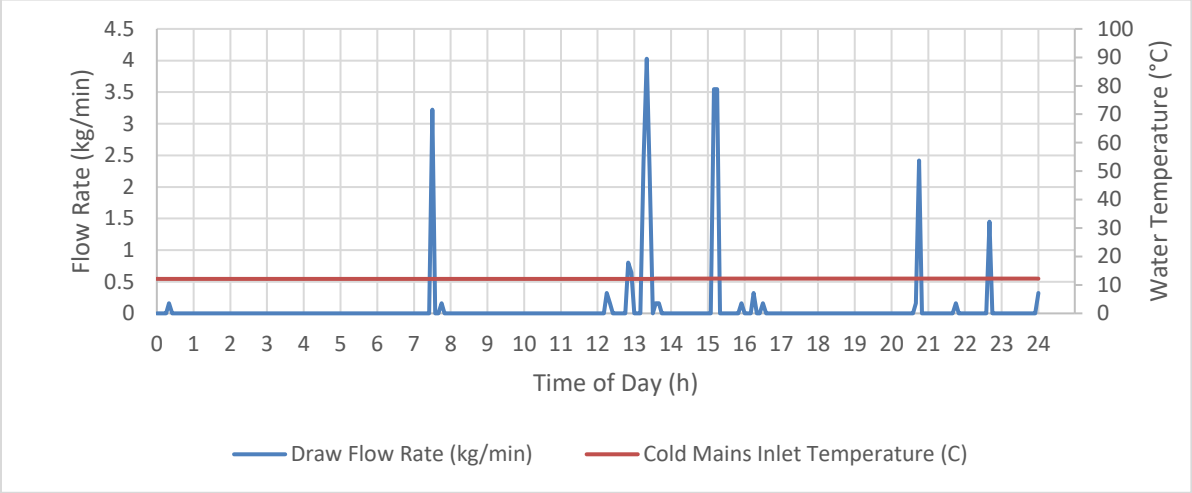


Figure 6-3: Draw flow rate and mains inlet temperature inputs to model for July 13th example simulation

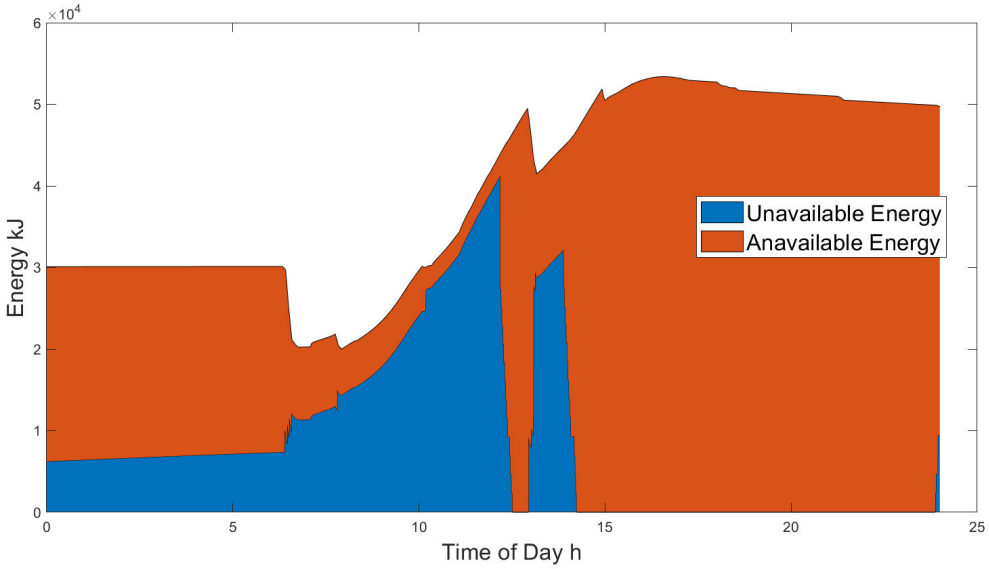


Figure 6-4: DHW Tank energy contained in available and unavailable state for an example 24 hours of system operation

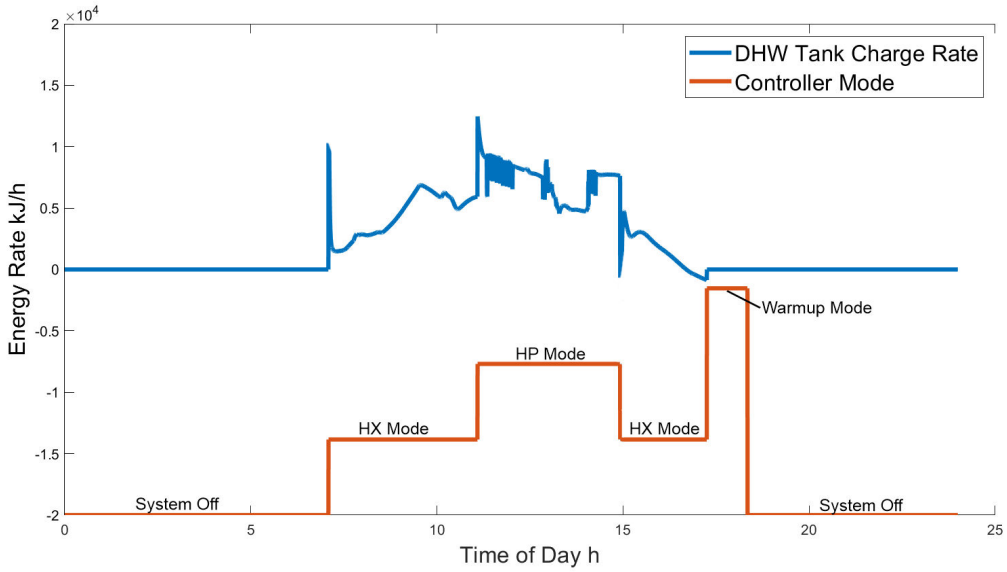


Figure 6-5: DHW Tank charging rate and accompanying controller mode for an example 24 hours of system operation

The sample day shown above was selected because, unlike previous examples shown in this thesis, July 3rd in the Ottawa TMY file is an uncharacteristically cloudy day with interruptions in solar irradiation. This low-solar-input day makes for a more interesting control problem and charging profile to demonstrate the utility of the "Available Energy" metric.

It is observed that from 7h to approximately 10h, despite considerable energy gains operating in HX mode, the available energy of the tank decreases. This is due to mixing of the tank that occurs when the circulator pumps are activated. Later in the day, from approximately 11h to 12h, under HP mode, the DHW charge rate is at its maximum for the day, and the total energy in the tank increases rapidly, but still the amount of water above 35°C is quite small and therefore the available energy is low compared to the unavailable energy. After 12h, the available energy outstrips the unavailable energy rapidly despite the energy collection rate of the system being lower than it was at its peak. Later in the day, a DHW draw removes hot water from the top of the DHW tank and replaces it with cooler water at the bottom, now there is a band of unavailable energy represented once again.

This non-linear connection between the energy addition rate and the available energy in the DHW tank, along with the requirement to satisfy the fault constraints described in Section 6.1.3 lead to the application of an iterative solution methodology to minimize the objective function. The algorithm used to develop candidate control series is described in the next section.

6.1.5 Generation of Control Series

The candidate control series is stored as an $n \times 5$ array for each timestep between the time that the sub-simulation is called and the prescribed time horizon. In the first column is the time of year for which that row of the array applies. The second column holds a value of 1 or 0 to indicate whether solar modes are permitted during that timestep. The third column holds the controller mode command, an integer between 0 and 3. The fourth column holds a value of 1 or 0 and controls the AUX heating element. The fifth column holds a 1 or 0 to indicate whether HP operation is permitted. Explanations of each control field and its possible values are show in Table 6-4.

Table 6-4: Candidate control series field values and description.

Control Field	Value(s)	Description
Field 1	0.000 to 8760.000	Time of year (h)
Field 2	1	HX, HP permitted (daylight hours)
	0	HX, HP not permitted (outside daylight hours)
Field 3	0	System off
	1	HX Charging
	2	HP Charging
	3	Warmup Mode
Field 4	1	AUX Heat On
	0	AUX Heat Off
Field 5	1	HP Permitted
	0	HP Locked Out (e.g. cycling)

The algorithm developed to iteratively generate the candidate control series using system model results is represented in the flow chart in Figure 6-6. This sequence of operations is carried out once per iteration of the loop previously shown in Figure 6-1.

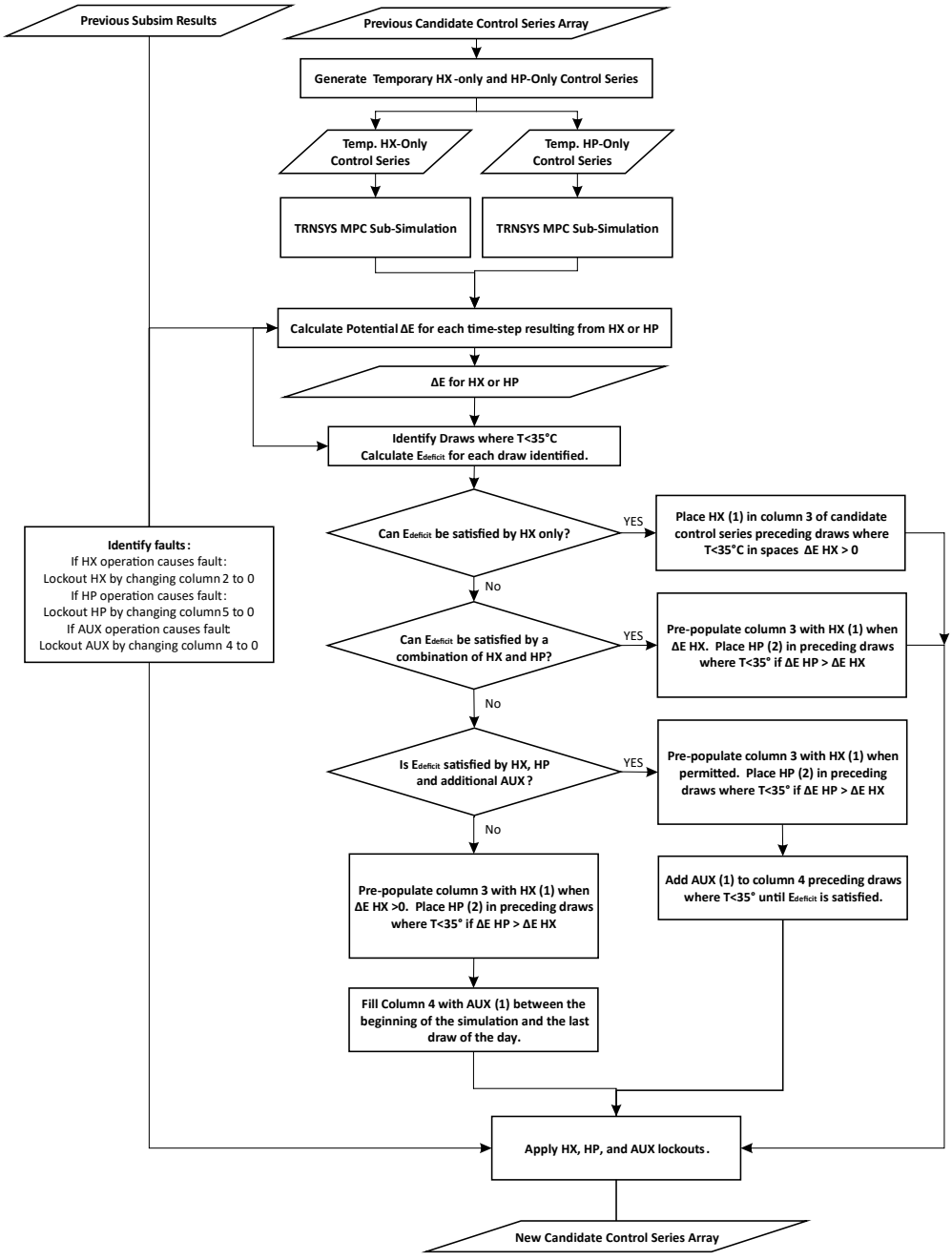


Figure 6-6: Simplified algorithm to generate an iterative update of the candidate control series

For the first call to the sub-simulation, the candidate control series is initialized with zeros, then the first column is pre-populated with a grid of times stepping forward from the initial time, by the simulation timestep, to the prescribed time horizon. Next, the weather forecast (currently still read from a TMY file) is parsed to determine daylight hours and column 2 is updated for the timeseries.

At each update stage, the algorithm generates two temporary simulations used as a basis to determine approximate HX and HP operating conditions for each time step. These temporary simulation outputs are used to evaluate ΔE_{HX}, ΔE_{HP}, which represent the difference in energy collection potential for each time step if HX or HP were activated, relative to the previous sub-simulation's result for that timestep. Another energy difference ΔE_{HP-HX} represents the difference in collection potential for each timestep comparing the HP to the HX.

Using these ΔE measures, it is possible to approximate how many time steps of HX, or HP operation would be required to satisfy the energy deficit due to under-temperature DHW draws. The script attempts to minimize or eliminate $E_{Deficit}$ by placing control commands into the timeseries preceding the identified draws. A pictographic example of a control series that might result can be seen on the following page in Figure 6-7.

In Figure 6-7, timestep *k* is identified by the crossing of T_{DHW} from above to below 35°C. In time steps preceding this event, the algorithm schedules blocks of HX, HP, and AUX heat addition to the tank. Future iterations of the sub-simulation will preserve these scheduled heat additions unless they evolve to cause a fault, at which point they may be removed and locked out.

If the initially scheduled heat additions still fail to satisfy the DHW draw $E_{deficit}$, more heat addition blocks will be scheduled in any available time steps between sunrise and time step *k*. This process is repeated for each under-temperature draw event, and reiterated until either the sub-simulations converge (the candidate control series stops changing), or a maximum number of iterations is reached.

The priority of scheduled heat additions is preserved from Chapter 5, so HX operation is scheduled first, then HP, followed by AUX. Generally, the most energy that can be added to the DHW tank is when HP is scheduled for all daylight hours (subject to freeze and short-cycle protection) and AUX is scheduled on top of the HP. If this condition still fails to satisfy the $E_{deficit}$, it indicates that the draw load exceeds the overall capacity of this SAHP-DHW system.

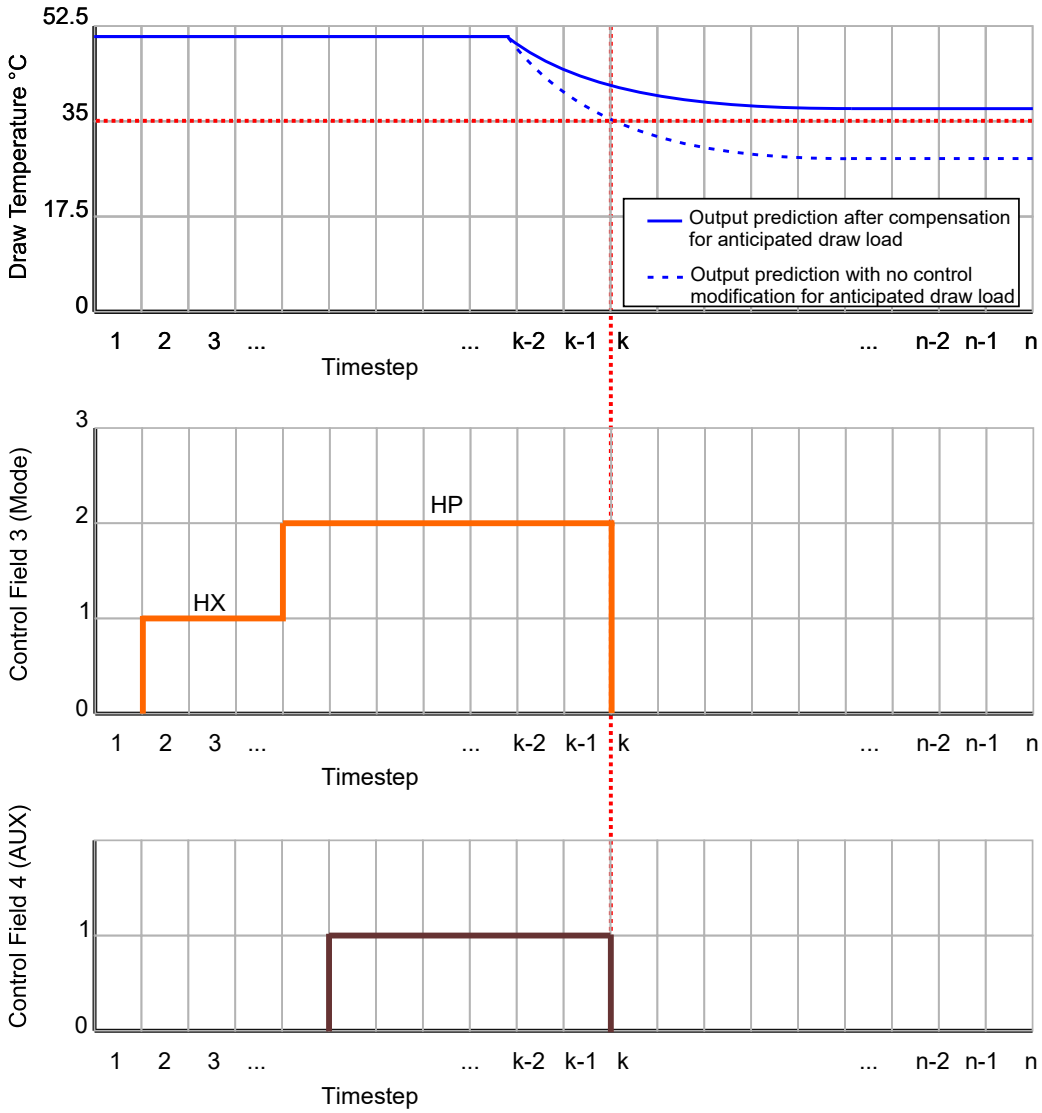


Figure 6-7: Example control series showing heat addition scheduled ahead of a detected under-temperature draw event

6.1.6 Control Series Aggregation and Time Horizon

Once the sub-simulations have converged on a candidate control series, the signals must be incorporated into the main control series. The main control script only needs control fields 1 (time), 3 (Mode), and 4 (AUX) to operate the system. This subset of the candidate control array is aligned with the main simulation control array based on the time values in field 1.

Various schemes were considered for aggregating candidate controls and calling sub-simulations, including different durations of sub-simulation and various degrees of overlap between sub-simulation. Ultimately, a sub-simulation length of 10 hours was selected, and the scheme was set so that sub-simulations overlap by 5 hours. In this scheme, the time-horizon of the controller is always at least 5 hours in the future, but displaces 5 hours further into the future when a sub-simulation is called. This aggregation scheme is illustrated in Figure 6-8 below.

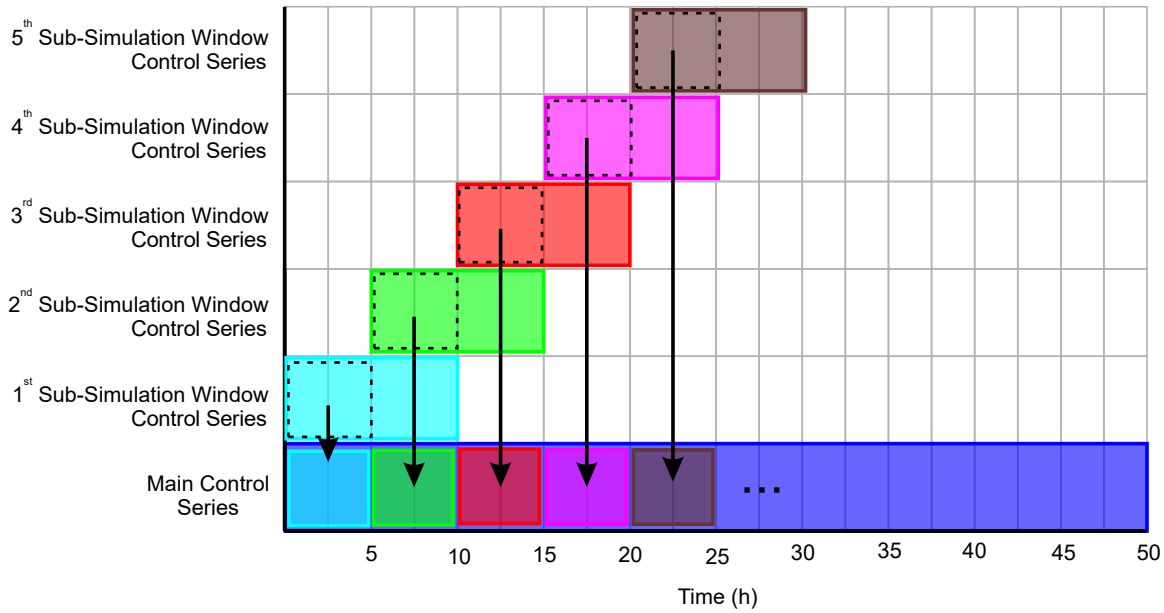


Figure 6-8: Aggregation of sub-simulation candidate control series into main control series

The time horizon is limited to 10 hours for two reasons. First, it is desired to stay within the time-range for which the overall system model showed agreement with experimental results in the validation trials presented in Chapter 4.7. It is also desired to have long enough sub-simulations that energy demands that are reasonably far in the future will be accounted for in the control decisions made at the current time-step.

As the main simulation evolves, successive sub-simulations are called at regular intervals with initial conditions set at the current system state of the main simulation. The candidate control series for each sub-simulation is initialized with the candidate control series from the overlapping part of the previous sub-simulation window. In doing so, the iterative portion of the scheme that generates the new candidate control series carries information forward about faults that were previously identified and resolved, but proceeds to adjust the control series for the entire sub-simulation window to address any energy deficit or faults that emerge in the later part of the time window.

6.2 Results and Analysis

The architecture and control scheme described in Section 6.1 were used in an annual simulation using TMY weather and Edwards' draw profile. Simulation results are discussed and analyzed in the following sections. The annual inputs to the model are not plotted for concision, however when representative sections of the simulation results are extracted and presented, the corresponding parts of the model input data are plotted adjacent.

6.2.1 Visualizing the Iterative Control Process

In generating the control series for the main simulation, many possible control solutions are iterated through. As described in Section 6.1.5, the control series is selected on the basis of the least energy deficit while avoiding other identified fault constraints. The sample shown below is for 2 consecutive days June 2nd and 3rd from TMY weather for Ottawa and Edwards draws. Weather inputs are plotted first in Figure 6-9, followed by draw information in Figure 6-10.

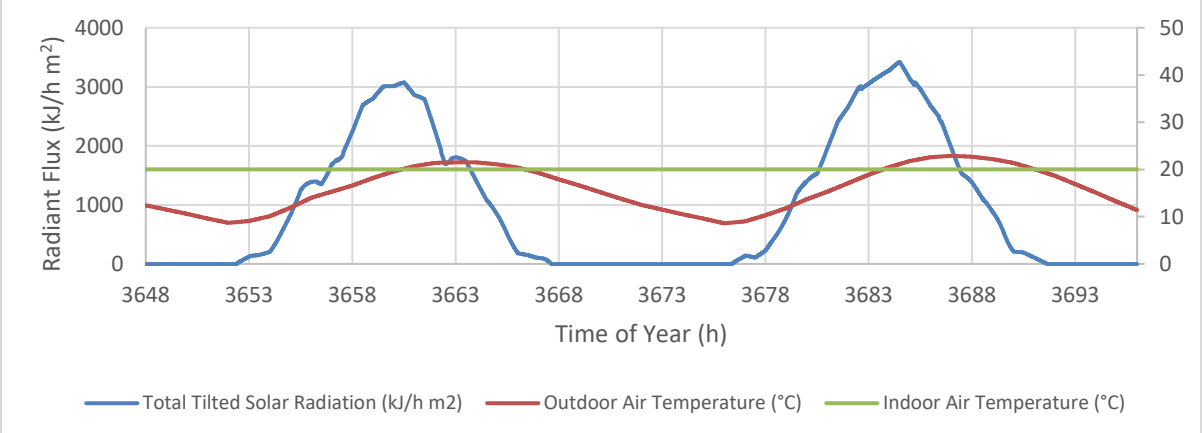


Figure 6-9: Solar radiation and ambient temperature inputs to the model for June 2nd & 3rd

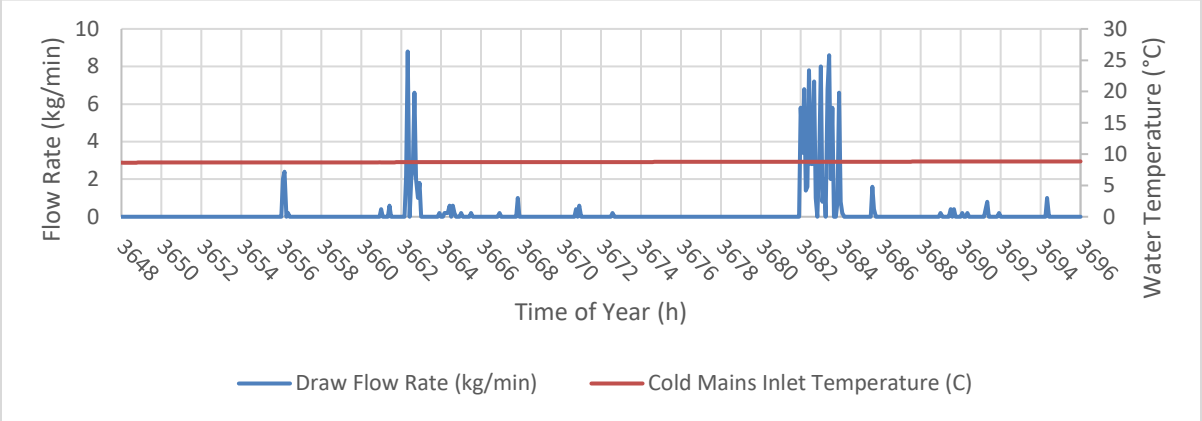


Figure 6-10: Draw flow rate and mains inlet temperature inputs to model for June 2nd & 3rd

In the figures below, dashed lines of any colour represent possible paths that the system state could follow. These possible paths are sub-simulation results for an intermediate candidate control series as it is iteratively improved. The solid blue line in the plots below represents the chosen path, which is the simulation result after the best candidate control series is passed back up to the main control level.

In Figure 6-11, the primary (top) axes contain a plot of available energy versus time of year. Available energy is used as a proxy for the objective function for visualization, since the energy deficit is only evaluated by the controller when draws are occurring. It should be understood that any instance where the available energy metric goes to zero that coincides with a draw event would produce an energy deficit.

Detected faults are highlighted in red for stagnation, magenta for HP cycling, cyan for system over-temperature, and yellow for under-temperature draws. In Figure 6-11, red, magenta and yellow highlights can be seen. There is no system over-temperature fault in the operation of this system because the draw schedule and control scheme reliably leave enough reserve capacity to charge the tank that the condition doesn't occur. The fault detection is left in place to provide a safety limit.

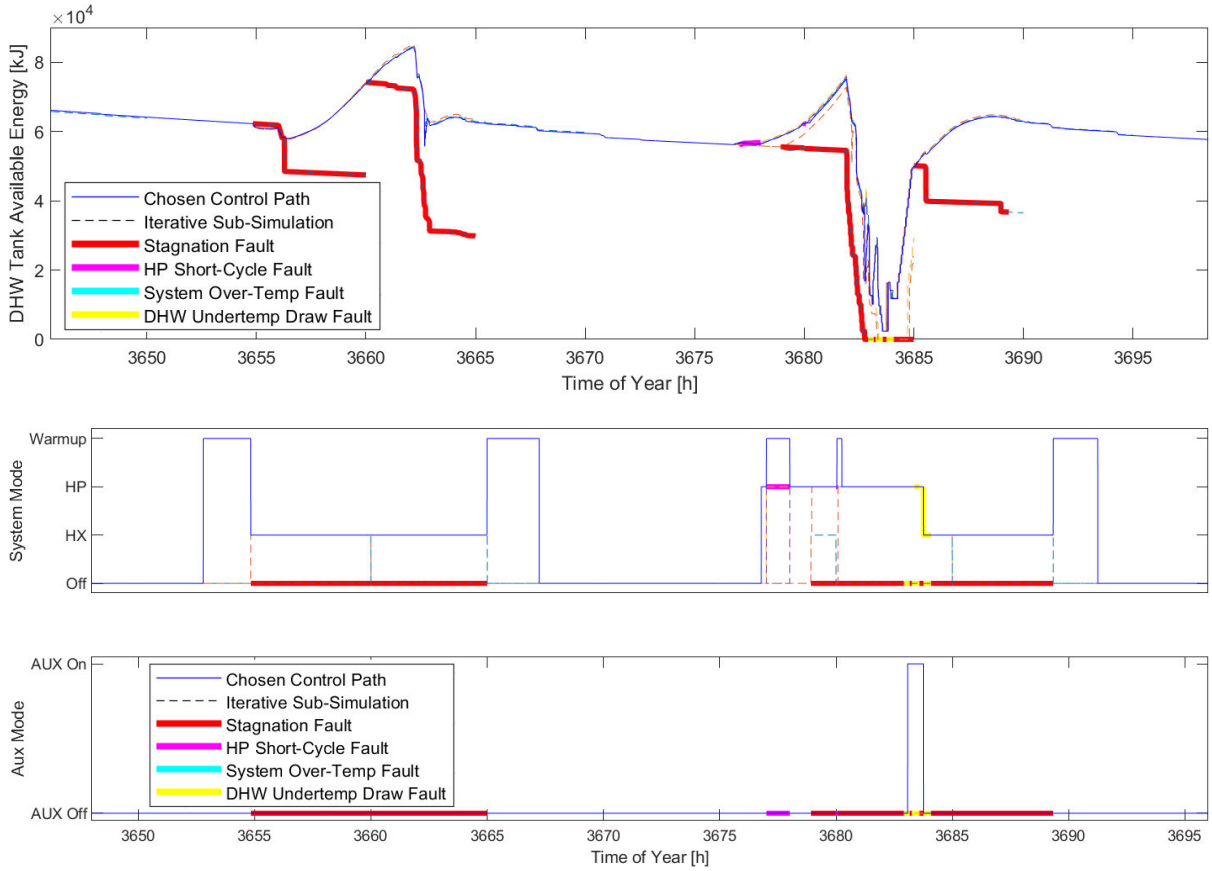


Figure 6-11: Visualization of multiple sub-simulation control paths and the chosen control path

Worth noting in Figure 6-11, is that the first day of operation pictured begins with the DHW tank pre-charged with enough available energy to meet the draw demands that day. This means that the only required controller decisions on this day are to address the risk of stagnation, which is achieved by the operation of the system in warmup and HX modes symmetrically throughout the day even when there was no energy deficit. Sub-simulations during this period assessed the possibility of resting the system in "off" mode, but avoided that mode because it led to stagnation faults which are seen highlighted in red on the plots. The blue trace deviates from the dashed line that corresponds to the stagnation faults.

The second day of operation shown in the plots above has a significantly higher DHW draw schedule. This results in the possible depletion of the available energy within the DHW tank if the system were to only operate in HX and warmup mode as it had the previous day. The controller therefore identifies a need for additional energy in the tank and schedules charging modes in escalating order ahead of the period highlighted in yellow (detected DHW under-temp event). It can be seen that in

the process of addressing the under-temperature event at 3683 h, the controller schedules HP for the entire daylight hours preceding that fault. A later sub-simulation predicted HP short-cycling near the beginning of the operation period (highlighted in magenta) at 3677 h. This short-cycling was addressed by the controller by switching to "warmup" mode for 1 hour. Replacing some HP operation with warmup to address short-cycling leaves some energy deficit that cannot be met by the solar modes, so auxiliary heat is required. The controller schedules auxiliary heat at the last moment before the draw and re-simulates to ensure that the demanded load is met. This call for auxiliary heat can be seen in the bottom set of axes.

Controlling to minimize the energy deficit is semantically the same as controlling to match the available energy to the DHW draw load, which is why that metric is plotted above. It can be seen that during the second day of operation plotted, the blue trace representing the available energy of the main simulation under the chosen control series approaches but does not touch zero at 3683 h. This indicates that for the period preceding this moment, the available energy has been kept reasonably near the minimum amount that was required to meet the draw load. Inconveniently, with a goal of keeping the available energy low, the dotted traces of sub-simulation results are often close-to or appear coincident with the blue main simulation results even if the control series that led to those predicted conditions varies. Inspecting a similar plot of average tank temperature over the same simulation window, we can see that the state of the system does vary significantly between the various sub-simulated results and the final main or chosen result.

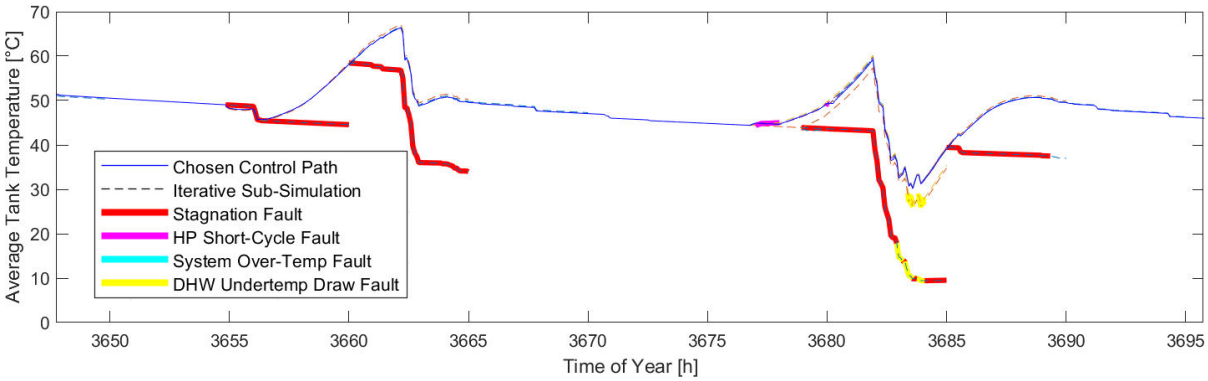


Figure 6-12: Average DHW tank temperature showing variation between results from candidate controls and main simulation results

In longer simulations, the control series and individual system state variables become background data handled by the controller but not as easily inspected or interpreted by the system operator. The

following sections present monthly and annual results using similar KPIs to those used in Chapter 5. In these longer simulations, the improved system under predictive control is compared against the system referred to as "All Modes, Modulating" in Chapter 5. As a reminder, the system from Chapter 5 is a benchmark of the same simulated hardware under a typical ΔT-based controller.

6.2.2 Solar Fraction

Figure 6-13 below shows the Solar Fraction, f_i, on a monthly basis extracted from an annual simulation. In blue are the results of the most recent simulation controlled using the scheme described above. For comparison, the simulation results from a comparable system with a temperature-based controller are plotted in red.

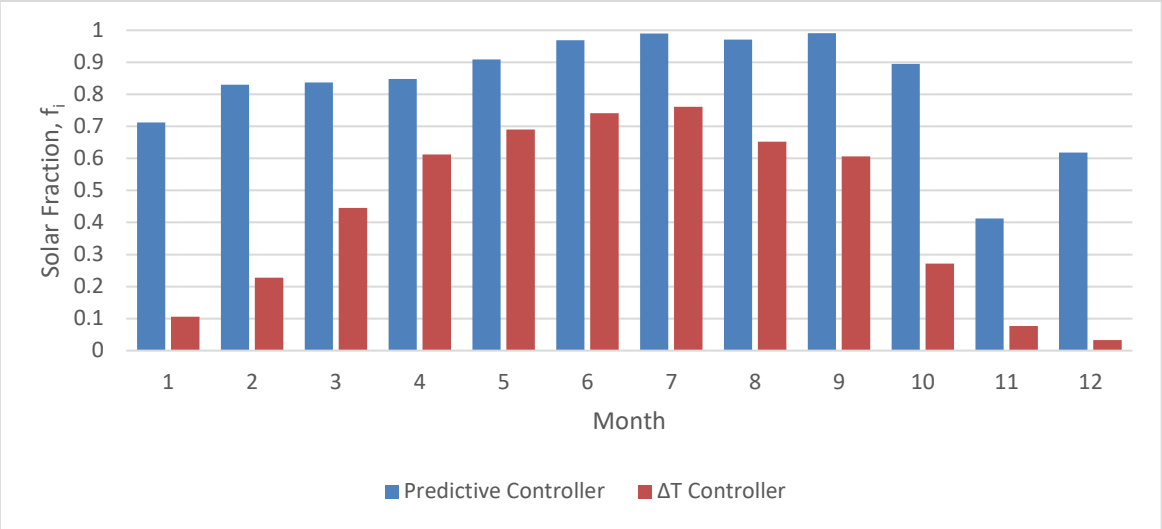


Figure 6-13: Monthly solar fraction resulting from predictive control based SAHP-DHW system compared with ΔT based controller

In the figure above, a significant increase in solar fraction can be seen. This increase is owing to two main contributing factors. First the predictive controller operates the HP more frequently than the ΔT controller allowing its benefits to be utilized to boost the collector outlet to usable temperatures when the HX cannot operate. This offsets auxiliary usage and therefore increases the solar fraction.

The second contributing factor is the elimination of auxiliary elements pre-heating the DHW tank overnight under the ΔT based controller. In the predictive control scheme, auxiliary heat is used as a last resort, only when the controller predicts an energy deficit at the time of a load. With the ΔT controller, the auxiliary heat runs until the tank is charged to setpoint overnight, leaving the system state of charge

relatively "full" going into the next morning, which limits the amount of solar energy that can be collected with the HX or HP.

Worth noting in Figure 6-13 as well, is that the predicted performance improvement during fall and winter is substantial. Referring to the baseline cases in Chapter 5, this control methodology enables the system to go from essentially inactive from November through February to a substantial contributor to DHW loads year-round by saving up charging capacity, and operating in the correct mode for the current conditions. One part of these optimistic predictions that has not been explored, however, in the current study, is freeze protection, and the temperature of heat-transfer fluid in the collector during periods of low-insolation and extremely low ambient temperatures. This would need to be provided for with antifreeze in a real system, and may require refinement of the system model in future work.

Table 6-5 below compares the Annual solar fraction, $\mathcal{F}$, for the most recently described controller against the ΔT-based system.

Table 6-5: Annual solar fraction of simulated predictive and ΔT based systems

	Predictive Controller	***ΔT* Based Controller**
Annual Solar Fraction	0.84	0.43

Similar to the results of the monthly solar fraction, the increase in performance reported by this metric is substantial. It must be noted again that while common and useful for comparing similar systems, the solar fraction's description of the system performance is limited as it does not take into account parasitic loads or standby tank losses. A case-in-point is that this system's high solar fraction describes a system that collects large amounts of solar energy throughout the year, but as we will discuss in Section 6.2.6, the system also experiences significantly higher losses than the system under ΔT Based control.

6.2.3 Solar Energy Collected

The amount of solar energy collected in the simulated predictive system is seen contrasted against the ΔT based system in Table 6-6 below. This quantity is not plotted monthly as it follows an analogous trend to the solar fraction plotted in the previous section.

Table 6-6: Annual solar Energy Collected of simulated predictive and ΔT based systems

	Predictive Controller	ΔT Based Controller
Solar Energy Collected [MJ]	11157.2	5624.9

As expected for a system with nearly double the solar fraction, the simulated solar energy collected by the predictive control based system is 98% greater than the result of the ΔT based system. The two systems are subject to the same draw schedule and therefore the draw load is the same between the two simulated systems. A whole system energy balance shown in Equation 6-5 allows us to account for how increases or decreases in simulated energy quantities offset each other.

$$E_{Collected,Solar} + E_{Purchased} - E_{Loss} - E_{Draw,Delivered} = \Delta E_{Stored,Year-over-Year} \qquad (6-5)$$

The majority of thermal capacitance in the simulated system is in the 364 L DHW tank, which sees temperatures that are never below the freezing point of water, 0°C and never above the boiling point 100°C, so an estimate for the largest $\Delta E_{Stored,Year-over-Year}$ that could occur is if the tank's average temperature varied by an absolute value of $100°C$ over the year. This can be computed as:

$$\Delta E_{Stored,MAX} = \rho c_p V \Delta T = \left(998\frac{kg}{m^3}\right)\left(4.19\frac{\text{kJ}}{\text{kgK}}\right)(0.364\,m^3)(100°C) \approx 1.52 \times 10^5\ kJ, or\ 152\ MJ$$

$\Delta E_{Stored,Year-over-Year}$ in the equation above is several orders of magnitude smaller than the other annualized energy quantities tracked by the simulation and can therefore be neglected for annual simulations, Equation 6-5 then becomes:

$$E_{Collected,Solar} + E_{Purchased} \approx E_{Loss} + E_{Draw,Delivered} \qquad (6-6)$$

It can therefore be seen that for systems delivering equivalent draw energy, there is a direct trade-off between purchased energy, collected solar energy, and system losses. The purchased energy is presented in the following section, and system losses will be quantified and discussed following that.

6.2.4 Purchased Energy

Monthly purchased energy follows the inverse trend compared to the solar fraction and solar energy collection. To run the HP compressor and the AUX elements, more electrical energy is purchased in the fall and winter than in the spring and summer months as shown in Figure 6-14.

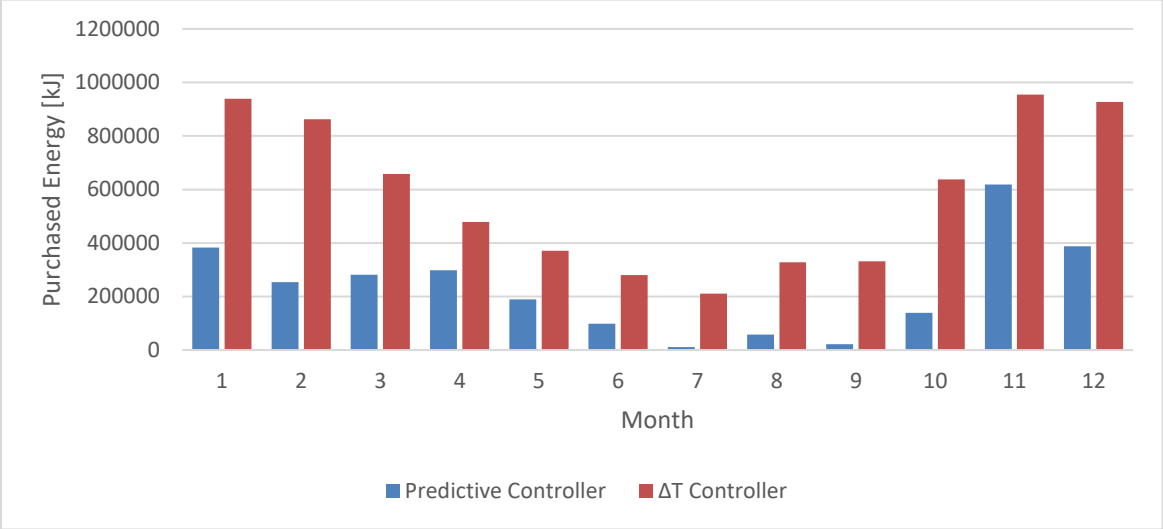


Figure 6-14: Monthly Purchased Energy for simulated predictive and *ΔT* controlled systems

In the predictive control based system where significantly higher quantities of solar energy are collected, the purchased energy quantities are much lower than in the *ΔT* based system. This desirable outcome is indicative of higher performance of the system. Table 6-7 summarizes the annual purchased energy of the simulated predictive control based system and compares it against the *ΔT* based system. A reduction in purchased energy of 60.8% is achieved through the implementation of this control scheme.

Table 6-7: Annual purchased energy for simulated predictive and *ΔT* controlled systems

	Predictive Controller	***ΔT* Based Controller**
Purchased Energy [MJ]	2739.1	6989.0

6.2.5 Makeup of Stored Energy

The makeup of stored energy can be thought of as the individual component fractions of energy supplied to charge the DHW tank by each of the operating modes that the controller selects. On a monthly basis, the makeup of stored energy is plotted as a stacked column chart in Figure 6-15.

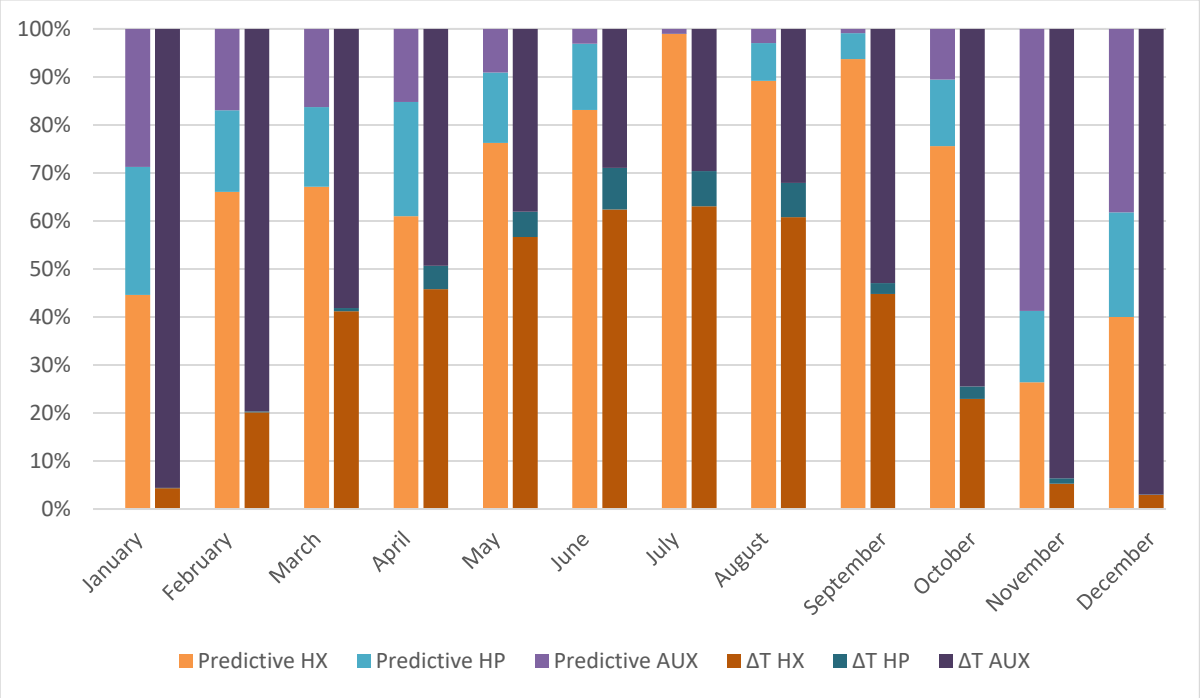


Figure 6-15: Monthly makeup of stored energy for predictive and *ΔT* controlled simulations

In the figure above, the left column for each month represents the energy makeup of the predictive-controlled system. The right column for each month represents the *ΔT* based system. The bottom colour band (orange) in each column represents the fraction of energy supplied to the DHW tank during HX operation. The middle band (blue) represents the fraction of energy supplied to the DHW tank during HP operation. Finally, the top colour band (purple) represents the fraction of energy supplied to the DHW tank by the AUX heating element.

Three trends are seen in the monthly makeup of energy storage. First, the predictive-controlled simulation exhibits a much higher HX and HP use in the shoulder seasons than the ΔT-controlled system. Secondly, the amount of AUX usage is greatly reduced in the predictive-controlled simulation as compared to the *ΔT* system, and thirdly, in the summer months, when there is a greater concentration of solar energy available, nearly no HP or AUX operation is required by the predictive-controlled system, as opposed to the *ΔT* controller, which unnecessarily calls for AUX and HP heat during the months of June, July, and August.

6.2.6 System adequacy and Other Metrics

Similar to Chapter 5, additional metrics are presented here. Table 6-8 lists the number of draws and mass of water drawn below $35°C$ and Table 6-9 lists the number of compressor cycles.

Table 6-8 Annual DHW draws whose temperature fell below the required 35°C threshold

	Predictive Control	***ΔT* Control**
# of under-temperature draws	18	65
Mass of under-temperature draws [kg]	590.8	3302.6
Minimum Draw Temperature [°C]	24.8	13.0

Table 6-9 Compressor cycles and short-cycles for predictive and ΔT controlled simulations

	Predictive Control	***ΔT* Control**
# of Compressor Cycles	590	1374
# of Short Cycles	369	1183

In addition to the two metrics presented above, two other performance indicators are introduced here. The annual number of stagnation events and the monthly and annualized system losses. System *stagnation* was described in Section 6.1.3 as a fault that occurs when the STC experiences no flow during a period of significant insolation. Conditions with a risk of stagnation were identified by post-processing the simulation results to identify periods greater than 15 minutes where the STC was exposed to solar input greater than 720 kJ/h m^2 but had no flow. A count of these occurrences, as well as an accumulated duration of these occurrences are displayed in Table 6-10.

Table 6-10: Number and duration of stagnation for predictive and *ΔT* controlled simulations

	Predictive Control	***ΔT* Control**
# Stagnation Events	16	666
Duration of Stagnation [h]	8.8	1889.4

As seen in Table 6-10, there is a significantly higher risk of stagnation occurring in the system controlled with the temperature differential controller as opposed to the predictive controller. This is attributable to the constraints applied to the predictive controller which directly seek to avoid

stagnation events by reducing the energy stored in the tank leading into a period of low DHW draw demand. On the other hand, the *ΔT* Control system always seeks to achieve the same "charged" state represented by the setpoint temperature at the top of the DHW tank. This often means the application of AUX heat in the overnight or early morning hours, reducing the remaining capacity of the DHW tank to accept solar input when it is available. Under conventional *ΔT* Control, collection modes are often switched off when the tank reaches its temperature setpoint regardless of whether the system is still receiving solar energy.

Lastly, the losses of the system were extracted from simulation results. From a whole-system energy balance with simplification applied to Equation 6-6 from a previous section, monthly system losses are calculated as a percentage monthly of input energy using Equation 6-7 below.

$$\%\ Loss = \frac{E_{Loss}}{E_{Collected,Solar} + E_{Purchased}} \qquad (6-7)$$

These quantities are plotted in Figure 6-16 below and annual quantities are shown in Table 6-11 on the following page. These losses are for the whole system, including all components, not just the DHW storage tank. This includes the STC, HP, HX, etc.

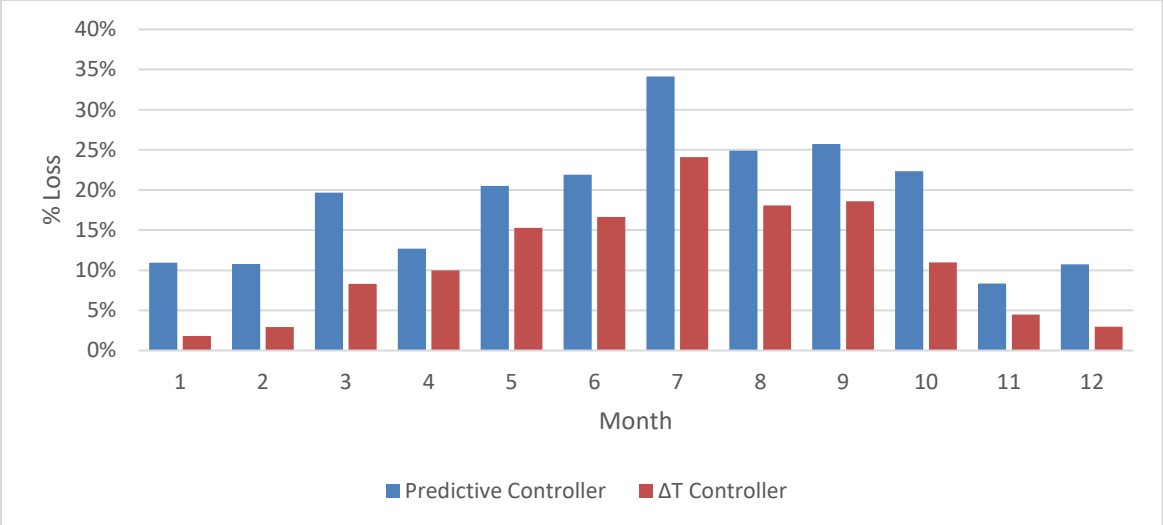


Figure 6-16: Monthly System losses as a percentage of input energy for predictive and *ΔT* controlled systems

Table 6-11: Annual losses as a percentage of input energy for predictive and *ΔT* controlled systems.

	Predictive Control	***ΔT* Control**
% Loss	18.6 %	10.9 %

As seen in Figure 6-16 and Table 6-11, system losses are the first metric that has been discussed by-which the predictive control appears to perform worse than the traditional *ΔT* Control. This warrants an explanation for the increased losses seen in the predictive controller. The first contributing factor to the higher losses seen in the predictive controller are attributed to the temperature profile in the DHW tank. In order to avoid under-temperature draws from time to time, the predictive controller charges the DHW tank to a higher average temperature than the *ΔT* controlled system. Also, there is more stratification in the *ΔT* controlled system than the predictive system because under predictive control, the HP operates more frequently and while the HP is operating, higher flow rates in the DHW tank led to mixing.

The rate of thermal losses from the DHW tank is limited by conduction through the insulated walls of the tank and therefore is approximately proportional to the difference in temperature between the water inside the tank and the air temperature in the surrounding room. Figure 6-17 shows the average tank temperature distribution of the DHW tank from the annual simulations under both control systems. As shown in the figure, the average condition of the DHW tank under the predictive control led to heat *losses* throughout the entire vertical profile of the tank. The average conditions of the *ΔT* controlled system include a temperature lower than the ambient at the bottom of the tank, but higher than the ambient at the top. The result is a profile that leads to small heat *gains* in the bottom portion of the tank, and larger heat *losses* near the top. Integrating the differential heat transfers shown over the entire area of the DHW tanks would reveal larger losses from the DHW tank in the predictive-controlled simulation than the temperature-controlled simulation.

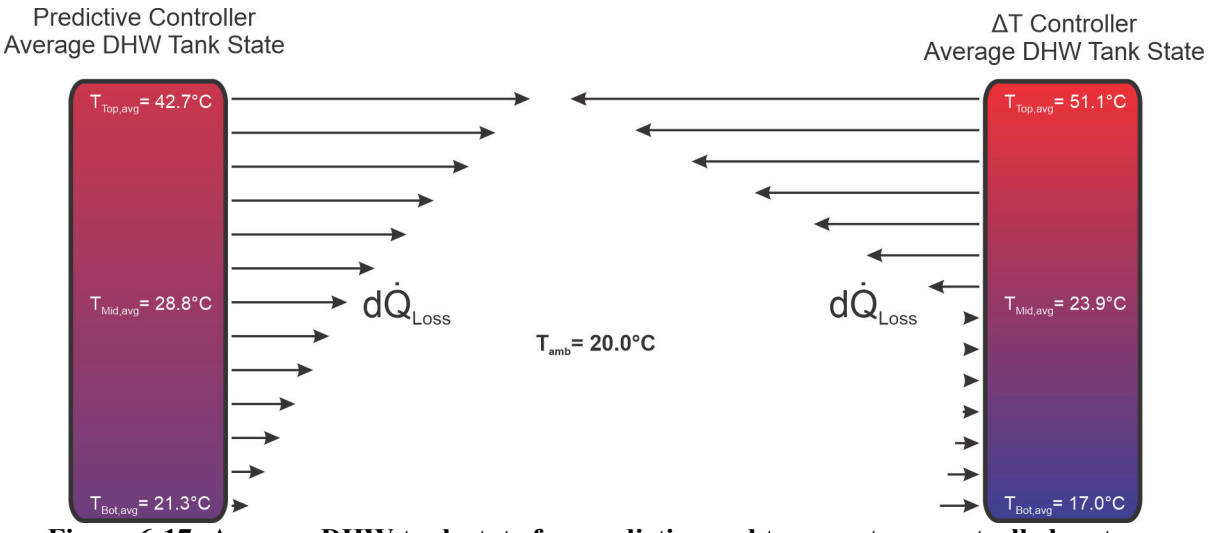


Figure 6-17: Average DHW tank state for predictive and temperature-controlled systems

The second, and often larger contributing factor to the higher losses in the predictive system is due to reverse operation of the HX and STC. The *ΔT* controller, by design, will never activate the HX if the fluid on the source side is lower than the fluid on the load side of the HX, however the predictive system sometimes will. When the predictive controller detects a potential for over-charging the tank late in the day, or when iterative simulations show that stagnation can be avoided by alternating the operating modes between HX, HP and Warmup mode, there are times when heat will be transferred in the reverse direction from the DHW tank to the STC and ultimately rejected to the surroundings.

On the surface, this behavior seems undesirable, since it leads to higher losses, but analyzing the totality of the situation, some energy loss can be considered the price to be paid to avoid overheating and stagnation. Looking back at the previous sections, it is noted that the 60.8% reduction in annual purchased energy for the predictive system significantly outweighs the 7.7% increase in annual energy losses.

6.3 Conclusions and Recommendations

Previous simulations of the SAHP-DHW system of interest have shown modest benefits associated with the addition of a HP to SDHW systems when using traditional temperature-based controllers. The findings in the previous chapter concurred with those of other researchers, that in order to best-showcase the benefits of an SAHP-DHW system, a more advanced controller needed to be utilized.

The application of predictive-control to the system under study allows the system to implement a forward-looking optimal control series that takes into account the predicted system state as a result of the external forcing functions (weather and draws). In this chapter, the author's approach to simulating predictive control of this system was presented. The model incorporated for use in the controller was the same dynamic numerical model of the system that was verified in Chapter 4.

Significant improvements in system performance were seen across a variety of key performance indicators for the system as compared with the same system simulated using temperature-based control. Solar Fraction increased by a factor of 1.95, the annual purchased energy used by the system was reduced by 60.8%, and the makeup of energy source contributions showed a more favorable distribution of HX, HP, and reduced AUX heat input.

System adequacy and other metrics were also improved by the application of predictive control to this system. Under-temperature draws were reduced by 72% to 18 total for the year. The number of compressor short-cycles was reduced by 68.8%, and the number of events where a significant risk of system stagnation was reduced from more than once per day, to 16 times in the year. Independent of energy-based metrics, these improvements suggest that an SAHP-DHW system utilizing predictive control behaves better, would have a longer service life, and would more readily meet the needs of a building owner or occupant.

The metric of system energy losses, however, was negatively impacted by the change in control regime. As explained in the previous section, a combination of lower amounts of stratification, and reverse operation of the HX led to an increase in system losses from an annual 10.9% to 18.6% of energy stored in the DHW tank. The increased solar energy collected by the system does outweigh the increased losses, but significant losses from the DHW tank would need to be accounted for in other building energy calculations, for example, if the tank exists within a conditioned space.

Some other limitations to these results must be recognized as a result of the implementation of the control architecture to a simulation study. The work in this chapter simulated an essentially perfect scenario for a controller. The model used for the iterative steps to develop the control series was practically the same as the main simulation model that was implementing the control series. This means that in this chapter, the main simulation output always matched with the predictions of the iterative sub-simulations. Similarly, the forcing functions (weather and draw profile) used in the iterative sub-simulations for this study matched the inputs to the main simulation environment. This

means that the controller had the benefit of perfect foresight with respect to upcoming weather changes and draw events. In a more realistic application of predictive control, the weather forecast data used by the iterative sub-simulations would come from a forecasting agency, rather than a TMY input file, so it would be subject to some degree of error. The draw prediction information used by the sub-simulations could be derived statistically from a large set of real home draw data, or a live model could be developed to make predictions about upcoming DHW draw loads based on historical use, current occupancy data, etc.

With a controller benefiting from perfect foresight engaged in an iterative process to optimize the control series, the performance of the system presented in this chapter can be thought of as a best-case-scenario or an approximate upper limit on the performance of a SAHP-DHW system as configured. The fact that such a dramatic improvement has been shown in this study is quite encouraging, but must be tempered by the implication that once real-world error in system model predictions, and forcing function/load predictions manifest, the true performance benefit will be diminished. The following chapter will partially address the limitations mentioned above by implementing the controller developed in this study to a physical experiment in the lab rather than a solely simulation-based study.

Chapter 7
Experimental Operation of Improved Controller

Finally, in this chapter, a series of daily experimental trials are shown using a version of the predictive controller described in Chapter 6 ported to an instance of LabVIEW running on the ETU in real time. The system controller uses iterative sub-simulations to investigate potential operating paths and selects the best from the set using the same objective and constraints as before.

Time and resource constraints limited this work to be performed in the lab, rather than in-situ in an occupied residence, so the material presented in this chapter can be thought of as short time-scale HIL simulations.

7.4 Method

As with previous experimentation using the ETU, the flow paths from Section 3.3 were used to incorporate the HX, HP, DHW Storage Tank, and the STC. Functionality and instrumentation of the ETU were used as described in previous chapters to simulate a solar input, control circulation pumps, draw flow rates and HP compressor speed when required.

The simulation architecture presented before in Section 6.1.2 is modified for the current study so that instead of a top-level TRNSYS simulation standing in for the physical equipment, the instrumented state of the ETU is used directly. Secondly, the predictive controller that was formerly called by TRNSYS is now called by LabVIEW, and the necessary modifications to that script were made so that it is compatible with the data structures used by the ETU. Figure 7-1 on the following page shows an updated architecture for the HIL simulations conducted using the ETU.

Initial conditions for each trial were as described in Section 4.6. The DHW tank begins in a stratified condition that is experimentally obtained by first charging the tank to a mixed 50°C, then drawing off 150L of water from the top of the tank and replacing it with mains-temperature water at the bottom. Inputs to the system are again as described previously, using extracts from the TMY weather file for Ottawa, ON, and Edwards House 38 draw schedule. Each trial presented in the next section represents operation over the daylight hours of the day in question. The relevant inputs are plotted adjacent to their associated experimental results.

MPC Experimental Control Architecture:

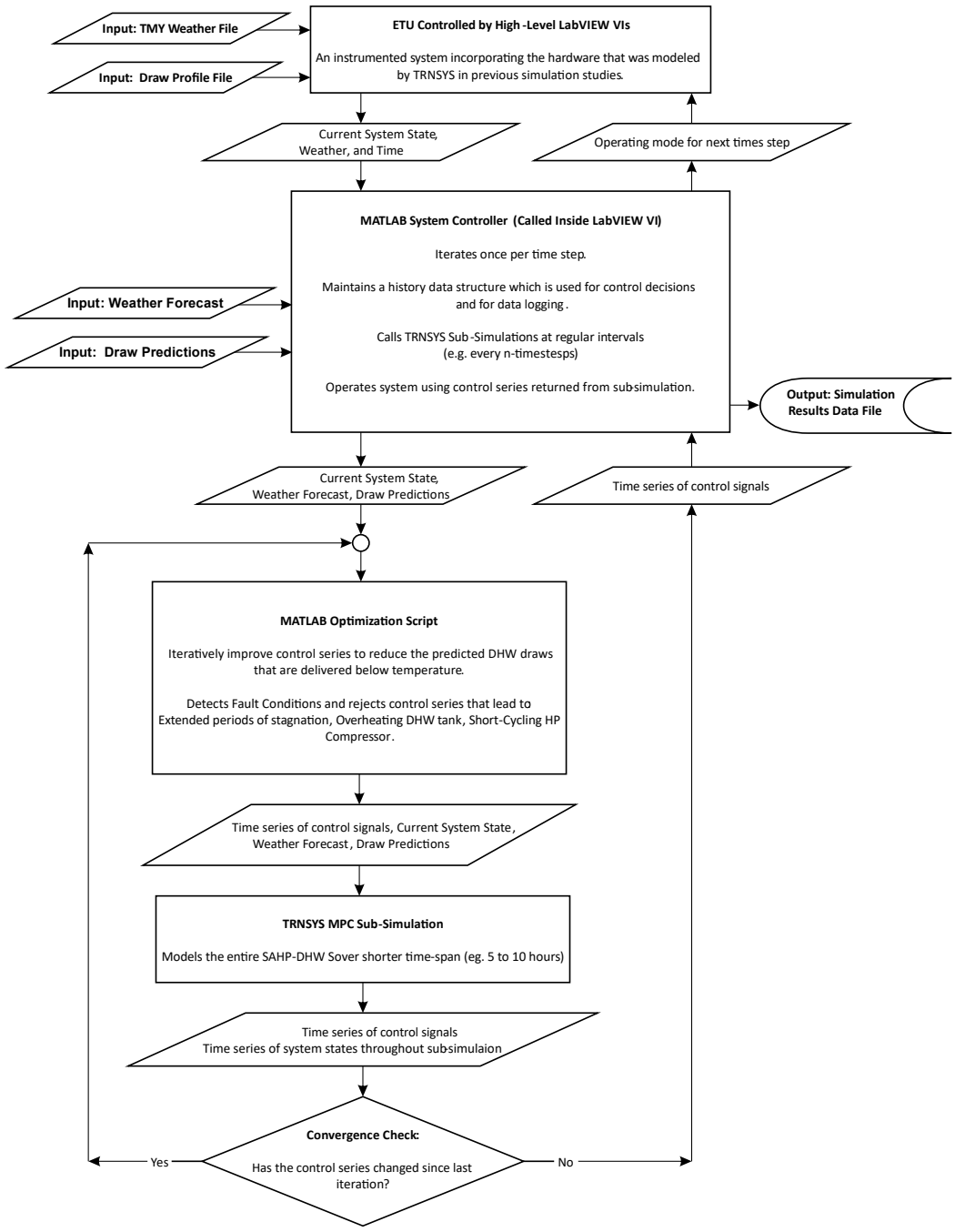


Figure 7-1: Experimental HIL Control Structure for Predictive Control of ETU

7.5 Results and Analysis

The results presented below are from a range of dates throughout the year selected to provide a representative sample of the system's operation under various solar input and DHW draw load conditions. Trials run for April 16th, May 29th, June 5th, and lastly September 7th based on the TMY for Ottawa and Edwards House 38 draws are each presented in their own sub-section below.

7.5.1 April 16th

The model input weather conditions for April 16th are show in Figure 7-2. For this daily trial, the HIL simulation was run from 2524h (4:00 AM) to 2539h (7:00 PM). In the plot of weather inputs, note that in previous sections the indoor air temperature was a constant 20°C in previous simulation-based studies, but in the trials shown in this chapter, the measured room temperature in the lab, approximately 22°C is shown instead.

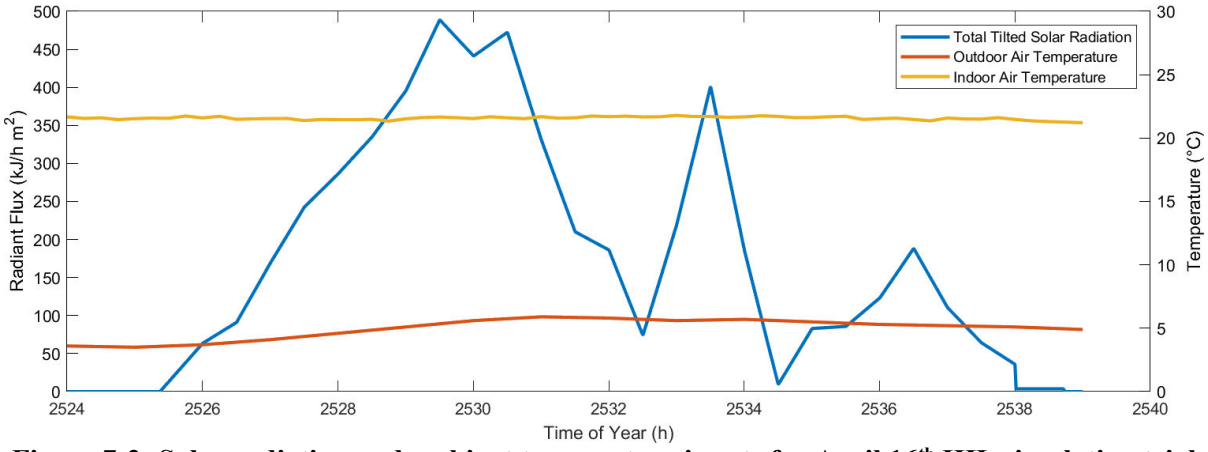


Figure 7-2: Solar radiation and ambient temperature inputs for April 16th HIL simulation trial

Noting the scale of the primary y-axis in the plot above ranging from 0 to only 500 kJ/h m^2 (approximately 139 W/m^2) this day has categorically low insolation owing to overcast conditions. In fact, relative to the sunnier days presented in the next subsections, April 16th has less than 20% of the irradiation over the course of the day and one of the lowest ambient air temperatures.

Figure 7-3 on the following page shows the draw schedule from Edwards house 38 used for April 16th and the predicted temperature of the cold mains water inlet from the TMY weather file. This day is characterized by approximately 294 L of DHW consumption over the course of the day, which is 1.16 standard deviations above the average daily consumption for this house of 176.64 L.

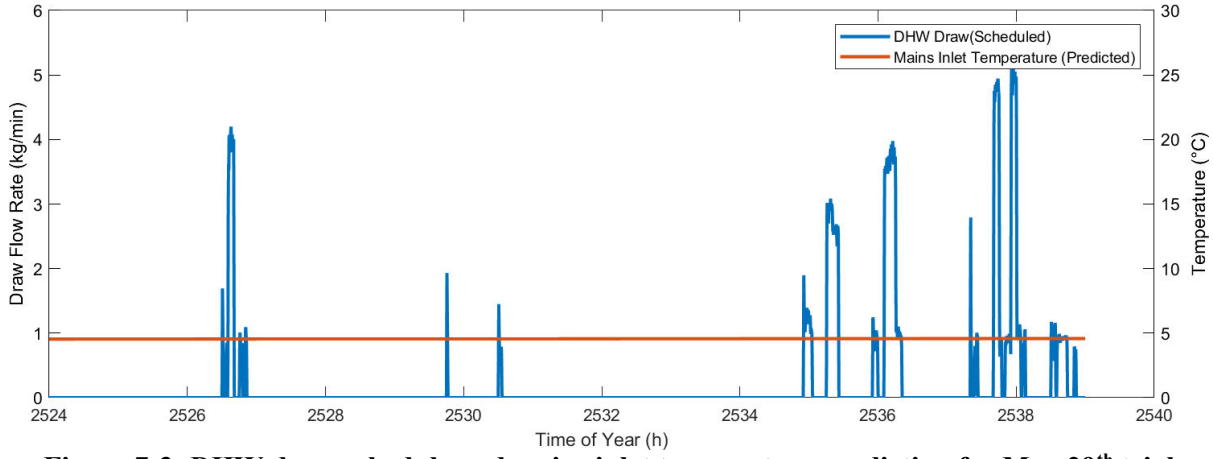


Figure 7-3: DHW draw schedule and mains inlet temperature prediction for May 29th trial

Intuitively, with extremely low solar radiation and above-average DHW consumption, this day would not be a good candidate for solar-thermal collection. This bears out in the system operation observed for this day. The system state, represented by the average temperature of the DHW tank over the course of the daily trial is shown below in Figure 7-4. As with the iterative process visualization plots in Section 6.2.1, dashed lines in the plot below represent alternative operation paths predicted through the iterative sub-simulations. The solid blue line represents the observed condition of the ETU's DHW tank throughout the trial. The magenta highlighted trace in the plot below represents the controller simulating and predicting that the HP would not be able to run without short-cycling. Therefore, the HP is disabled for the duration of the day. The yellow highlights later in the day represent draws whose temperature are predicted to fall below the acceptable temperature threshold.

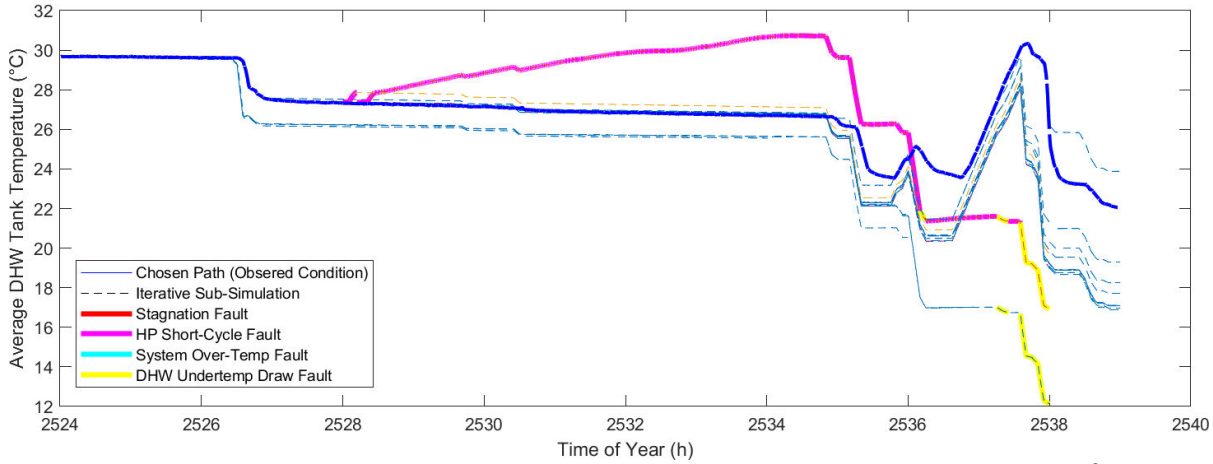


Figure 7-4: Average DHW tank temperature from HIL simulation of April 16th

For clarity, unlike the visualization of the multiple candidate controls displayed in Figure 6-11 in the previous chapter, the following plot in Figure 7-5 is of the chosen control path only. The solar operating mode (HX, HP, and Warmup) is plotted on the top axes while the independent operation of the AUX heating elements is plotted on the lower axes.

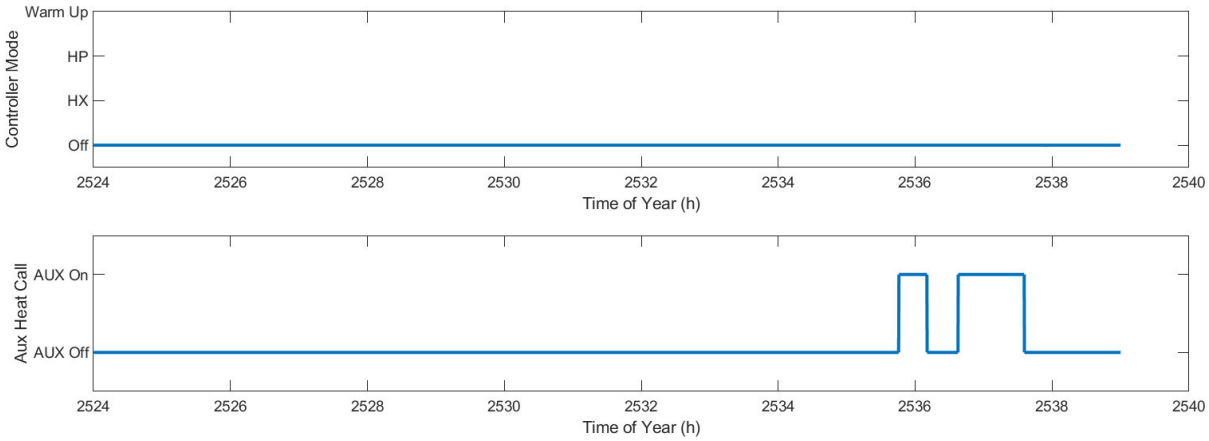


Figure 7-5: Chosen control series operated during HIL simulation of April 16th

As seen in the plots above, the system operates on this day without activating the solar modes at all. According to the sub-simulations being conducted, the DHW tank began the day with enough hot water to satisfy DHW draws up until 2536h, where a draw was predicted that would produce a below-threshold temperature, and a second pair of draws near 2537h would do similar. The response of the controller on this poor-solar day is to wait until just-before the draws, and then schedule AUX heat to charge the top of the tank with just enough energy to meet the load at 2535.75h and 2563.6h.

As a means of assessing whether the controller adequately did its job, two additional plots are provided below, an area plot of energy availability in Figure 7-6 and a plot of DHW draw temperature versus time in Figure 7-7.

Recall while examining the plot that, as defined in Section 6.1.4, "Available" energy is calculated as an integral of the energy content of the volume of water held in the DHW tank above 35°C, while the "Unavailable" energy is calculated as an integral of the remaining volume of water held in the tank with reference to 0°C.

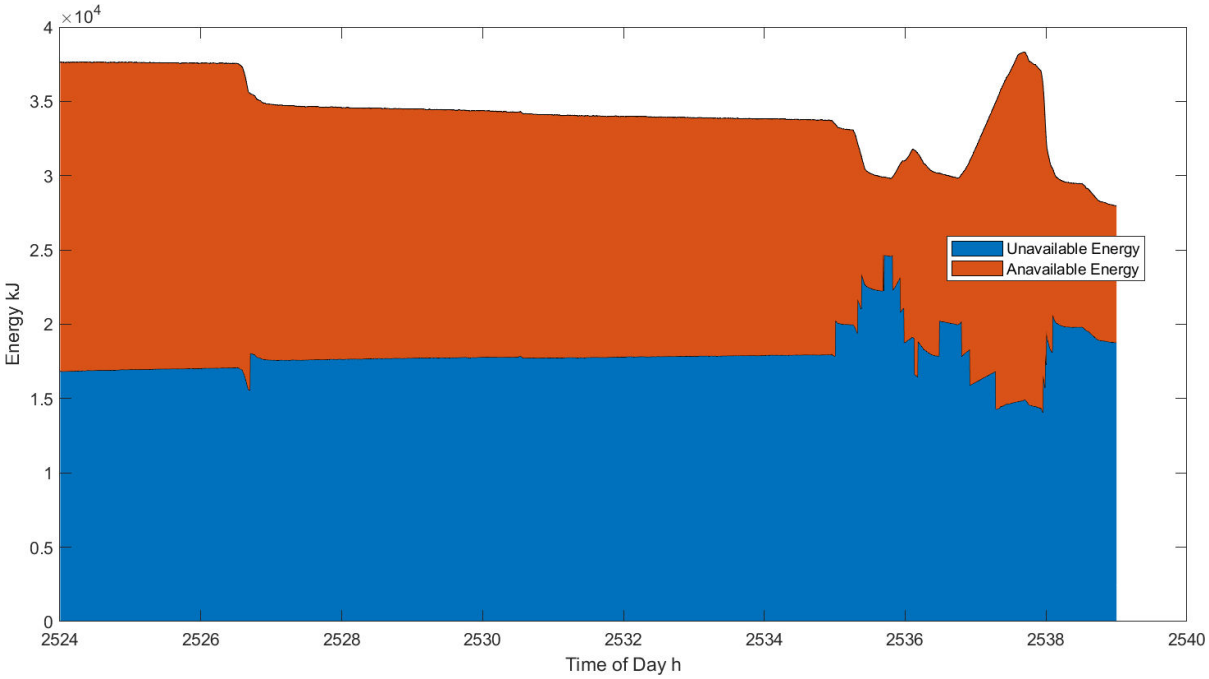


Figure 7-6: Plot of energy availability during HIL simulation of April 16th

In the plot above, the effect of each draw can be seen on the available energy in the tank. Each time DHW is removed from the top of the tank and cold main water is introduced at the bottom, both the available energy and the unavailable energy decrease. Because there are no instances of solar operation on this day, there is little mixing to comment on. When the auxiliary heaters are activated in the afternoon, we see two peaks where available energy increases. The first at 2536h coincides with a draw, so it is not as pronounced, and the second at 2537.5h precedes the largest draw of the day, and briefly increases the available energy in the DHW tank to above the level at the beginning of the day.

It is noted that the available energy reaches local minima after draw events, but never actually approaches or reaches zero. This indicates that the system maintains a buffer of available energy ready to meet draw loads. In a real-world system, some reserve capacity would be desirable to meet unexpected loads that could occur for instance, if a draw event happened earlier than predicted, or if the mains inlet temperature was lower than predicted. However, in most of the cases presented in this chapter, the cold mains inlet temperature is in fact higher than predicted by the TMY file, so the sub-simulations used for control purposes appears to keep an even larger buffer of energy in the tank. This is further discussed in the next section, where the temperature residual is calculated and plotted.

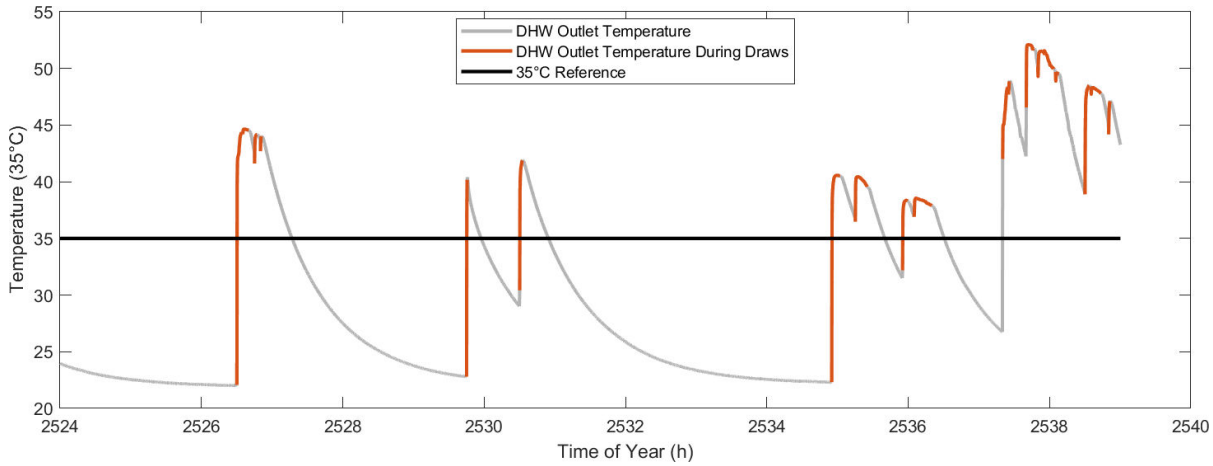


Figure 7-7: DHW draw temperature vs. time during HIL simulation of April 16th

Analyzing the plot of DHW temperature vs time allows us to assess whether the system would have met user demands over the course of the day. It is noted that during periods with no DHW draw flow rate, the temperature at the outlet of the DHW mixing valve relaxes to near room temperature. This behavior is plotted in grey for completeness, but should be ignored when considering the performance of the system since the temperature of water delivery is only relevant when there is flow. It can be seen in the plot above that neglecting a short 15s sample at the beginning of each draw, for all of the draws throughout the day the system reliably delivered DHW above the required temperature threshold. It can therefore be concluded that on a poor solar day, the system correctly schedules backup AUX Heat in time to meet DHW loads. This is consistent with the reduction in under-temperature draws reported in Section 6.2.6.

7.5.2 May 29th

The May 29th trial was run from the hours 3556.25 to 3571.6 of the year, which represents times of day from 4:15 AM to 7:36PM. This time range covers the daylight hours of the day in question with a buffer beforehand for the system to start-up and react to forecast draws. The model input weather and draw conditions for May 29th are show in Figure 7-8 and Figure 7-9.

Observing the solar radiation in the plot below, the smooth curve and peak of 3420 kJ/m^2 at 3564h (12:00 PM) indicates a day with very little occlusion of the sun by clouds, except for the small notch just after noon. The ambient outdoor temperature swings from 7.4°C, which could drive some losses from the STC during early morning operation to a peak of 23.9°C in the afternoon. This day would be considered a strong candidate for solar collection.

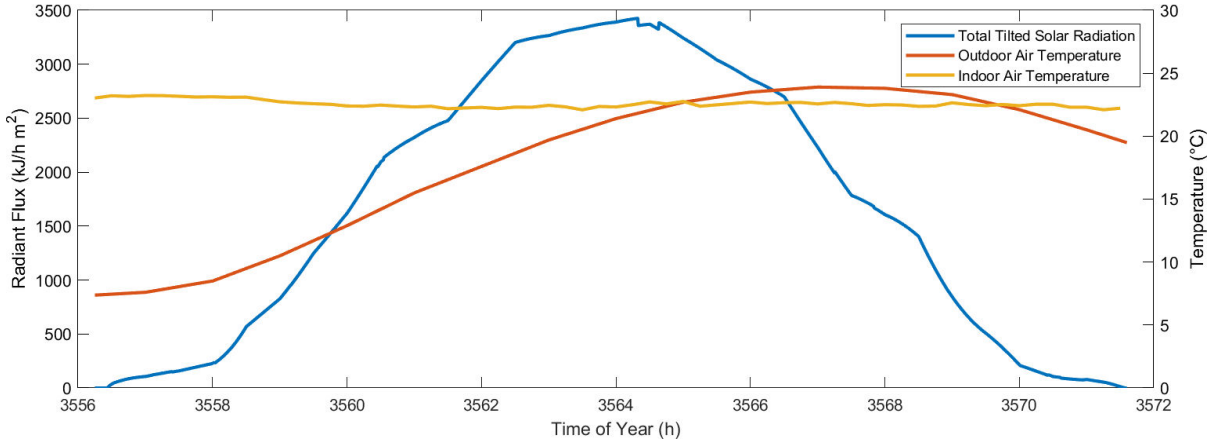


Figure 7-8: Solar radiation and ambient temperature inputs for May 29th HIL simulation trial

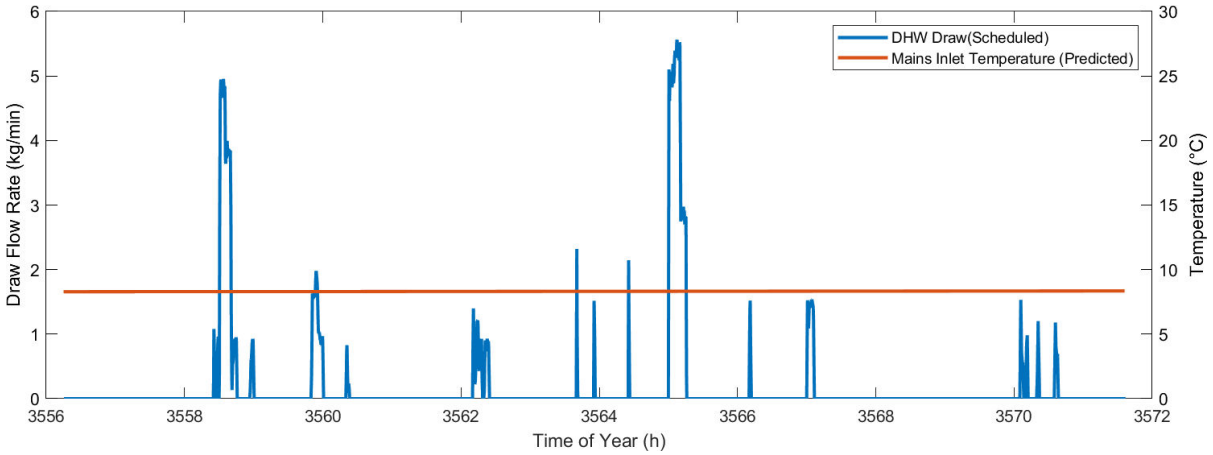


Figure 7-9: DHW draw schedule and mains inlet temperature prediction for May 29th trial

As with the previous trial, the observed and alternative operating state paths for the DHW tank are represented in Figure 7-10 on the following page. The only fault conditions detected are potential stagnation conditions which lie on alternative paths towards the end of the day. These potential stagnation conditions show that the controller has achieved a sufficient charge condition to cover forecasted energy demands and then simulated to find out whether switching off would be appropriate. The faults were avoided by the controller by operating in HX mode right up to the end of the trial, potentially triggering a small reverse HX operation as described in Section 6.2.6, but as seen in the figure being considered, there is very little change in the average tank temperature after the small draw at 3567 h (3:00 PM). Following the plot of DHW average temperature is the chosen control path that was operated throughout the day. This can be seen in Figure 7-11.

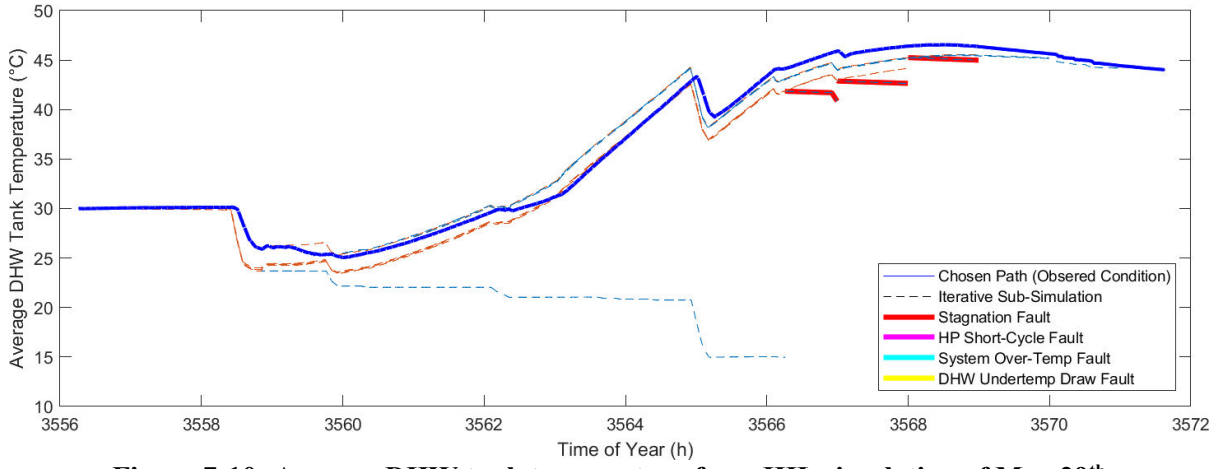


Figure 7-10: Average DHW tank temperature from HIL simulation of May 29th

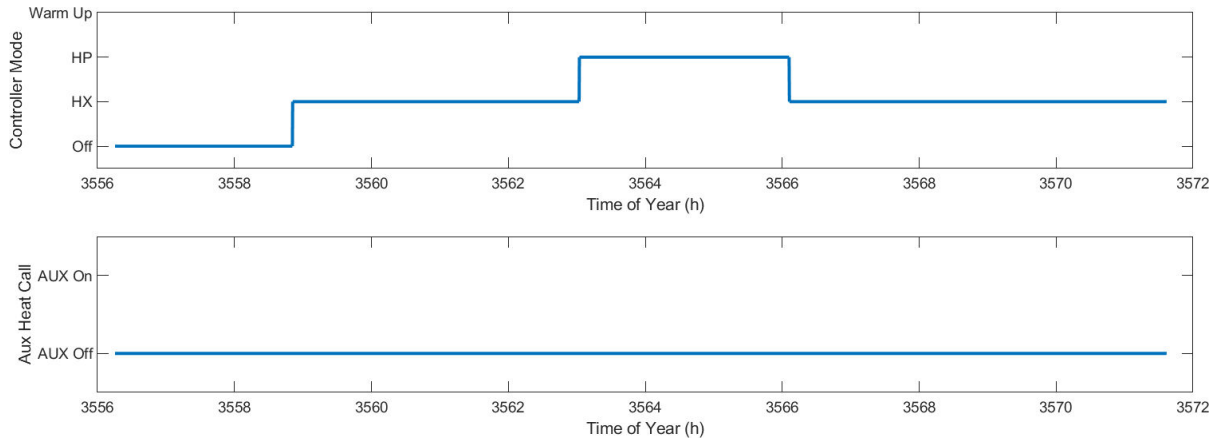


Figure 7-11: Chosen control series operated during HIL simulation of May 29th

In Figure 7-11, we see that the system cycles from off at the beginning of the day to HX mode at approximately 3559h (7:00 AM), then to HP mode at 3563h (11:00 AM) and finally back to HX mode for the remainder of the day at 3566h (2:00 PM). From the second subplot, we can see that on the day in question, there was no AUX heat usage.

The most significant deviation between the observed state of the system and the predicted operating path (between the solid blue line and the dashed lines in Figure 7-10) coincides with large DHW draws occurring throughout the day, for example, at 3558.5h and 3565h. The sub-simulations predict a larger reduction in tank temperature than what was seen in the experimental results. This is attributed to the TRNSYS sub-simulations using the mains temperature from the TMY file for Ottawa as the predicted cold inlet temperature for the DHW tank, while the ETU received much warmer

mains water at its inlet. Figure 7-12 shows the discrepancy between the model-input predicted mains temperature and the temperature of ETU mains inlet water observed during the trial.

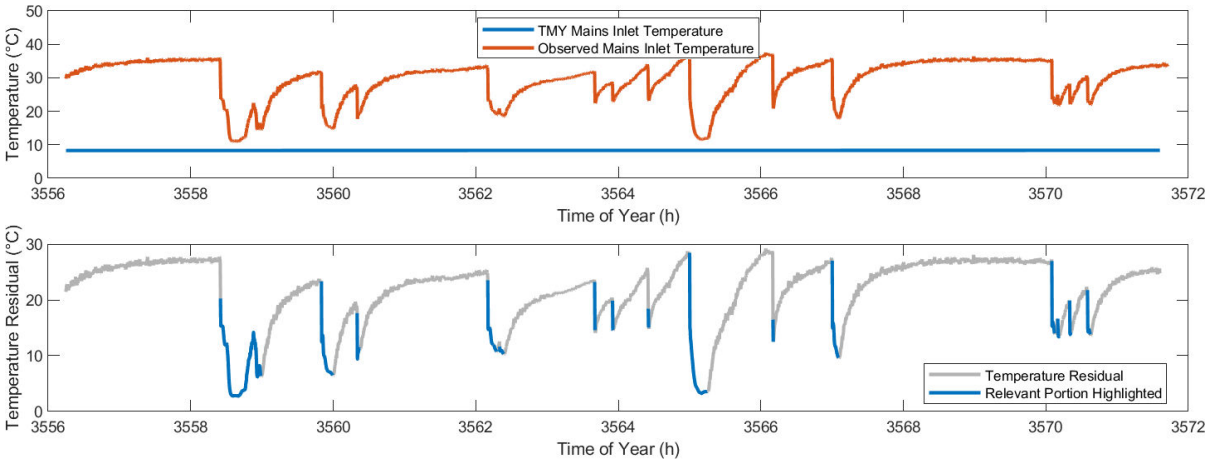


Figure 7-12: Predicted and observed mains temperature vs time and residual from May 29th HIL simulation trial

Similar to the trial presented in the previous section, the system's ability to meet demand is assessed by inspecting the available energy and draw temperature plots in Figure 7-13 and Figure 7-14 shown together on the following page. It can be seen from the plots that after a draw event at 3562.2h, the available DHW is completely depleted, resulting in a period from 3562.7h to 3563.6h where the entire tank is below the threshold temperature and registers as unavailable energy. In the plots on the following page, vertical lines have been added to bound this region for the purpose of illustrating its separation from the next draw of the day which occurs just after this period at 3563.7 h.

According to the plot of available energy, there should be DHW available to meet the draw at 3563.7 h, however according to the plot of DHW draw temperature vs. time, this draw was not met with adequate DHW. It should be noted, however, that this draw event was very short, lasting only 36-seconds, and flowing at a rate of 2.3 L/min, representing only 1.38 L of water. As seen in all of the draws in the previous trial, and in the other draws in the current trial, the draw temperature starts near room temperature and rapidly increases up to the draw temperature where it remains until the draw terminates. This start-up transient in the draw temperature is attributable to the plumbing configuration after the water leaves the DHW storage tank through the hot outlet port. An additional plot is included, Figure 7-15, which shows the temperature of water flowing just past the hot outlet port of the DHW tank, and it can be seen that this flow just barely reaches the desired 35°C.

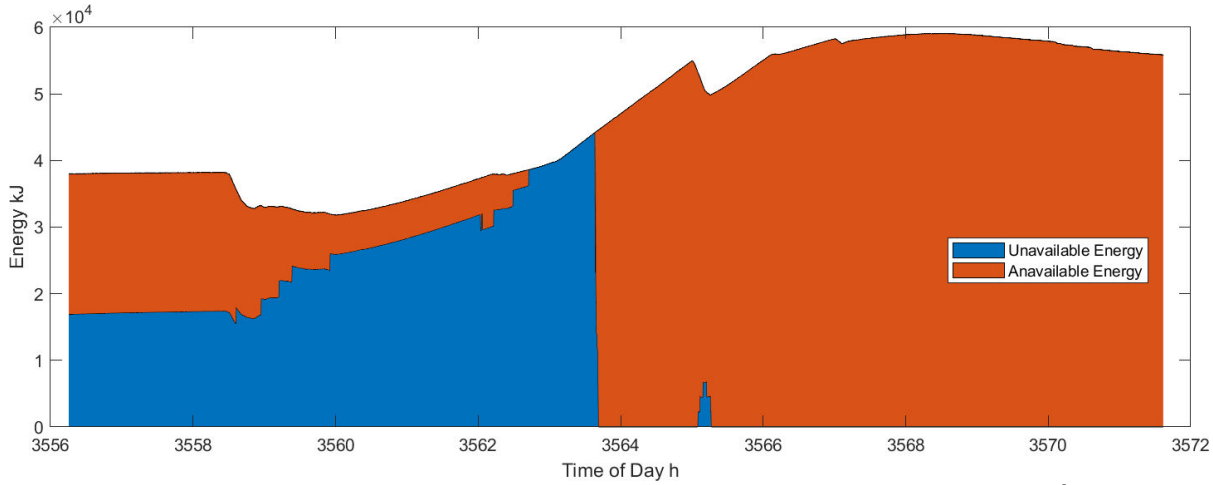


Figure 7-13: Plot of energy availability during HIL simulation of May 29th

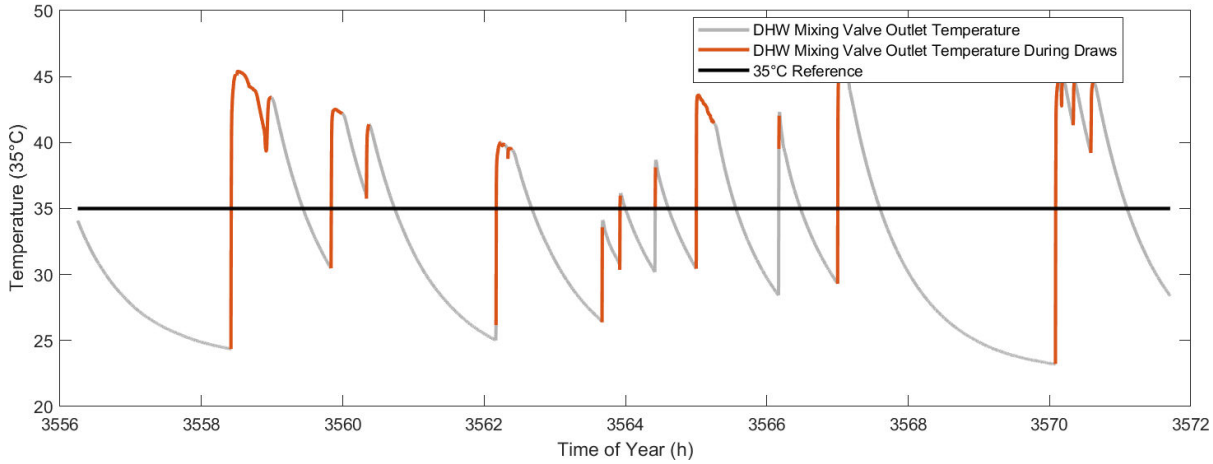


Figure 7-14: DHW draw delivery temperature vs. time during HIL simulation of May 29th

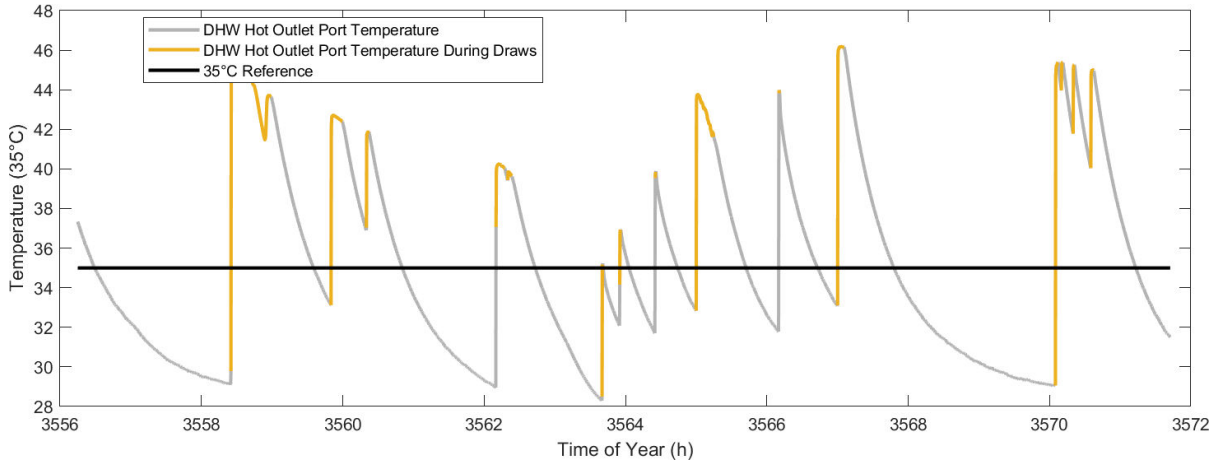


Figure 7-15: DHW Hot Outlet Port temperature vs. time during HIL simulation of May 29th

The behavior discussed and shown in the plots above indicates that the controller has performed as it was designed to do, but that the constraints applied to the controller did not account for the plumbing configuration after the DHW leaves the storage tank. In particular, the thermostatic mixing valve that is included between the tank and the DHW outlet is modeled as a perfect mixing device, but it could very well have some amount of bypass leakage allowing cold mains water to mix with the DHW stream even when the hot outlet from the tank is below the setpoint of 55°C.

An additional factor that is highlighted in the current trial is that of timing. When this controller was applied in numerical simulations, there was never a time delay or error in control timing. In the current application, between the sample rate of the DAQ, and the opening/closing time of mechanical valves, there can easily be an accumulated error between when a control change is commanded and when it actually takes effect. It would be prudent to account for this error in a future version of the sub-simulation model, either by programming in time-delays/advances, or by coding in an energy buffer to account for hysteresis and time offsets. It would be an improvement if the controller's objective function was to minimize the draw energy deficit subject to a constraint of marginally higher temperature water than desired and/or with a time offset between when the energy becomes available and when it is predicted to be needed.

7.5.3 June 5th

For the June 5th HIL simulation trial, the ETU was operated between 3724.5 h (4:30 AM) and 3739.75 h (7:45 PM) to capture the daylight hours and important draw events throughout the day. Figure 7-16 shows weather inputs for the day, while Figure 7-17 shows the draw schedule.

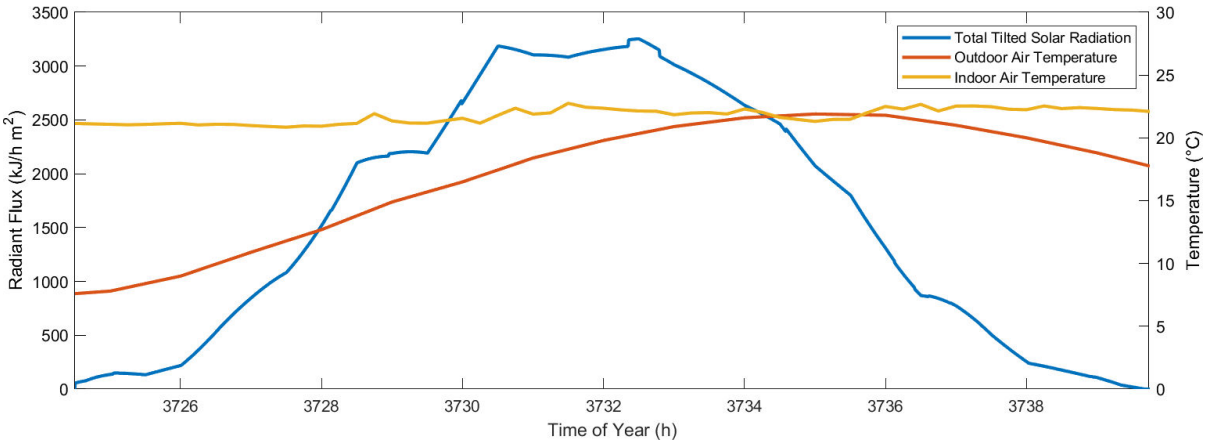


Figure 7-16: Solar radiation and ambient temperature inputs for June 5th HIL simulation trial

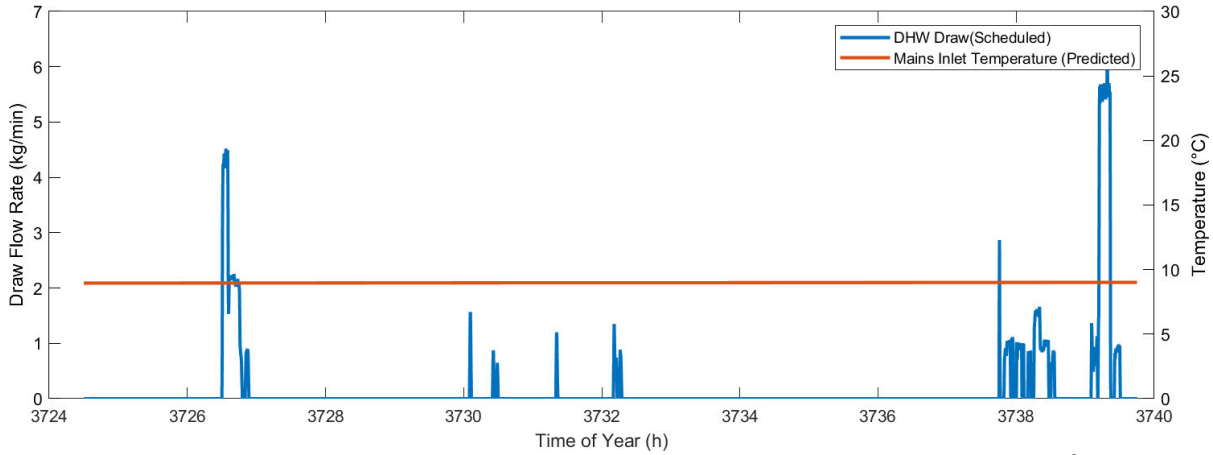


Figure 7-17: DHW draw schedule and mains inlet temperature prediction for June 5th trial

From the weather profile, we can discern that the day is partially cloudy, with some discontinuities in the irradiation. There is still a large amount of solar energy available for collection this day, and a solar-thermal system would be expected to collect for the majority of the day as long as there is capacity to store the energy. The draw schedule calls for 153 L of DHW use throughout the day, which is slightly below average for the data set. There is no reason to believe that the system would have trouble meeting this load with the use of HX alone.

The DHW tank state represented by average tank temperature is plotted in Figure 7-18 on the following page again showing the multiple possible operating paths as dashed lines and the observed condition as a result of the chosen operating path in solid blue. Figure 7-19 shows the chosen control series for the day. As seen in the plots on the following page, there is a draw in the morning beginning at 3726.4 h and terminating just as the solar collection starts with the HX at 3726.75 h. As discussed in the previous subsection, there is an over-prediction of energy removal by the draw as the actual mains temperature in the lab is higher than the TMY file predicts.

The controller operates in HX mode for the majority of the day from 3726.75h to 3736.9h. at which point it would switch the system off, but because this time is still before the sun has fully set, the system instead cycles between off, and “Warmup” mode to change-over the fluid in the STC and avoid stagnation.

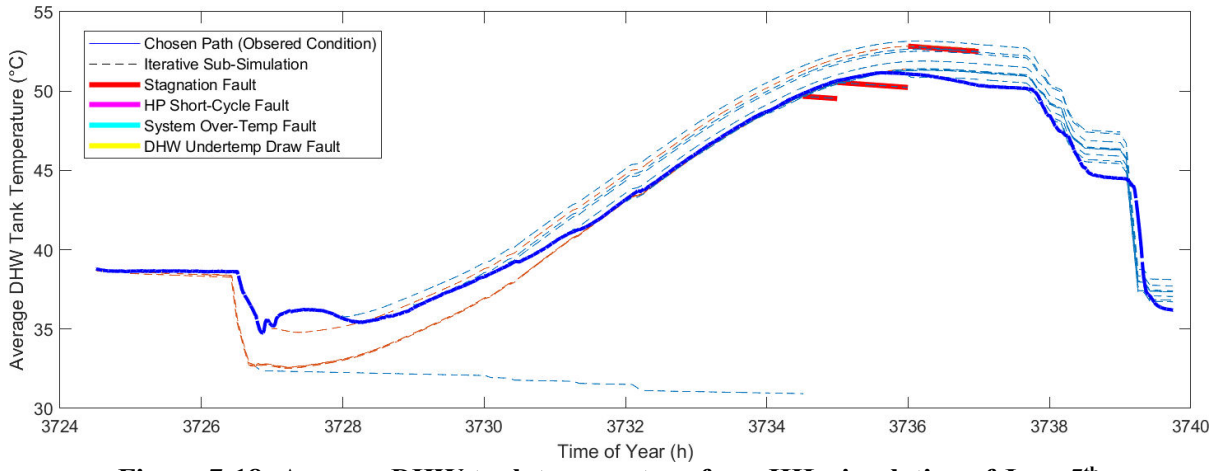


Figure 7-18: Average DHW tank temperature from HIL simulation of June 5th

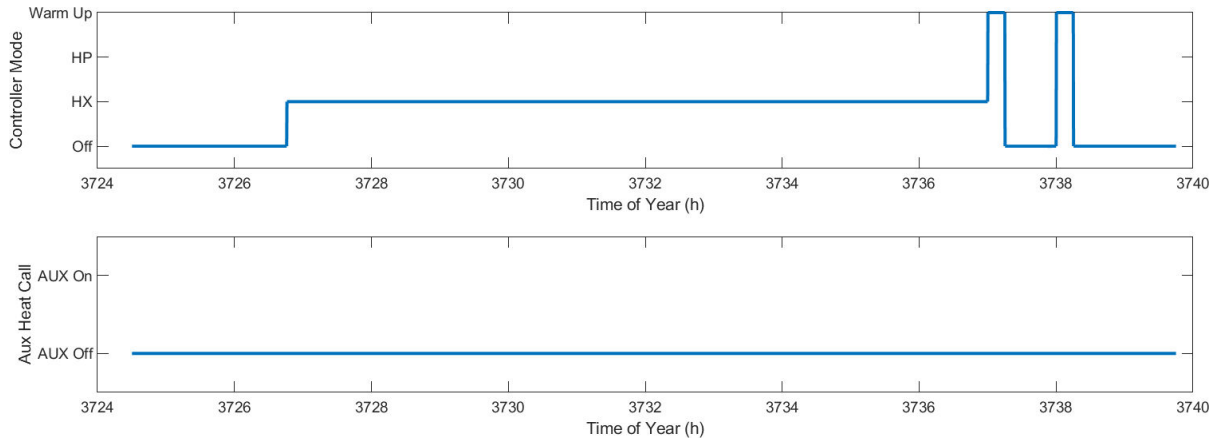


Figure 7-19: Chosen control series operated during HIL simulation of June 5th

One anomaly observed in Figure 7-18 is the positive deviation between the observed state path traced in blue and the prediction of the model traced as an orange dashed line between 3726.75h and 3728h, coinciding with the start-up of solar collection. The magnitude of this deviation is approximately 1.4°C and it lasts for around 1 hour. During this time, solar collection starts up using the HX, which had previously (outside of the data recording period) been used to pre-condition the DHW tank using the procedure described in Section 4.6. During the pre-conditioning process, the entire solar-side loop would have contained 50°C water slowly cooling down towards the indoor room temperature. At the moment when solar charging is activated in this trial, a plug of approximately 30°C water was circulated in the solar loop, which is significantly hotter than the 10°C outdoor ambient temperature read from the TMY file. The result in TRNSYS would be heat loss from the STC and a reduced rate of heat addition to the DHW tank. Unfortunately, a restriction of the ETU is its inability to physically

produce a negative net heat transfer rate from the STC. As a result, the DHW tank received more heat during the start-up transient of the HX than the model predicts and this caused the deviation being discussed. The controller compensates for this by re-simulating at regular intervals, so the predicted and observed traces on the plot converge after approximately an hour and stay coincident for the remainder of the trial.

Plots of the DHW available energy and DHW draw temperature in Figure 7-20 and Figure 7-21 show few interesting details but serve to confirm once again that the system meets the load requirements on this day. In addition, it can be seen by the large solid orange portion of the available energy area plot that the tank has a significant amount of reserve energy on this day. This is attributed to the controller staying in HX mode to avoid stagnation in the early afternoon, even though the DHW tank already has sufficient charge to meet draw demands later in the day.

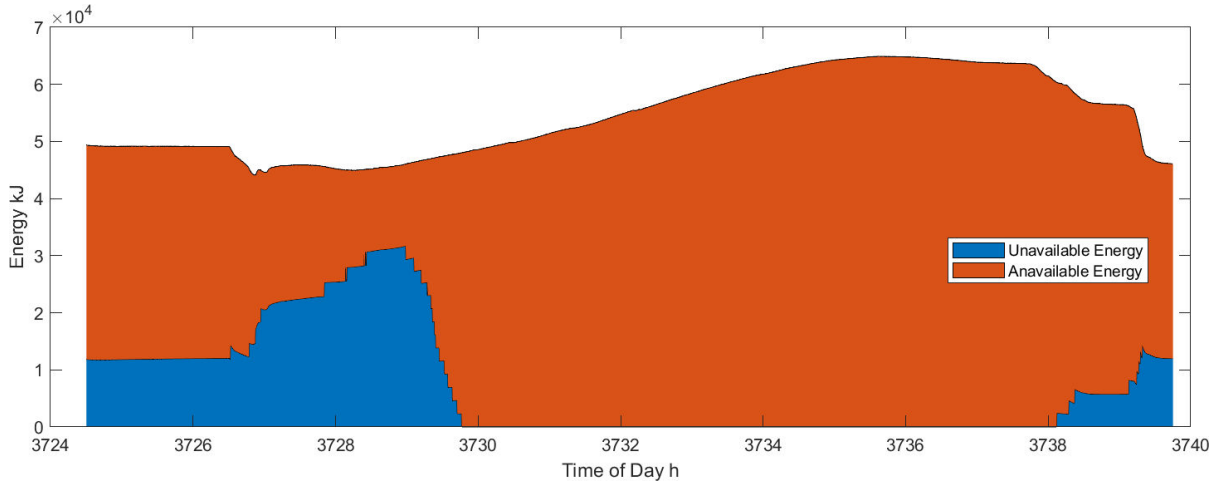


Figure 7-20: Plot of energy availability during HIL simulation of June 5th

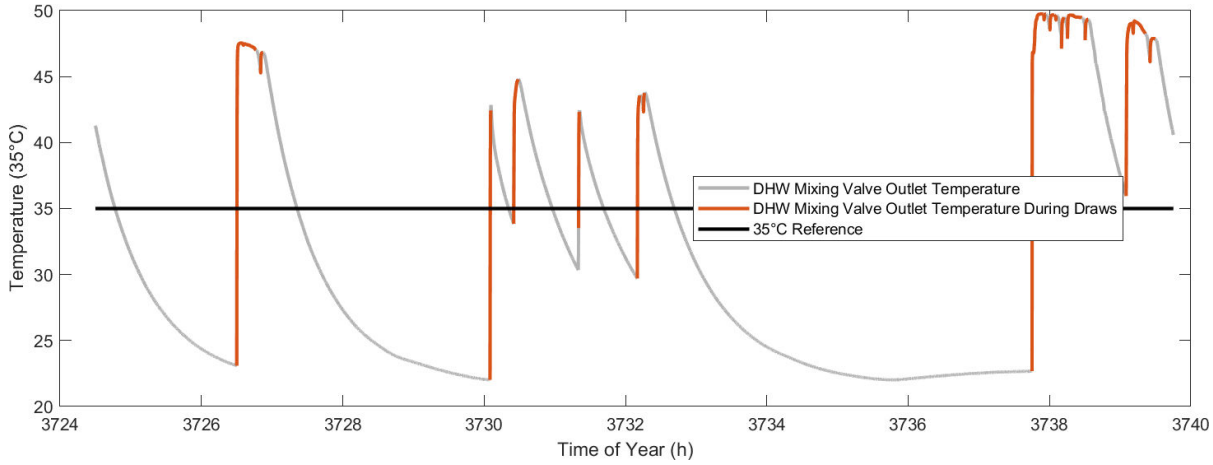


Figure 7-21: DHW draw temperature vs. time during HIL simulation of June 5

7.5.4 September 7th

A trial using weather and draw data for September 7th was conducted with the ETU operating between 5982h (6:00AM) and 5994h (6:00 PM). Weather and draw inputs are plotted below in Figure 7-22 and Figure 7-23. Notable in the solar radiation plotted below is that for a significant period of the day, sunlight is occluded by clouds, reducing the radiation incident on the panel in the morning by at least half. This day is not expected to be good for solar-thermal collection.

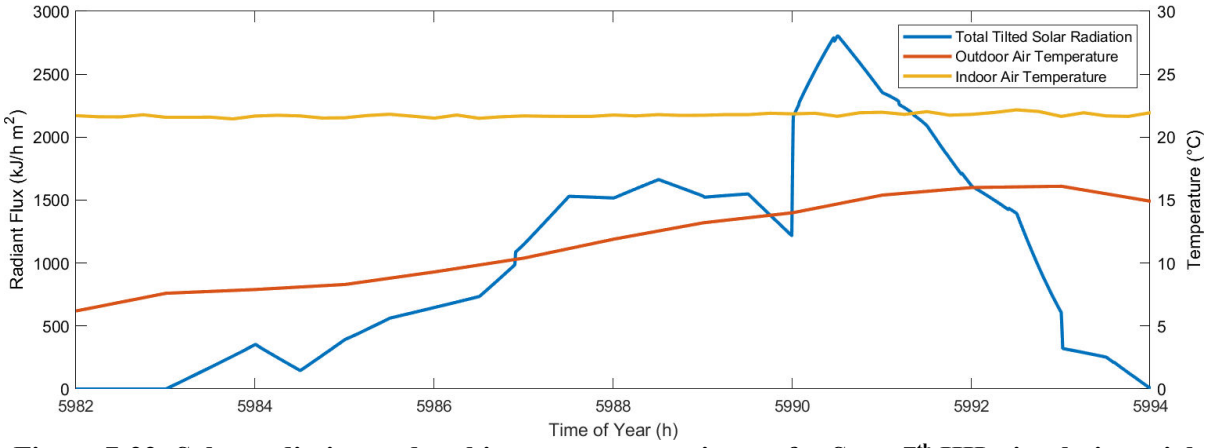


Figure 7-22: Solar radiation and ambient temperature inputs for Sept. 7th HIL simulation trial

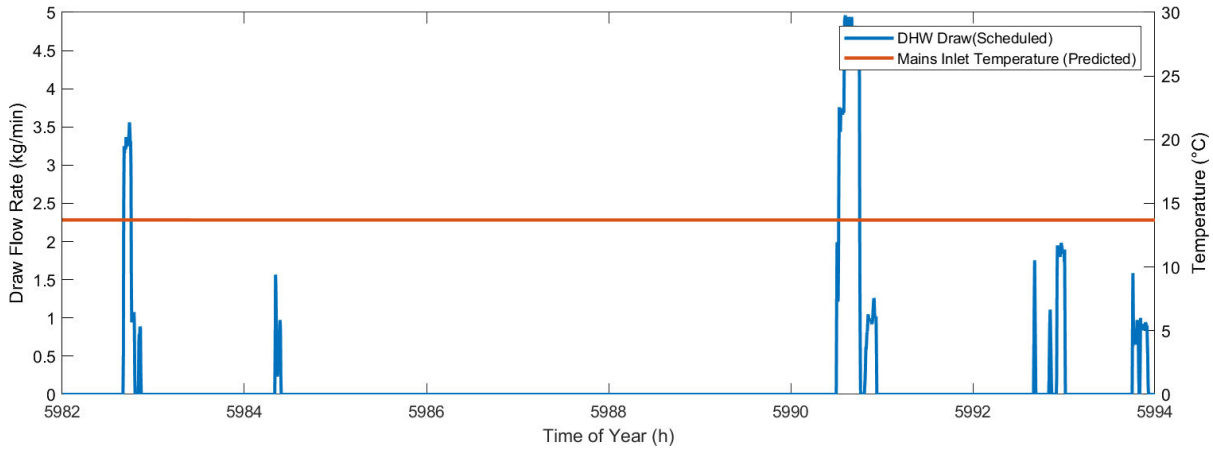


Figure 7-23: DHW draw schedule and mains inlet temperature prediction for Sept. 7th trial

The draws scheduled for Sept. 7th total 119.25 L which comes in below average for the dataset, but there is a particularly long draw in the middle of the day that, as will be shown in the following plots is a challenge for the system to satisfy. Plots of average tank temperature and operated control mode are included in Figure 7-24 and Figure 7-25.

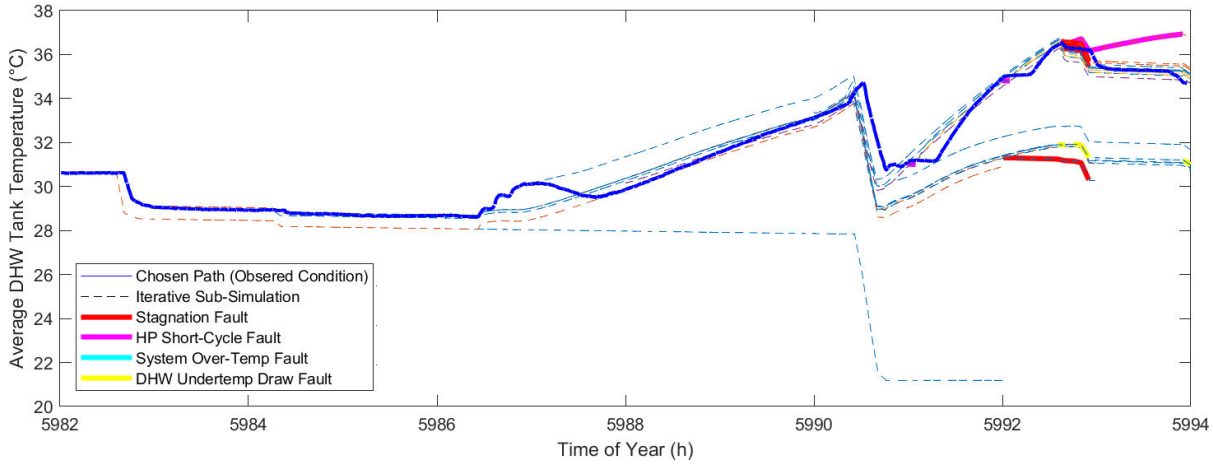


Figure 7-24: Average DHW tank temperature from HIL simulation of Sept. 7th

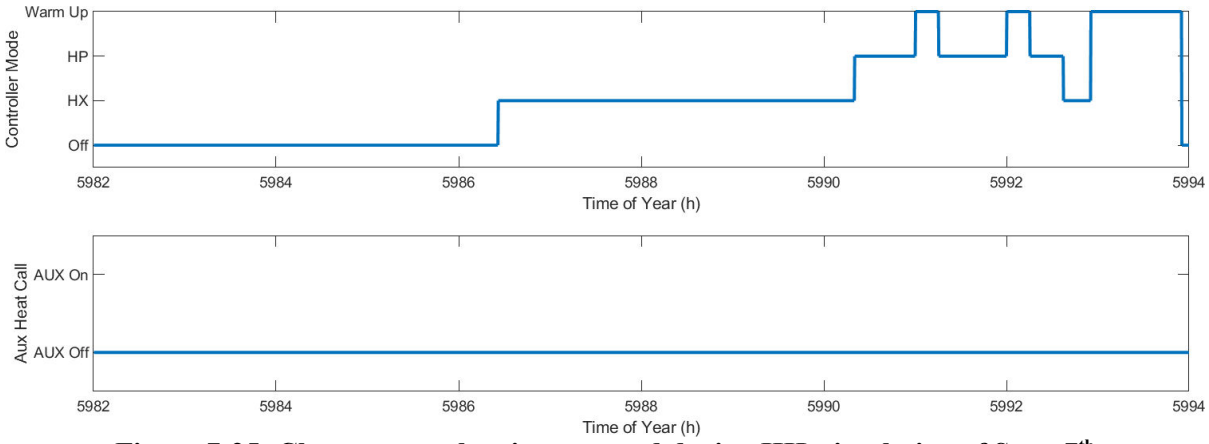


Figure 7-25: Chosen control series operated during HIL simulation of Sept. 7th

Notable features in Figure 7-24 include a similar deviation in the start-up transient of the HX as seen in the previous trial. For approximately 1 hour at 5986.45h, there is a positive deviation of the average tank temperature observed as compared with what was measured. Again, as explained for the previous trial, this is due to a pre-heated volume of water being contained within the HX and the solar plumbing loop on start-up and the ETU has no mechanism to simulate losses from the STC to the surroundings when they outweigh solar gains.

Fault conditions detected and avoided include magenta highlights indicating the potential for HP short-cycling, red highlights, indicating the potential for stagnation, and yellow highlights indicating draws whose load might not be met by the system. Reviewing the available energy and draw temperature plots on the following page illustrates how the system performed for the day.

Looking at the two figures below, a notable feature is the period of time between 5990.6 h and 5992 h when there is no DHW available at all (i.e. all energy storage in the tank is in water below 35°C). Unfortunately, this period of time coincides with the largest draw event of the day. This represents a significant malfunction of the controller.

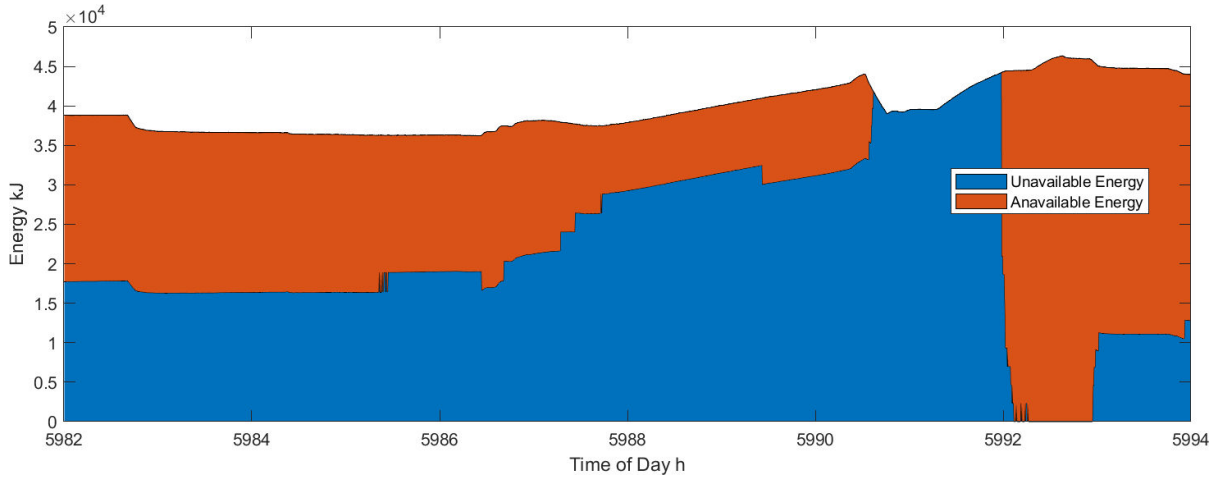


Figure 7-26: Plot of energy availability during HIL simulation of Sept. 7th

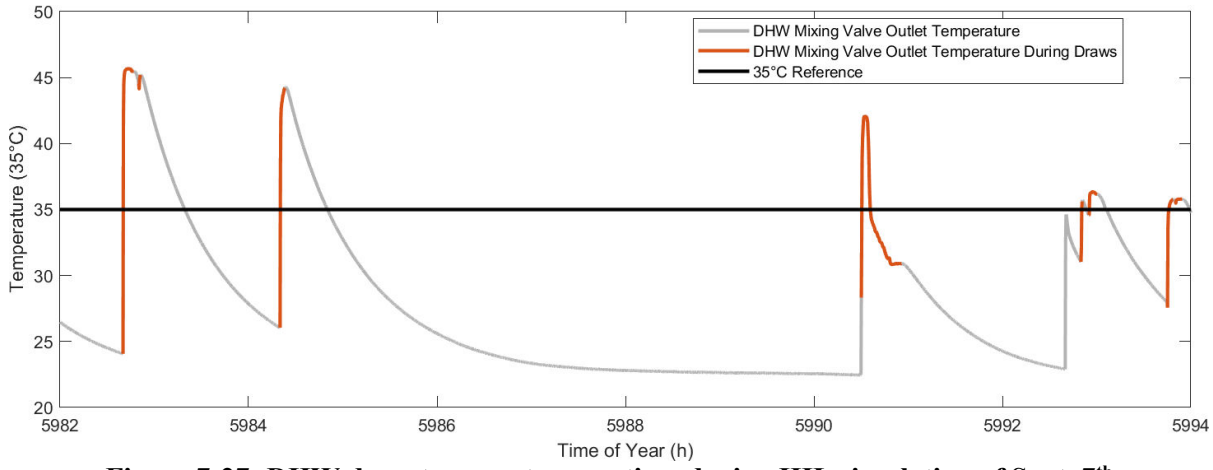


Figure 7-27: DHW draw temperature vs. time during HIL simulation of Sept. 7th

As seen in Figure 7-27, the draw that occurs at 5990.5 h experiences the usual start-up transient, then begins drawing at approximately 42°C before rapidly falling in temperature and leveling out at 30.9°C just before the draw terminates. This results in approximately 55 L of DHW being delivered below the threshold temperature. There were 18 under-temperature draws identified in the annual simulation using this control scheme presented in Chapter 6. This experimental trial highlights the type of conditions that can lead to these draws, understanding the cause is an important step toward future improvements to this system.

The following narrative highlights the series of events leading to the under-temperature delivery at 5990.5 h. Beginning early in the day, the system has been charging in HX mode and due to a small amount of tank mixing, although the total energy content of the tank has increased, the amount of

available energy has not significantly increased up to the hour before the largest draw of the day. It is likely at this point, however, that the system would have met the draw demand if solar radiation conditions had been better. As it is, in the hour before the large draw, the solar irradiation decreases sharply to a local minimum at 5990 h. The decrease in solar radiation causes the controller to change modes to use the HP to boost the STC outlet temperature so that it can maintain a positive charge rate. This mode change occurs at 5990.34 h and for the next 10 minutes, while there is still available DHW in the tank, it is decreasing even before the draw begins because of tank mixing during HP operation. Finally, as the DHW draw begins, the mains inlet temperature recorded by the ETU for the first time falls below the temperature predicted by the TMY file, and therefore even more energy is removed from the tank than predicted by the controller. The mains inlet temperature and residual are plotted below in Figure 7-28.

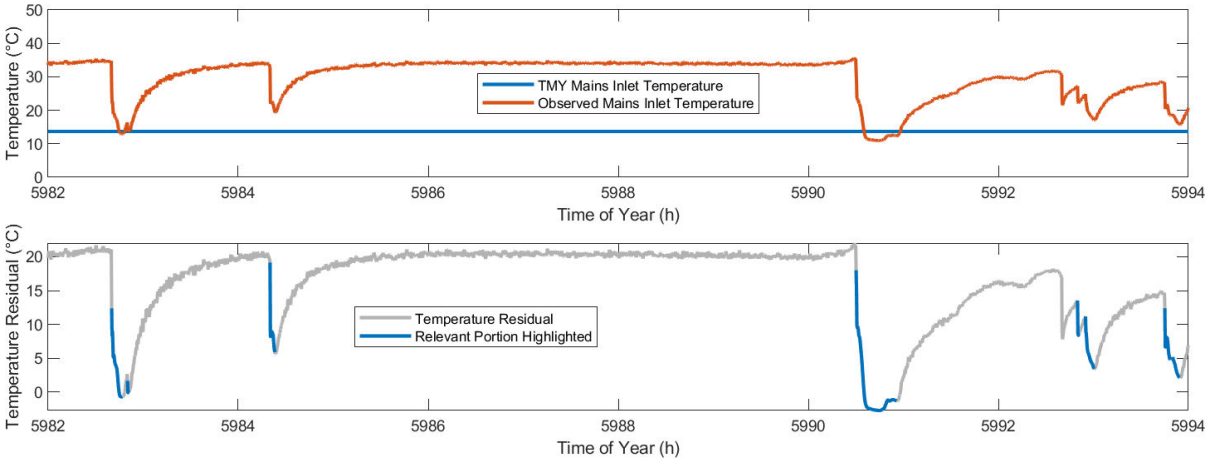


Figure 7-28: Predicted and observed mains temperature vs time and residual from Sept. 7th HIL simulation trial

In the above series of events more than one factor impacted the predictive controller's ability to properly predict the tank state. The combination of error in model inputs (the mains temperature), poor solar conditions (leading to a lack of reserve charge), and DHW tank model inaccuracy in modeling destruction of stratification, the controller predicted that the available DHW would be depleted at 5990.75 h, rather than the 5990.61 h that was observed. Figure 7-29 compares the observed conditions in the DHW Tank to the prediction made by the sub-simulation between 5990 h and 5994 h. Vertical lines in the plot indicate the timing error in predicting the depletion of DHW. A difference of 8.5 minutes is seen, which explains why the controller failed to schedule additional AUX heat to meet the load.

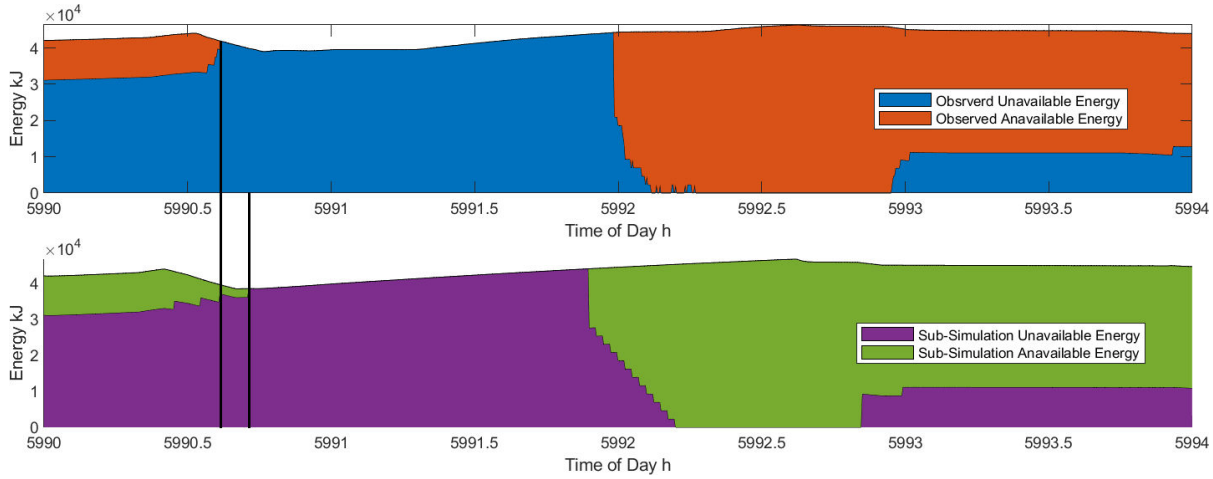


Figure 7-29: Comparison of observed and sub-simulated available energy in DHW tank leading into afternoon draw during Sept. 7thth HIL simulation trial

7.6 Conclusions and Recommendations

In this chapter, a series of hardware in the loop experimental trials were presented to demonstrate operation of the ETU being controlled by the predictive controller detailed in the previous chapter. The controller was implemented in LabVIEW with linked TRNSYS sub-simulations, again through custom MATLAB scripts.

Four day-long trials were reported that show a range of system behavior in response to variable weather and draw inputs. The trial for April 16th demonstrated that the controller correctly behaves during an extremely poor solar irradiation day by scheduling AUX heat to meet the draw demand.

The trial on May 29th showed reasonable operation of the system on a strong solar irradiation day, but some trouble was noted for an early-afternoon draw owing to both a small error in timing of when DHW would become available after the tank was mixed by HP operation, and a difference in the physical behavior of the thermostatic mixing valve compared to what was modeled in TRNSYS. It was also noted that the mains temperature in the STRL was significantly warmer than predicted by the TMY file on the day in question.

The trial on June 5th represented another fair-weather day for solar-thermal collection and the system performed amicably, charging the DHW tank to meet all draw demands using the HX alone.

The final trial presented was of Sept. 7th, which was shown to be a poor solar day in the morning, with a particularly large draw event in the early afternoon. On this day, similar to the May 29th trial, an error in timing between the predicted depletion of DHW and when it was actually observed caused a

part of one draw on this day to fall below the threshold temperature. It was observed that on this day poor insolation, below-expected mains inlet temperature, and an inaccuracy in TRNSYS' simulation of tank mixing contributed to this fault.

It should be noted that for the 4 trials presented, only 2 draw events were observed where the system missed the target DHW temperature during a draw. These are 2 faults out of 38 significant draw events observed during the trials, which indicates that the controller is largely performing as expected, but there are certain sets of challenging conditions including weather inputs, draw and mains temperature inputs, and timing that the controller is not yet able to handle.

A future iteration of this controller could be improved so that draw temperatures are more consistently above-threshold by implementing one or more of the following: First, whereas the current mains inlet temperature for the simulations is clearly not equal to the observed mains temperature in the STRL, either a new dataset for mains temperature should be used as a model input, or live measurement of the cold inlet temperature to the ETU should be used to calibrate the model input to better simulate energy removals during draw events. Second, while the objective function described in Section 6.1.4 seeks to minimize the "Energy Deficit", this may not provide an adequate buffer for the impact of timing differences in reaching key threshold temperature values within the DHW tank that may be owing to small differences in charge rate, mode switching, and tank mixing, especially when the HP is called. This could be mitigated by programming in either an energy buffer or a time buffer, or both. These buffers would have the effect that rather than targeting a certain amount of available energy for DHW at the beginning of a predicted draw event, the energy would be available some period before the draw and/or a certain amount of additional available energy would be available above what was predicted as the draw load. Further investigation would be required to determine what amount of time or energy buffer would be appropriate.

Yet another possible improvement that could be brought to this sort of system is a more accurate representation of the tank state at all times during operation and simulation. It has been mentioned more than once in this thesis that tank models in TRNSYS experience a weakness when it comes to simulating stratification and mixing effects. As a separate line of inquiry to the current work, the author has explored Bayesian updating for non-stationary state estimation as applied to a DHW storage tank. This separate work is included in E, and details the use of Kalman Filtering to improve the accuracy of tank state predictions during an experiment. It is possible that this sort of filtering

could be applied to model inputs or to the state representation of the system in the model used to predict the system state for control. While this is outside of the scope of the current project, it is suggested for consideration in future work.

Overall, the trials presented in this chapter show that the control architecture presented in Chapter 6 can be applied to a physical system with positive results. There would be value in continued experimental investigation of this system under a similar controls approach with some refinement to address the anomalies that were revealed. Furthermore, it is acknowledged that the trials presented in this chapter were limited to single-day trials. It was seen in the previous chapter that the system does not end its day in a consistent state of charge, and therefore 2^{nd}, 3^{rd}, to n^{th} day operation of the system is worth investigation through longer periods of HIL experimentation/simulation to continue to build confidence in this control strategy's ability to mitigate fault conditions while meeting the forecasted loads.

Finally, as a step towards more realistic in-situ testing, it is recommended that future work include the adoption of a live predictive model for DHW draws as opposed to the pre-packaged draw schedule used in this study. It would also be of benefit to either perturb the weather inputs to the hardware versus the sub-simulation model, or if possible, to feed the sub-simulations weather data derived from a weather forecast, while operating the ETU on weather data recorded at a local weather station. These suggestions would move the system closer, in terms of development, to a working prototype that might be retrofit in an existing dwelling, which should remain a long-term goal of the current line of inquiry into these types of SAHP systems.

Chapter 8
Overall Conclusions and Recommendations for Future Work

8.1 Conclusions

Work presented in this thesis is motivated by the high and ever-increasing cost of energy and the link between society's energy use and global climate change causing emissions. Material presented in the introduction section makes clear the need to reduce or eliminate emissions, which can be achieved in part through a reduction in the use of combustion as a heat source for a variety of end-uses. With the supply of domestic hot water accounting for nearly 20% of energy use in residential buildings, a focus on a renewable energy source for this load deserves attention.

While a body of literature on SDHW and SAHP systems exists, there remain gaps in the literature around system design and control. In particular the size of the HP has been problematic in previous studies because researchers have found it difficult to source HPs commercially with a small enough heat transfer capacity to function during low-sun conditions. This challenge leads to the current attention to a variable-capacity HP, which can turn down the capacity to accommodate reduced input.

In Chapter 3, a study was conducted to characterize and model a custom water-to-water HP with a variable speed compressor. The resulting model is in the form of three multivariate polynomial hypersurfaces that describe the unit's electric work input, and source and load side heat transfer rates. The power consumption of the HP is modeled within 5%, the source side heat transfer is modeled within 15% for source-side heat transfer, and the load side heat transfer rate is modeled within 13%.

The HP characterization study culminated in the development of a custom TRNSYS "TYPE" model that was later incorporated into overall SAHP system models in Chapter 4, that were used to generate the simulation results presented in subsequent chapters. Also, in Chapter 4, validation was presented for key system components as well as on the entire system model for short time-scale simulations.

The overall system model was created so that it could use interchangeable controllers implemented as MATLAB scripts. A series of short timescale simulations were conducted under temperature-based control which pointed to the advantage of modulating the HP compressor speed using a staged approach based on the source-side inlet temperature. It was concluded that with no appreciable change in electrical energy consumption, the HP could cycle less frequently and ultimately collect slightly more energy in a given period under modulating control.

Also conducted under temperature-based control were a series of annual simulations to develop a comparative set of performance indicators between multi-mode and single-mode solar domestic hot water systems operating with all combinations of external HX, HP, and electric resistance heating. Simulations involving the HP tested it under fixed and modulating capacity. Results from these annual simulations established a hierarchy of performance based on the quantity of solar energy collection favoring multi-mode systems, followed by HX-only systems, followed by HP-only systems. On the basis of other performance metrics measuring system adequacy and reliability, HP-Only systems were ranked as better performers, which is attributed to their general tendency to mix the DHW tank which leads to a larger amount of energy storage within the same tank.

It has been speculated by many researchers that systems that use thermostatic control as the main decision-making criteria have performed below the potential of the hardware under more advanced controls. The results from temperature-based controller simulations in this thesis led the author to the same conclusion. Predictive control of multi-mode SAHP/SDHW systems was investigated and reported on in Chapter 6. For annual simulations, a link was created between a top-level TRNSYS simulation and subordinate TRNSYS simulations covering a smaller timeframe through custom MATLAB functions. Multiple iterations of sub-simulations allowed the control algorithm to converge on an optimal control series for the timespan being considered. The objective function was to best-match the available energy stored in the DHW tank to the energy required by anticipated draw events. Under this control scheme, numerous advantages were observed.

The system under predictive control exhibited performance advantages in the areas of solar-fraction, purchased energy, system adequacy (draw capacity), and reduced fault conditions. The solar fraction nearly doubled under predictive control, with an increase of 98% in the amount of solar energy collected. Similarly, under the new control scheme, the amount of purchased energy was reduced by 61.8% on an annual basis. The makeup of energy stored in the DHW tank was compared between temperature-based and the predictive controller and it was seen that the predictive controller uses more HX and more HP during the shoulder seasons and also effectively reduces unnecessary AUX and HP use during the summer months. On the basis of system adequacy (draw capacity), there was a reduction by 82% of the mass of DHW that was drawn below the threshold temperature of 35°C.

On the basis of energy loss from the simulated systems, the predictive controller does show a performance disadvantage to simple temperature-based controllers. The reasons for this were

discussed, but ultimately amount to the use of the HX as a means of heat rejection during periods of low insolation in preparation for a later period of higher solar radiation that would otherwise lead to system stagnation.

In addition to the energy-based performance metrics, a key advantage of the new control scheme is its ability to predict and avoid potentially damaging fault conditions. Through forward-looking sub-simulations, the system is able to identify periods where system stagnation is likely to occur and operate in a manner that avoids an undesirable outcome. Another fault that was effectively reduced by the new predictive control scheme was HP short-cycling/under temperature conditions.

The application of predictive control to a variable-speed multi-mode SAHP system has been clearly demonstrated to offer advantages in performance and reliability, however the current implementation has the following aspects untested. First, the tolerance of the controller to model error has not been tested over long-term experiments. Second, the system has been simulated using TMY weather, which is not always a reflection of true weather patterns. Third, while the simulations incorporated draw data taken from an experimental study from real Canadian residences, the forcing functions were identical between top-level and subordinate simulations which represents an unrealistic advantage to the controller in this case. For these reasons, the results presented in Chapter 6 are thought to approach an upper-limit of performance for the system under study, while future additions can bring simulation results back towards a more probable, realistic operation of a physical system.

In Chapter 7, an experimental, hardware in the loop approach was presented for four day-long trials representing the operation of a Variable-SAHP system under predictive control similar to the one that was described and assessed as a simulation only in Chapter 6. Results in this section showed a promising response of the physical system to control based on iterative sub-simulations. Some deviations between model predictions and observed experimental results were highlighted and explained. The future work section that follows will outline steps that should be taken to address these shortcomings and further improve the state of the literature regarding SAHP systems.

8.1 Future Work

Future work that stems from the line of inquiry presented in this thesis is suggested to address the three limitations of the current control scheme's implementation that were previously mentioned. This work should be performed in the following order:

First of all, to assess whether the current controller implementation is tolerant of measurement and model error, a longer-term in-situ prototype trial should be conducted. The current implementation has been ported to a LabVIEW VI for this purpose and short HIL simulation trials have been conducted with promising results that should inform the next round of development on this system. To the extent that issues noted in Chapter 7 are due to ETU limitations, it is assumed that future work will either address them, or continue to expect and account for them as explained. To the extent that the system model could be improved to reduce timing or other errors between the predicted and observed system state, that should be a goal.

Secondly, in either the simulation-space, or using the in-situ prototype that is called for in the previous paragraph, the weather input to the system should be replaced from the current TMY file to real weather observations for the top-level and forecasted weather predictions for the sub-simulations. In doing so, the tolerance of the controller to realistic variation between observed and predicted input conditions can be assessed. Environment and Climate Change Canada provides research access to The Global Ensemble Prediction System (GEPS) [50] and Regional Deterministic Prediction System (RDPS) [51], which can be accessed programmatically and parsed to generate input weather forecast data for a location. In the case of GEPS, the ensemble forecasts can be interpreted as different weather scenarios that are probabilistically determined. These could be useful in controlling the SAHP system by assessing worst-case scenarios for solar irradiation, etc. In the case of RDPS, the most accurate forecast for the region could be extracted and used to drive the sub-simulations.

A final, and likely most impactful area of future work that immediately follows the work presented in this thesis is to implement some form of predictive model for DHW draw loads. In the current work the findings are limited in applicability due to the perfect clairvoyance that the predictive controller has of upcoming DHW draw energy demands. As a first-pass approach to this task, the DHW draw profiles from Edwards could be treated statistically to determine a probability density function that could be used along with a Monte Carlo style simulation to generate a different predicted draw schedule.

Another direction to take with draw prediction could involve training a machine-learning (ML) model on the existing draw data, and using it to predict upcoming draw events. An issue that the author sees with this approach, is that there likely isn't enough information encoded in previous draw events to fully predict the time and duration of the next draw. An ML model of draw loads would likely need to be augmented with additional inputs such as building occupancy sensor data, and information about the daily schedule of the users.

Yet another method that could be used to separate the DHW draw forcing function from the DHW draw prediction could be to establish an operating schedule that might be edited by the building occupants. Occupants could set time periods when they expect to draw significant amounts of DHW, (e.g. showers, laundry, dish-washing) and adhere to those times that they have entered. In between scheduled draws, the system could be configured to maintain a certain amount of DHW available for smaller draw events such as hand-washing. To the author, this seems to be a practical approach, but implementing and testing this scheduling strategy would likely need an in-situ prototype installation in an actual occupied residence.

In addition to work directly following the line of inquiry presented in this thesis, there are a number of areas called for in the literature that remain unexplored or have yet to be concluded. While this section will not seek to list all remaining gaps in the literature, some noteworthy ones will be explained. To begin with, previous studies at the STRL have investigated various system topologies and concluded with a call for further control development. As the predictive controller implementation in this thesis is improved, it will continue to be generally applicable to other SAHP system configurations that can be modeled in TRNSYS. As such it seems reasonable that future work would include investigation of the performance potential of the dual-tank SAHP system presented by Banister [3] under predictive control. In order to perform this study, care would need to be taken in the selection of sub-simulation time horizon since the large float tank in Banister's 2-tank system is often used to store heat at a relatively low temperature for longer periods of charge and discharge than the single tank system investigated in the current work.

Two other system configurations that have been preliminarily investigated in the STRL are a series-flow configuration where solar (source) side heat-transfer fluid passes first through a HX and then through a HP before returning to the STC, and a parallel flow configuration where some flow of the source is diverted through a HX while the remainder is passed through a HP before returning to the

STC. Both of these configurations have been preliminarily modeled in TRNSYS and it was determined that the capacity of HP required to investigate their performance was on the order of half of the capacity of the HP originally built into the ETU. Modifications made for the work presented in this thesis open the possibility of experimentally testing the series configuration using the lowest setting on the new variable capacity HP. The parallel-flow configuration remains outside of the range of conditions that the ETU could generate without modification, however it may be possible to modify the system further to facilitate future study of that configuration.

It remains to be explored, how SAHP systems might perform when coupled with other renewable and energy saving devices. Parallel work by others in the STRL has been focused on performance modeling of DWHR systems [52]. In the referenced work, TRNSYS models have been developed for DHW configurations using traditional electrical resistance or fossil fuel heat sources and DWHR systems to reduce energy consumption. A logical extension of both lines of inquiry would be to model a DHW plumbing configuration that uses SAHP as the heat source and incorporates a DWHR system to recover some of the DHW load energy. The key interactions of these two systems would be as follows: The DWHR system would pre-heat some of the mains water that enters the SAHP system, thereby reducing the load on the SAHP system. The SAHP system, however, may supply lower temperature water than expected to the DWHR system which could impede its performance and lead to a diminished return. Questions to be answered from this proposed work would then relate to whether such a combination could be justified on the basis of economics, lifecycle energy, and user-comfort.

The final avenue for future investigation that the author wishes to suggest is the construction and monitoring of an in-situ system incorporating a rooftop STC exposed to real-world weather, subject to real DHW loads of live building occupants, and instrumented and monitored for performance over a long-term study. The suggestion is not the construction of a test-house, which has been repeated by several researchers, but instead to engage in the exercise of a retro-fit installation to serve as close to a *real* client as possible. This would take the next step in moving the technology out of the lab and simulation space, with a view towards partnering with an equipment manufacturer and ultimately commercializing a packaged SAHP system operating on the control principles presented in this thesis.

Letter of Copyright Permission

begell
New York • Connecticut

50 North Street
Danbury, CT 06810
203-456-6161
journals@begellhouse.com
www.begellhouse.com

January 23, 2024

Julian Howarth
PhD Candidate and Sessional Lecturer
Department of Mechanical and Mechatronics Engineering
University of Waterloo

Re: Permission to Reprint Material

Dear Julian Howarth,

Permission is hereby granted to use **part or all material** from the following articles published by Begell House: "Experimental Characterization of a Variable Refrigerant Flow Heat Pump for Solar-Domestic-Hot-Water Applications," by J.C. Howarth & M.R. Collins, which was published in the *5-6th Thermal and Fluids Engineering Conference (TFEC)*, Virtual, pp. 405-414, May 26–28, 2021; "3-Stage Control of a Variable-Refrigerant-Flow Heat Pump for Solar Domestic Hot Water Applications," by J.C. Howarth & M.R. Collins, which was published in the *7th Thermal and Fluids Engineering Conference (TFEC)*, Las Vegas, NV, pp. 1227-1233, May 15-18, 2022; and "Annual Performance of a Simulated Multi-Mode SDHW/SAHP System with Various Control Options," by J.C. Howarth & M.R. Collins, which was published in the *8th Thermal and Fluids Engineering Conference (TFEC)*, College Park, MD, pp. 1113-1122, March 26-29, 2023.

Proposed Use:	The material will be used as part of a doctoral thesis titled, "Variable-Speed and Multi-Mode Solar Assisted Heat Pumps System Design and Controls Development"
Publisher:	University of Waterloo, Waterloo, Ontario, Canada
Year of Publication:	2024

As per your email, we hereby grant you permission to reproduce the aforementioned material in electronic and print format at no charge subject to the following conditions:

1 – If any part of the material to be used (for example, figures) has appeared in our publication with credit or acknowledgement to another source, permission must also be sought from that source. If such permission is not obtained then that material may not be included in your publication.

2 – Suitable acknowledgement to the source must be made, either as a footnote or in a reference list at the end of your publication, as follows:

"Reprinted from Publication title, Volume number, Author(s), Title of article, Page Nos., Copyright (Year), with permission from Begell House, Inc.

3 – Reproduction of this material is confined to the purpose for which permission is hereby given.

Yours sincerely

Brandon T. Bisceglia

Brandon T. Bisceglia
Begell House Inc.

References

[1] M. Allen, O. Dube, W. Solecki, F. Aragón-Durand, W. Cramer, S. Humphreys, M. Kainuma, J. Kala and N. Mahowald, "Framing and Context. In: Global Warming of 1.5°C.," The Intergovernmental Panel on Climate Change, 2018.

[2] W. R. Wagar, "Simulation and Validation of a Single Tank Heat Pump Assisted Solar Domestic Water Heating Sysem," University of Waterloo, Waterloo, 2013.

[3] C. J. Banister, "Design, Analysis, and Optimizaion of a Dual Tank Solar-Assisted Heat Pump System," University of Waterloo, Waterloo, 2015.

[4] Canada, "CANADA'S INDC SUBMISSION TO THE UNFCCC," 15 May 2015. [Online]. Available: https://www4.unfccc.int/sites/submissions/INDC/Published%20Documents/Canada/1/INDC%20-%20Canada%20-%20English.pdf. [Accessed March 2019].

[5] Statistics Canada, "Table 18-10-0204-01 Electric power selling price index, monthly," [Online]. Available: https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=1810020401. [Accessed May 2024].

[6] Statistics Canada, "Table 25-10-0033-01 Natural Gas Monthly Sales," [Online]. Available: https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=2510003301. [Accessed May 2024].

[7] Natural Resources Canada, "Comprehensive Energy Use Database," [Online]. Available: http://oee.nrcan.gc.ca/corporate/statistics/neud/dpa/menus/trends/comprehensive_tables/list.cfm. [Accessed 21 2019 March].

[8] A. A. M. Sayigh, "Characteristics of Solar Radiation," in *Solar Energy Conversion*, Toronto, Pergamon Press, 1979, pp. 9-36.

[9] International Energy Agency, "Total Final Consumption (TFC) by source," 2018. [Online]. Available: https://www.iea.org/statistics/?country=WORLD&year=2016&category=Energy%20consumption&indicator=TFCbySource&mode=chart&dataTable=BALANCES. [Accessed 27 March 2019].

[10] Natural Resourses Canada, "The Atlas of Canada - Clean Energy Resources and Projects (CERP)," 10 December 2018. [Online]. Available: http://atlas.gc.ca/cerp-rpep/en/. [Accessed 27 March 2019].

[11] M. A. Green, "Photovoltaic Physics and Devices," in *Solar Energy The State of The Art*, London, James & James, 2001, pp. 307-311.

[12] National Renewable Energy Laboratory, "Best Research-Cell Efficiency Chart," 21 March 2019. [Online]. Available: https://www.nrel.gov/pv/assets/pdfs/best-reserch-cell-efficiencies.pdf. [Accessed 26 March 2019].

[13] G. L. Morrison, "Solar Collectors," in *Solar Energy The State of the Art*, London, James & James, 2001, pp. 159-162.

[14] J. A. Duffie and W. A. Beckman, Solar Engineering of Thermal Processes, Hoboken: Wiley, 2013.

[15] M. J. Moran, H. N. Shapiro, D. D. Boettner and M. B. Bailey, Fundamentals of Engineering Thermodynamics, Hoboken, NJ: Wiley, 2011.

[16] S. J. Sterling, "Feasibility Analysis of Two Indirect Heat Pump Assisted Solar Domestic Hot Water Systems," University of Waterloo, M.A.Sc. Thesis, Waterloo, 2011.

[17] A. Bridgeman and S. Harrison, "Preliminary experimental evaluations of indirect solar assisted heat pump systems," in *3rd Canadian Solar Building Conference*, Fredericton, New Brunsiwck, Canada, 2008.

[18] C. T. Chasse, "Evaluation of a Series-Configuration Solar-Assisted Heat Pump with Performance-Predictive Controls," 2017.

[19] B. Christopher and C. Cynthia A., "Assessing the Potential for Reduction in Peak Residential Electrical Loads Using a Heat Pump and Thermal Storage Systems. International High Performance Buildings Conference," in *International High Performance Buildings Conference*, 2016.

[20] S. J. Harrison, "The Potential and Challenges of Solar Boosted Heat Pumps for Domestic Hot Water Heating," in *12th IEA Heat Pump Conference*, 2017.

[21] E. F. Camacho and C. A. Bordons , Model Predictive Control, London: Springer, 2007.

[22] T. Zheng, Model Predictive Control, Rijeka: IntechOpen, 2010.

[23] J. Chu, C. A. Cruickshank, W. Choi and S. J. Harrison, "Modelling of an Indirect Solar-Assisted Heat Pump System for a High Performance Residential House," in *ASME 2013 7th International Conference on Energy Sustainability*, 2013.

[24] J. Chu and C. A. Cruickshank,, "Solar-Assisted Heat Pump Systems: A Review of Existing Studies and Their Applicability to the Canadian Residential Sector," *Journal of Solar Energy Engineering,* vol. 136, no. 4, p. 041013, 2014.

[25] J. Bursill and C. C. A., "Heat Pump Water Heater Control Strategy Optimization for Cold Climates," in *ASME 2015 Power Conference*, 2015.

[26] J. Berger and S. J. Harrison, "MODELING OF PHOTOVOLTAIC, SOLAR THERMAL, AND PHOTOVOLTAIC/THERMAL DOMESTIC HOT WATER SYSTEMS," in *IBSA eSIM 2016 Conference Proceedings*, 2016.

[27] S. R. Asaee, V. I. Ugursal and I. Beausoleil-Morrison, "Techno-economic assessment of solar assisted heat pump system retrofit in the Canadian housing Stock," *Applied Energy,* vol. 190, p. 439–452, 2017.

[28] C. M. Unrau and M. F. Lightstone, "Analysis of Numerical Error in One-Dimensional Storage Tank Models for Solar Energy System Simulations," *Journal of Solar Energy Engineering,* vol. 140, no. 5, p. 051004, 2018.

[29] M. S. Buker and S. B. Riffat, "Solar assisted heat pump systems for low temperature water heating applications: A systematic review," *Renewable and Sustainable Energy Reviews,* vol. 55, p. 399–413, 2016.

[30] D. Carbonell, D. Philippen, E. Frank and M. Granzotto, "Simulation of Combined Solar Thermal, Heat Pump, Ice Storage and Waste Water Heat Recovery Systems. Design Criteria and Parametric Studies," in *Proceedings of the EuroSun 2014 Conference*, 2014.

[31] C. Fraga, P. Hollmuller, F. Mermoud and B. Lachal, "Solar assisted heat pump system for multifamily buildings: Towards a seasonal performance factor of 5? Numerical sensitivity analysis based on a monitored case study," *Solar Energy,* vol. 146, p. 543–564, 2017.

[32] S. Poppi, C. Bales, A. Heinz, F. Hengel, D. Chèze, I. Mojic and a. C. Cialani, "Analysis of system improvements in solar thermal and air source heat pump combisystems," *Applied Energy,* vol. 173, p. 606–623, 2016.

[33] E. Bellos and C. Tzivanidis, "Energetic and financial sustainability of solar assisted heat pump heating systems in Europe," *Sustainable Cities and Society,* vol. 33, p. 70–84, 2017.

[34] W. Youssef, Y. G. and S. Tassou, "Effects of latent heat storage and controls on stability and performance of a solar assisted heat pump system for domestic hot water production," *Solar Energy,* vol. 150, p. 394–407, 2017.

[35] Z. Wang, P. Guo, H. Zhang, W. Yang and a. S. Mei, "Comprehensive review on the development of SAHP for domestic hot water," *Renewable and Sustainable Energy Reviews,* vol. 72, p. 871–881, 2017.

[36] J. Howarth and M. Collins, "Experimental Characterization of a Variable Refrigerant Flow Heat Pump for Solar-Domestic-Hot-Water Applications," in *5-6th Thermal and Fluids Engineering Conference (TFEC)* , 2021.

[37] J. C. Howarth and M. R. Collins, "Annual performance of a simulated multi-mode SDHW/SAHP system with various control options," in *8th Thermal and Fluids Engineering Conference (TFEC)*, Maryland, MD, 2023.

[38] A.O. Smith, "Residential Direct Solar Booster Water Heaters," A.O. Smith, Fergus, Ontario, February 2011.

[39] Taco (Canada) Ltd, Model 008-IFC Cartridge Circulator (Product Data Sheet), Milton, Ontario.

[40] B. R. Munson, T. H. Okiishi, W. W. Huebsch and A. P. Rothmayer, Fundamentals of Fluid Mechanics, Hoboken, NJ: Wiley, 2013.

[41] Northern Lights Solar Solutions, "Solar Plate Heat Exchangers SL15-25 (25 Plate)," December 2023. [Online]. Available: https://www.solartubs.com/solar-plate-heat-exchanger.html.

[42] F. P. Incropera, D. P. Dewitt, T. L. Bergman and A. S. Lavine, Fundamentals of Heat and Mass Transder, Hoboken: Wiley, 2007.

[43] J. Antony, Design of Experiments for Engineers and Scientists, Oxford: Elsevier, 2003.

[44] M. Daskilewicz and B. German, "RAVE: A Graphically Driven Framework for Agile Design-Decision Support," in *AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, Fort Worth, Texas, 2010.

[45] Honeywell Chemicals, *Honeywell Genetron®134a Properties, Uses,Storage, and Handling,* Morristown, NJ: Honeywell Chemicals, 2006.

[46] L. Huang, V. Aute and R. and Radermacher, "Uncertainty Analysis on Prediction of Heat Transfer Coefficient and Pressure Drop in Heat Exchangers Due to Refrigerant Property Prediction Error," in *International Refrigeration and Air Conditioning*, 2014.

[47] Canadian Standards Association, *Packaged solar domestic hot water systems (liquid-to-liquid heat transfer) for all-season use,* Mississauga, ON: Canadian Standards Association, 2013.

[48] S. Edwards, I. Beausoleil-Morrison and A. Laperrière, "Representative hot water draw profiles at high temporal resolution for simulating the performance of solar thermal systems," *Solar Energy,* no. 111, pp. 43-52, 2015.

[49] J. Howarth and M. Collins, "3-Stage Control Of A Variable-Refrigerant-Flow Heat Pump For Solar Domestic Hot Water Applications," in *7th Thermal and Fluids Engineering Conference (TFEC)*, Las Vegas, NV, 2022.

[50] Environment and Climate Change Canada, "Global Ensemble Prediction System," Government of Canada, 2022. [Online]. Available: https://catalogue.ec.gc.ca/geonetwork/srv/api/records/6d9dd2f8-202e-58cb-a110-e2168832aacb. [Accessed 12 March 2024].

[51] Environment and Climate Change Canada, "Regional Deterministic Prediction System," Government of Canada, 2022. [Online]. Available: https://open.canada.ca/data/en/dataset/a9f2828c-0d78-5eb6-a4c7-1fc1219f1e3d. [Accessed 12 March 2024].

[52] R. Manouchehri and C. R. Michael, "Investigating the Impact of Plumbing Configuration on Energy Savings for Falling-Film Drain Water Heat Recovery Systems," *Energies,* vol. 3, no. 15, p. 1141, 2022.

[53] T. Kreuzinger, M. Bitzer and W. Marqardt, "State estimation of a stratified storage tank," *Control Engineering Practice,* no. 16, pp. 308-320, 2008.

[54] J. Kaipio and E. Somersalo, Statistical and Computational Inverse Problems, New York: Springer, 2005.

[55] Wan and v. d. Merwe, "Kalman Filtering and Neural Networks," Department of Electrical and Computer Engineering, Oregon Graduate Institute ofScience and Technology, Beaverton, Oregon, 2001.

[56] J. McNames, "SST T20 Unscented Kalman Filter," [Online]. Available: https://youtu.be/m5k5Gjsc2sU.

[57] The Intergovernmental Panel on Climate Change, "CLIMATE CHANGE," Cambridge University Press, Cambridge, 1990.

[58] United Nations Climate Change, "Global Warming Potentials," [Online]. Available: https://unfccc.int/process/transparency-and-reporting/greenhouse-gas-data/greenhouse-gas-data-unfccc/global-warming-potentials. [Accessed March 2019].

[59] Fchart Software, "EES Property Information," Fchart Software, Madison, WI, 2020.

[60] Stocker, T.F., D. Qin, G.-K. Plattner, M. Tignor, S. Allen, J. Boschung, A. Nauels, Y. Xia and V. B. a. P. Midgley, "IPCC, 2013: Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change," Cambridge University Press, New York, 2013.

Appendices

Appendix A

Example Uncertainty Calculations

A.1 Example Uncertainty Calculation for Mass Flow Rate:

The following is an uncertainty calculation for a typical mass flow rate of water.

Working Data:

Parameter	Value and Uncertainty
Typical Volumetric Flow Rate	$\dot{V} = 10.75 \pm 0.16\,\mathrm{L/min}$
Typical Water Temperature	$T = 20 \pm 0.1°\mathrm{C}$
Typical Water Pressure (Mains Pressure)	$P = 3.5 \pm 0.7\ \mathrm{bar}$
DAQ Uncertainty:	Taken to be $\pm$ 0 Hz, see note in Table 3-1.

Sample Calculation:

$$\dot{m} = \rho \cdot \dot{V} \qquad (A-1.1)$$

$$\rho(T,P) = 998.3[kg/m^3], \qquad \frac{\partial \rho}{\partial T} = -0.209, \qquad \frac{\partial \rho}{\partial P} = 0.04574^{1}$$

$$\delta\rho = \sqrt{\left(\delta T \cdot \frac{\partial \rho}{\partial T}\right)^2 + \left(\delta P \cdot \frac{\partial \rho}{\partial P}\right)^2} \qquad (A-1.2)$$

$$= \sqrt{(0.1 \cdot -0.209)^2 + (0.7 \cdot 0.04574)^2} = \pm 0.03823$$

$$\delta\rho = \pm 0.04[kg/m^3], \qquad \rho = 9.983 \times 10^{-1} \pm 4 \times 10^{-5}[\mathrm{kg/L}]$$

$$\dot{m} = 9.983 \times 10^{-1}\,[kg/L] \cdot 10.75\,[L/min] = 10.73[\mathrm{kg/min}]$$

$$\frac{\partial \dot{m}}{\partial \rho} = \dot{V}, \qquad \frac{\partial \dot{m}}{\partial \dot{V}} = \rho$$

$$\delta\dot{m} = \sqrt{\left(\dot{V} \cdot \delta\rho\right)^2 + \left(\rho \cdot \delta\dot{V}\right)^2} \qquad (A-1.3)$$

$$\delta\dot{m} = \sqrt{(10.75 \cdot 4 \times 10^{-5})^2 + (9.983 \times 10^{-1} \cdot 0.16)^2} = 0.399$$

$$\delta\dot{m} = \pm 0.4\ [\mathrm{kg/min}]\ or\ 3.7\%, \qquad \boxed{\dot{m} = 10.7 \pm 0.4\ [\mathrm{kg/min}]}$$

[1] Fluid Properties and associated uncertainties were evaluated using Fchart Software's EES data lookup tool [59]

A.2 Example Uncertainty Calculation for Heat Transfer Rate:

The following is an uncertainty calculation for a typical heat transfer rate. This could apply at the HP evaporator, condenser, or at the HX in the ETU.

Working Data:

Parameter	Value and Uncertainty
Typical Mass Flow Rate	$\dot{m} = 10.7 \pm 0.4$ [kg/min] $= 0.1783 \pm 6.67 \times 10^{-3}$ [kg/s]
Typical Water Inlet Temperature	$T_{in} = 18.5 \pm 0.1$℃
Typical Water Outlet Temperature	$T_{out} = 21.5 \pm 0.1$℃
Typical Water Pressure (Mains Pressure)	$P = 3.5 \pm 0.7$ bar
DAQ Uncertainty:	Taken to be included in the temperature uncertainty above.

Sample Calculation:

$$\dot{Q} = \dot{m}C_p(T_{out} - T_{in}) = \dot{m}C_p(\Delta T) \qquad (A-2.1)$$

$$C_p(T_{avg}, P_{avg}) = 4.182[kJ/kg\ K], \qquad \frac{\partial C_p}{\partial T} = 7.90 \times 10^{-5}, \qquad \frac{\partial C_p}{\partial P} = 3.16 \times 10^{-4}$$ [2]

$$\delta C_p = \sqrt{\left(\delta T \cdot \frac{\partial C_p}{\partial T}\right)^2 + \left(\delta P \cdot \frac{\partial C_p}{\partial P}\right)^2} \qquad (A-2.2)$$

$$\delta C_p = \pm 2.21 \times 10^{-4}[KJ/Kg\ K], \qquad C_p = 4.182 \pm 2.21 \times 10^{-4}[kJ/kg\ K]$$

$$\Delta T = T_{out} - T_{in} = (21.5 \pm 0.1°C) - (18.5 \pm 0.1°C) = 3 \pm 0.2°C$$

$$\frac{\partial \dot{Q}}{\partial \dot{m}} = C_p \Delta T, \qquad \frac{\partial \dot{Q}}{\partial \mathrm{C_p}} = \dot{m}\Delta T, \qquad \frac{\partial \dot{Q}}{\partial \Delta \mathrm{T}} = \dot{m}\mathrm{C_p}$$

$$\delta \dot{Q} = \sqrt{\left(\delta \dot{m} \cdot C_p \Delta T\right)^2 + \left(\delta C_p \cdot \dot{m}\Delta T\right)^2 + \left(\delta \Delta T \cdot \dot{m}\mathrm{C_p}\right)^2} \qquad (A-2.3)$$

$$\delta \dot{Q} = \sqrt{(6.67 \times 10^{-3} \cdot 4.182 \cdot 3)^2 + (2.21 \times 10^{-4} \cdot 0.1783 \cdot 3)^2 + (0.2 \cdot 0.1783 \cdot 4.182)^2}$$

$$\delta \dot{Q} = \pm 0.171[\mathrm{kW}]\ or\ 7.6\%, \qquad \boxed{\dot{Q} = 2.24 \pm 0.17\ [\mathrm{kW}]}$$

[2] Fluid Properties and associated uncertainties were evaluated using Fchart Software's EES data lookup tool [59]

A.3 Example Uncertainty Calculation for Electrical Power Consumption

The following is an uncertainty calculation for Electrical Power Consumption. This could apply at the HP or the DHW auxiliary heater.

Working Data:

Parameter	Value and Uncertainty
Typical Voltage	$V = 200$ [V]
Typical Current Draw	$I = 15$ [A]
Potential Transformer Accuracy	$\pm 1\%$
Current Transformer Accuracy	$\pm 1\%$
DAQ Accuracy (Voltage Measurement):	$\delta V_{daq} \pm 0.006$ [V]

Sample Calculation:

Follow Mains Voltage Signal from measurement point to DAQ.

$$V_{Transformer\ out} = \frac{0.333}{230} V_{transformer\ in} \qquad (A-3.1)$$

$$V_{Transformer\ out} = \frac{0.333}{230} * 200[\text{V}] = 0.2896[\text{V}]$$

$$\delta V_{transformer\ out} = 2.9 \times 10^{-3}[\text{V}]\ (1\%)$$

$$V_{measured\ ,DAQ} = \frac{230}{0.333} \cdot V_{transformer\ out} \qquad (A-3.2)$$

$$\delta V_{measured\ ,DAQ} = \frac{230}{0.333} \sqrt{\left(\delta V_{DAQ}\right)^2 + \left(\delta V_{Transformer\ out}\right)^2} \qquad (A-3.3)$$

$$\delta V_{measured\ ,DAQ} = \frac{230}{0.333} \sqrt{(6.0 \times 10^{-3})^2 + (2.9 \times 10^{-3})^2} = \pm 4.6[\text{V}]$$

$$V_{measured\ ,DAQ} = 200.0 \pm 4.6[\text{V}]$$

Follow Current Signal from measurement point to DAQ.

$$V_{Current\ Signal\ out} = \frac{0.333}{25} I_{in} \qquad (A-3.1)$$

$$V_{Current\ Signal\ out} = \frac{0.333}{25} * 15[A] = 0.1998[\text{A}]$$

$$\delta V_{Current\ Signal\ out} = 2.0 \times 10^{-3}[\text{V}]\ (1\%)$$

$$I_{measured\ ,DAQ} = \frac{25}{0.333} \cdot V_{transformer\ out} \qquad (A-3.2)$$

$$\delta I_{measured\ ,DAQ} = \frac{25}{0.333} \sqrt{\left(\delta V_{DAQ}\right)^2 + \left(\delta V_{Current\ Signal\ out}\right)^2} \qquad (A-3.3)$$

$$\delta I_{measured\,,DAQ} = \frac{25}{0.333}\sqrt{(6.0\times10^{-3})^2 + (\,2.0\times10^{-3})^2} = \pm 0.47[\mathrm{A}]$$

$$I_{measured\,,DAQ} = 15.00 \pm 0.47[\mathrm{A}]$$

Calculate Power using measured V and I.

$$\dot{W}_{el} = I \cdot V \qquad (A-3.4)$$

$$\frac{\partial \dot{W}_{el}}{\partial I} = V, \qquad \frac{\partial \dot{W}_{el}}{\partial V} = I$$

$$\delta \dot{W}_{el} = \sqrt{\left(V \cdot \delta I_{measured\,,DAQ}\right)^2 + \left(I \cdot \delta V_{measured\,,DAQ}\right)^2} \qquad (A-3.5)$$

$$\delta \dot{W}_{el} = \sqrt{(200\cdot 0.47)^2 + (15\cdot 4.6)^2} = \pm 100[\mathrm{W}]\ or\ 3.3\%, \qquad \boxed{\dot{W}_{el} = 3000 \pm 100[\mathrm{W}]}$$

A.4 Example Uncertainty Calculation for HP COP:

The following is an uncertainty calculation for heat Pump COP.

Working Data:

Parameter	Value and Uncertainty
Typical HP Power Consumption	$\dot{W}_{el} = 564 \pm 19$ [W]
Typical Load-Side Heat Transfer	$Q_{load} = 2240 \pm 170$ [W]

Sample Calculation:

$$COP_h = \frac{\dot{Q}_{load}}{\dot{W}_{el}} \qquad (A-4.1)$$

$$\frac{\partial COP}{\partial \dot{Q}_{load}} = \frac{1}{\dot{W}_{el}}, \qquad \frac{\partial COP}{\partial \dot{W}_{el}} = -\frac{\dot{Q}_{load}}{\dot{W}_{el}^2}$$

$$\delta COP = \sqrt{\left(\delta \dot{Q}_{load}\cdot\frac{1}{\dot{W}_{el}}\right)^2 + \left(\delta \dot{W}_{el}\cdot\frac{-\dot{Q}_{load}}{\dot{W}_{el}^2}\right)} \qquad (A-4.2)$$

$$\delta COP = \sqrt{\left(170\cdot\frac{1}{564}\right)^2 + \left(19\cdot\frac{-2240}{564^2}\right)^2} = 0.3298$$

$$\delta COP = \pm 0.33\ or\ 8.3\%, \qquad \boxed{COP = 3.97 \pm 0.33[\mathrm{dimensionless}]}$$

Appendix B

Coefficients for HP Characteristic Equations

Table B-1: Listing of coefficients of HP characteristic equations for $\dot{W}_{el}$, $\dot{Q}_s$, and $\dot{Q}_L$ as functions of $\dot{m}_1$, $\dot{m}_2$, T_{si}, T_{li}, and N_c

Coeff.	$\dot{W}_{el}$	$\dot{Q}_S$	$\dot{Q}_L$	Coeff.	$\dot{W}_{el}$	$\dot{Q}_S$	$\dot{Q}_L$
a_1	0.000	0.000	0.000	a_{29}	1.030×10^{-3}	0.01040	-0.05044
a_2	0.000	0.000	0.000	a_{30}	50.07	464.5	-1416
a_3	-1816	-3032	-3136	a_{31}	-8.309	-627.7	3055
a_4	226.9	-4092	3780	a_{32}	-4.239×10^{-3}	0.2521	-0.03720
a_5	20.18	6.025	433.3	a_{33}	0.01592	-5.529×10^{-3}	0.3503
a_6	3.040	-14.88	6.628	a_{34}	-1.829	4.999	-71.96
a_7	-0.6882	12.59	-14.21	a_{35}	458.7	6482	-1.843×10^{4}
a_8	8.419×10^{-3}	0.4145	-0.2863	a_{36}	-563.6	-6711	1.157×10^{4}
a_9	6.877×10^{-4}	-0.02041	0.02212	a_{37}	314.5	498.1	9159
a_{10}	-201.2	3925	-3029	a_{38}	-3.465	8.241	16.52
a_{11}	-1.937	267.4	-862.2	a_{39}	1.892	-1.883	-23.09
a_{12}	-2.331	0.4195	31.46	a_{40}	1.365×10^{-3}	-9.587×10^{-3}	0.01371
a_{13}	-0.4381	1.154	0.9131	a_{41}	-0.7272	32.66	-107.4
a_{14}	-3.611×10^{-4}	-0.6382	0.4163	a_{42}	2.323	-35.58	105.6
a_{15}	5.115×10^{-4}	0.02010	-0.02004	a_{43}	1.723×10^{-3}	-6.796×10^{-3}	-2.263×10^{-3}
a_{16}	0.9634	-14.35	12.32	a_{44}	-3.702×10^{-3}	0.01941	-0.01468
a_{17}	-0.09393	0.1040	-0.4129	a_{45}	-1.025	-478.9	3816
a_{18}	2.633×10^{-4}	-0.02090	0.01836	a_{46}	-0.3380	581.1	-4040
a_{19}	-1.017×10^{-3}	0.02243	-0.01868	a_{47}	2.800×10^{-3}	0.2552	-0.4286
a_{20}	-492.6	1865	-8546	a_{48}	-9.830×10^{-3}	-0.1092	0.4658
a_{21}	0.000	0.000	0.000	a_{49}	0.1279	-6.254	9.330
a_{22}	88.01	240.1	-9115	a_{50}	31.74	-649.4	268.6
a_{23}	4.436	-11.97	31.56	a_{51}	-57.38	503.3	-286.2
a_{24}	-3.906	3.126	-17.97	a_{52}	0.01310	-0.01277	0.1251
a_{25}	7.884×10^{-4}	0.01759	-0.01643	a_{53}	0.01375	-0.3731	0.4690
a_{26}	-3.714	-15.06	33.37	a_{54}	0.05739	-3.053	4.603
a_{27}	3.708	17.51	-12.74	a_{55}	0.6770	11.35	-7.928
a_{28}	-0.003096	-0.02706	1.209×10^{-3}				

The following equation B-1 is an example of the form of the characteristic equations that are used to calculate $\dot{W}_{el}$, $\dot{Q}_s$, or $\dot{Q}_L$ as functions of $\dot{m}_1$, $\dot{m}_2$, T_{si}, T_{li}, and N_c.

$$Y = a_1N_c + a_2N_c^2 + a_3N_c^3 + a_4T_{li} + a_5T_{li}N_c + a_6T_{li}N_c^2 + a_7T_{li}^2 + a_8T_{li}^2N_c + a_9T_{li}^3 + a_{10}T_{si} + a_{11}T_{si}N_c + a_{12}T_{si}N_c^2 + a_{13}T_{si}T_{li} + a_{14}T_{si}T_{li}N_c + a_{15}T_{si}T_{li}^2 + a_{16}T_{si}^2 + a_{17}T_{si}^2N_c + a_{18}T_{si}^2T_{li} + a_{19}T_{si}^3 + a_{20}\dot{m}_2 + a_{21}\dot{m}_2N_c + a_{22}\dot{m}_2N_c^2 + a_{23}\dot{m}_2T_{li} + a_{24}\dot{m}_2T_{li}N_c + a_{25}\dot{m}_2T_{li}^2 + a_{26}\dot{m}_2T_{si} + a_{27}\dot{m}_2T_{si}N_c + a_{28}\dot{m}_2T_{si}T_{li} + a_{29}\dot{m}_2T_{si}^2 + a_{30}\dot{m}_2^2 + a_{31}\dot{m}_2^2N_c + a_{32}\dot{m}_2^2T_{li} + a_{33}\dot{m}_2^2T_{si} + a_{34}\dot{m}_2^2 + a_{35}\dot{m}_1 + a_{36}\dot{m}_1N_c + a_{37}\dot{m}_1N_c^2 + a_{38}\dot{m}_1T_{li} + a_{39}\dot{m}_1T_{li}N_c + a_{40}\dot{m}_1T_{li}^2 + a_{41}\dot{m}_1T_{si} + a_{42}\dot{m}_1T_{si}N_c + + a_{43}\dot{m}_1T_{si}T_{li} + a_{44}\dot{m}_1T_{si}^2 + a_{45}\dot{m}_1\dot{m}_2 + a_{46}\dot{m}_1\dot{m}_{2N_c} + a_{47}\dot{m}_1\dot{m}_2T_{li} + a_{48}\dot{m}_1\dot{m}_2T_{si} + a_{49}\dot{m}_1\dot{m}_2^2 + + a_{50}\dot{m}_1^2 + a_{51}\dot{m}_1^2N_c + a_{52}\dot{m}_1^2T_{li} + a_{53}\dot{m}_1^2T_{si} + a_{54}\dot{m}_1^2\dot{m}_2 + a_{55}\dot{m}_1^3 \qquad B-1$$

Appendix C

Fortran Source Code for Custom Type 1162 HP Model

The following FORTRAN code was written to incorporate the empirical model of the Ecologix W2W090-VF-UW134A Heat pump for use in the simulations presented in the above sections.

```
1       Subroutine Type1162
2
3   ! Object: Variable Speed Water-to-Water Heat Pump
4   ! Simulation Studio Model: Type1162
5   !
6
7   ! Author: Julian Howarth
8   ! Editor: N/A
9   ! Date:  February 05, 2019
10  ! last modified: July 8, 2019
11  !
12  !
13  ! ***
14  ! *** Model Parameters
15  ! ***
16  !           CpSource    kJ/kg.K [-Inf;+Inf]
17  !           CpLoad  kJ/kg.K [-Inf;+Inf]
18
19  ! ***
20  ! *** Model Inputs
21  ! ***
22  !           HPCall  - [0;1]
23  !           nCompressor - [0;1]
24  !           SourceFlowRate  kg/hr [-inf;+Inf]
25  !           LoadFlowRate    kg/hr [-Inf;+Inf]
26  !           TSourceIn   C [-Inf;+Inf]
27  !           TLoadIn C [-Inf;+Inf]
28
29  ! ***
30  ! *** Model Outputs
31  ! ***
32  !           SourceFlowRate  kg/hr [-Inf;+Inf]
33  !           LoadFlowRate    kg/hr [-Inf;+Inf]
34  !           QSource W [-Inf;+Inf]
35  !           QLoad   W [-Inf;+Inf]
36  !           Power   W [-Inf;+Inf]
37  !           COP - [-Inf;+Inf]
38  !           TSourceOut  C [-Inf;+Inf]
39  !           TLoadOut    C [-Inf;+Inf]
40  !           TSourceIn   C [-Inf;+Inf]
41  !           TLoadIn C [-Inf;+Inf]
42  !           EBE W [-Inf;+Inf]
43
```

```
44    ! ***
45    ! *** Model Derivatives
46    ! ***
47
48    ! (Comments and routine interface generated by TRNSYS Simulation Studio)
49
50          Use TrnsysConstants
51          Use TrnsysFunctions
52
53    !-------------------------------------------------------------------------------
54
55    !DEC$Attributes DLLexport :: Type1162
56
57    !-------------------------------------------------------------------------------
58    !Trnsys Declarations
59          Implicit None
60
61          Double Precision Timestep,Time
62          Integer CurrentUnit,CurrentType
63
64
65    !     PARAMETERS
66          DOUBLE PRECISION CpSource
67          DOUBLE PRECISION CpLoad
68
69    !     INPUTS
70          DOUBLE PRECISION HPCall
71          DOUBLE PRECISION nCompressor
72          DOUBLE PRECISION SourceFlowRate
73          DOUBLE PRECISION LoadFlowRate
74          DOUBLE PRECISION TSourceIn
75          DOUBLE PRECISION TLoadIn
76
77    !-------------------------------------------------------------------------------
78    !USER Declarations
79    !Storage to replace values if outside of range of operation
80          DOUBLE PRECISION HPCall_Stor
81          DOUBLE PRECISION nCompressor_Stor
82          DOUBLE PRECISION SourceFlowRate_Stor
83          DOUBLE PRECISION LoadFlowRate_Stor
84          DOUBLE PRECISION TSourceIn_Stor
85          DOUBLE PRECISION TLoadIn_Stor
86    ! Power fit
87         DOUBLE PRECISION Power
88         DOUBLE PRECISION Power_a1
89         DOUBLE PRECISION Power_a2
90         DOUBLE PRECISION Power_a3
91         DOUBLE PRECISION Power_a4
92         DOUBLE PRECISION Power_a5
93         DOUBLE PRECISION Power_a6
94         DOUBLE PRECISION Power_a7
95         DOUBLE PRECISION Power_a8
```

```
96        DOUBLE PRECISION Power_a9
97        DOUBLE PRECISION Power_a10
98        DOUBLE PRECISION Power_a11
99        DOUBLE PRECISION Power_a12
100       DOUBLE PRECISION Power_a13
101       DOUBLE PRECISION Power_a14
102       DOUBLE PRECISION Power_a15
103       DOUBLE PRECISION Power_a16
104       DOUBLE PRECISION Power_a17
105       DOUBLE PRECISION Power_a18
106       DOUBLE PRECISION Power_a19
107       DOUBLE PRECISION Power_a20
108       DOUBLE PRECISION Power_a21
109       DOUBLE PRECISION Power_a22
110       DOUBLE PRECISION Power_a23
111       DOUBLE PRECISION Power_a24
112       DOUBLE PRECISION Power_a25
113       DOUBLE PRECISION Power_a26
114       DOUBLE PRECISION Power_a27
115       DOUBLE PRECISION Power_a28
116       DOUBLE PRECISION Power_a29
117       DOUBLE PRECISION Power_a30
118       DOUBLE PRECISION Power_a31
119       DOUBLE PRECISION Power_a32
120       DOUBLE PRECISION Power_a33
121       DOUBLE PRECISION Power_a34
122       DOUBLE PRECISION Power_a35
123       DOUBLE PRECISION Power_a36
124       DOUBLE PRECISION Power_a37
125       DOUBLE PRECISION Power_a38
126       DOUBLE PRECISION Power_a39
127       DOUBLE PRECISION Power_a40
128       DOUBLE PRECISION Power_a41
129       DOUBLE PRECISION Power_a42
130       DOUBLE PRECISION Power_a43
131       DOUBLE PRECISION Power_a44
132       DOUBLE PRECISION Power_a45
133       DOUBLE PRECISION Power_a46
134       DOUBLE PRECISION Power_a47
135       DOUBLE PRECISION Power_a48
136       DOUBLE PRECISION Power_a49
137       DOUBLE PRECISION Power_a50
138       DOUBLE PRECISION Power_a51
139       DOUBLE PRECISION Power_a52
140       DOUBLE PRECISION Power_a53
141       DOUBLE PRECISION Power_a54
142       DOUBLE PRECISION Power_a55
143       DOUBLE PRECISION Power_a56
144
145   ! Q Source fit
146       DOUBLE PRECISION QSource
147       DOUBLE PRECISION QSource_a1
```

```
148         DOUBLE PRECISION QSource_a2
149         DOUBLE PRECISION QSource_a3
150         DOUBLE PRECISION QSource_a4
151         DOUBLE PRECISION QSource_a5
152         DOUBLE PRECISION QSource_a6
153         DOUBLE PRECISION QSource_a7
154         DOUBLE PRECISION QSource_a8
155         DOUBLE PRECISION QSource_a9
156         DOUBLE PRECISION QSource_a10
157         DOUBLE PRECISION QSource_a11
158         DOUBLE PRECISION QSource_a12
159         DOUBLE PRECISION QSource_a13
160         DOUBLE PRECISION QSource_a14
161         DOUBLE PRECISION QSource_a15
162         DOUBLE PRECISION QSource_a16
163         DOUBLE PRECISION QSource_a17
164         DOUBLE PRECISION QSource_a18
165         DOUBLE PRECISION QSource_a19
166         DOUBLE PRECISION QSource_a20
167         DOUBLE PRECISION QSource_a21
168         DOUBLE PRECISION QSource_a22
169         DOUBLE PRECISION QSource_a23
170         DOUBLE PRECISION QSource_a24
171         DOUBLE PRECISION QSource_a25
172         DOUBLE PRECISION QSource_a26
173         DOUBLE PRECISION QSource_a27
174         DOUBLE PRECISION QSource_a28
175         DOUBLE PRECISION QSource_a29
176         DOUBLE PRECISION QSource_a30
177         DOUBLE PRECISION QSource_a31
178         DOUBLE PRECISION QSource_a32
179         DOUBLE PRECISION QSource_a33
180         DOUBLE PRECISION QSource_a34
181         DOUBLE PRECISION QSource_a35
182         DOUBLE PRECISION QSource_a36
183         DOUBLE PRECISION QSource_a37
184         DOUBLE PRECISION QSource_a38
185         DOUBLE PRECISION QSource_a39
186         DOUBLE PRECISION QSource_a40
187         DOUBLE PRECISION QSource_a41
188         DOUBLE PRECISION QSource_a42
189         DOUBLE PRECISION QSource_a43
190         DOUBLE PRECISION QSource_a44
191         DOUBLE PRECISION QSource_a45
192         DOUBLE PRECISION QSource_a46
193         DOUBLE PRECISION QSource_a47
194         DOUBLE PRECISION QSource_a48
195         DOUBLE PRECISION QSource_a49
196         DOUBLE PRECISION QSource_a50
197         DOUBLE PRECISION QSource_a51
198         DOUBLE PRECISION QSource_a52
199         DOUBLE PRECISION QSource_a53
```

```
200         DOUBLE PRECISION QSource_a54
201         DOUBLE PRECISION QSource_a55
202         DOUBLE PRECISION QSource_a56
203
204
205     ! Load Q Fit
206         DOUBLE PRECISION Qload
207         DOUBLE PRECISION Qload_a1
208         DOUBLE PRECISION Qload_a2
209         DOUBLE PRECISION Qload_a3
210         DOUBLE PRECISION Qload_a4
211         DOUBLE PRECISION Qload_a5
212         DOUBLE PRECISION Qload_a6
213         DOUBLE PRECISION Qload_a7
214         DOUBLE PRECISION Qload_a8
215         DOUBLE PRECISION Qload_a9
216         DOUBLE PRECISION Qload_a10
217         DOUBLE PRECISION Qload_a11
218         DOUBLE PRECISION Qload_a12
219         DOUBLE PRECISION Qload_a13
220         DOUBLE PRECISION Qload_a14
221         DOUBLE PRECISION Qload_a15
222         DOUBLE PRECISION Qload_a16
223         DOUBLE PRECISION Qload_a17
224         DOUBLE PRECISION Qload_a18
225         DOUBLE PRECISION Qload_a19
226         DOUBLE PRECISION Qload_a20
227         DOUBLE PRECISION Qload_a21
228         DOUBLE PRECISION Qload_a22
229         DOUBLE PRECISION Qload_a23
230         DOUBLE PRECISION Qload_a24
231         DOUBLE PRECISION Qload_a25
232         DOUBLE PRECISION Qload_a26
233         DOUBLE PRECISION Qload_a27
234         DOUBLE PRECISION Qload_a28
235         DOUBLE PRECISION Qload_a29
236         DOUBLE PRECISION Qload_a30
237         DOUBLE PRECISION Qload_a31
238         DOUBLE PRECISION Qload_a32
239         DOUBLE PRECISION Qload_a33
240         DOUBLE PRECISION Qload_a34
241         DOUBLE PRECISION Qload_a35
242         DOUBLE PRECISION Qload_a36
243         DOUBLE PRECISION Qload_a37
244         DOUBLE PRECISION Qload_a38
245         DOUBLE PRECISION Qload_a39
246         DOUBLE PRECISION Qload_a40
247         DOUBLE PRECISION Qload_a41
248         DOUBLE PRECISION Qload_a42
249         DOUBLE PRECISION Qload_a43
250         DOUBLE PRECISION Qload_a44
251         DOUBLE PRECISION Qload_a45
```

```
252       DOUBLE PRECISION Qload_a46
253       DOUBLE PRECISION Qload_a47
254       DOUBLE PRECISION Qload_a48
255       DOUBLE PRECISION Qload_a49
256       DOUBLE PRECISION Qload_a50
257       DOUBLE PRECISION Qload_a51
258       DOUBLE PRECISION Qload_a52
259       DOUBLE PRECISION Qload_a53
260       DOUBLE PRECISION Qload_a54
261       DOUBLE PRECISION Qload_a55
262       DOUBLE PRECISION Qload_a56
263
264
265   ! COP
266       DOUBLE PRECISION  COP
267   ! Energy Balance Equation
268       DOUBLE PRECISION EBE
269
270   ! Outlet Temperatures
271       DOUBLE PRECISION TLoadOut
272       DOUBLE PRECISION TSourceOut
273
274
275   !-----------------------------------------------------------------------
276   !Get the Global Trnsys Simulation Variables
277        Time=getSimulationTime()
278        Timestep=getSimulationTimeStep()
279        CurrentUnit = getCurrentUnit()
280        CurrentType = getCurrentType()
281   !-----------------------------------------------------------------------
282
283   !-----------------------------------------------------------------------
284   !Set the Version Number for This Type
285        If(getIsVersionSigningTime()) Then
286          Call SetTypeVersion(17)
287          Return
288        EndIf
289   !-----------------------------------------------------------------------
290
291   !-----------------------------------------------------------------------
292   !Do Any Last Call Manipulations Here
293        If(getIsLastCallofSimulation()) Then
294          Return
295        EndIf
296   !-----------------------------------------------------------------------
297
298   !-----------------------------------------------------------------------
299   !Perform Any "After Convergence" Manipulations That May Be Required
300        If(getIsEndOfTimestep()) Then
301          Return
302        EndIf
303   !-----------------------------------------------------------------------
```

```
304
305   !-----------------------------------------------------------------------
306   !Do All of the "Very First Call of the Simulation Manipulations" Here
307         If(getIsFirstCallofSimulation()) Then
308
309           !Tell the TRNSYS Engine How This Type Works
310           Call SetNumberofParameters(2)    !The number of parameters
311           Call SetNumberofInputs(6)        !The number of inputs
312           Call SetNumberofDerivatives(0)   !The number of derivatives
313           Call SetNumberofOutputs(11)     !The number of outputs
314           Call SetIterationMode(1)     !An indicator for the iteration mode
315           Call SetNumberStoredVariables(0,0)  !The number of static variables
316           Call SetNumberofDiscreteControls(0)  !The number of discrete control
317
318           Return
319
320         EndIf
321   !-----------------------------------------------------------------------
322
323   !-----------------------------------------------------------------------
324   !Do the First Timestep Manipulations Here - No Iterations at Intial Time
325         If (getIsStartTime()) Then
326         CpSource = getParameterValue(1)
327         CpLoad = getParameterValue(2)
328         HPCall = GetInputValue(1)
329         nCompressor = GetInputValue(2)
330         SourceFlowRate = GetInputValue(3)
331         LoadFlowRate = GetInputValue(4)
332         TSourceIn = GetInputValue(5)
333         TLoadIn = GetInputValue(6)
334
335
336      !Check the Parameters for Problems (#,ErrorType,Text)
337      !Sample Code: If( PAR1 <= 0.) Call FoundBadParameter(1,'Fatal',…
338
339      !Set the Initial Values of the Outputs (#,Value)
340           Call SetOutputValue(1, 0) ! SourceFlowRate
341           Call SetOutputValue(2, 0) ! LoadFlowRate
342           Call SetOutputValue(3, 0) ! QSource
343           Call SetOutputValue(4, 0) ! QLoad
344           Call SetOutputValue(5, 0) ! Power
345           Call SetOutputValue(6, 0) ! COP
346           Call SetOutputValue(7, 0) ! TSourceOut
347           Call SetOutputValue(8, 0) ! TLoadOut
348           Call SetOutputValue(9, 0) ! TSourceIn
349           Call SetOutputValue(10, 0) ! TLoadIn
350           Call SetOutputValue(11, 0) ! EBE
351
352
353      !If Needed, Set the Initial Values of the Static Storage Variables …
354      !Sample Code: SetStaticArrayValue(1,0.d0)
355
```

```
356     !If Needed, Set the Initial Values of the Dynamic Storage Variables …
357     !Sample Code: Call SetDynamicArrayValueThisIteration(1,20.d0)
358
359     !If Needed, Set the Initial Values of the Discrete Controllers (#,Value)
360     !Sample Code for Controller 1 Set to Off: Call …
361
362           Return
363
364         EndIf
365   !-----------------------------------------------------------------------
366
367   !-----------------------------------------------------------------------
368   !ReRead the Parameters if Another Unit of This Type Has Been Called Last
369         If(getIsReReadParameters()) Then
370           !Read in the Values of the Parameters from the Input File
371         CpSource = getParameterValue(1)
372         CpLoad = getParameterValue(2)
373
374
375         EndIf
376   !-----------------------------------------------------------------------
377
378   !Read the Inputs
379         HPCall = GetInputValue(1)
380         nCompressor = GetInputValue(2)
381         SourceFlowRate = GetInputValue(3)
382         LoadFlowRate = GetInputValue(4)
383         TSourceIn = GetInputValue(5)
384
385         TLoadIn = GetInputValue(6)
386
387
388       !Check the Inputs for Problems (#,ErrorType,Text)
389       !Sample Code: If( IN1 <= 0.) Call FoundBadInput(1,'Fatal','The …
390
391         If(ErrorFound()) Return
392   !-----------------------------------------------------------------------
393
394   !-----------------------------------------------------------------------
395   !    *** PERFORM ALL THE CALCULATION HERE FOR THIS MODEL. ***
396   !-----------------------------------------------------------------------
397
398       ! Store inputs in case there is a need to default back to them.
399        HPCall_Stor = HPCall
400        nCompressor_Stor = nCompressor
401        SourceFlowRate_Stor = SourceFlowRate
402        LoadFlowRate_Stor = LoadFlowRate
403        TSourceIn_Stor = TSourceIn
404        TLoadIn_Stor = TLoadIn
405
406       ! Power Fit Coefficients--
407        Power_a1 = 0
```

```
        Power_a2 = 0
        Power_a3 = -1815.84554881437
        Power_a4 = 226.858640474168
        Power_a5 = 20.1813297014303
        Power_a6 = 3.03982920401134
        Power_a7 = -0.688204704762801
        Power_a8 = 0.00841948503342864
        Power_a9 = 0.000687709205539901
        Power_a10 = -201.204392931104
        Power_a11 = -1.93718374388179
        Power_a12 = -2.33076413923059
        Power_a13 = -0.438058911144035
        Power_a14 = -0.000361131868722904
        Power_a15 = 0.000511486739734817
        Power_a16 = 0.963386528450312
        Power_a17 = -0.0939306406036504
        Power_a18 = 0.000263287627726379
        Power_a19 = -0.00101664589676955
        Power_a20 = -492.611563891718
        Power_a21 = 0
        Power_a22 = 88.0122385758513
        Power_a23 = 4.43647446520855
        Power_a24 = -3.90589633454556
        Power_a25 = 0.000788399515130142
        Power_a26 = -3.71428484356646
        Power_a27 = 3.70795123280703
        Power_a28 = -0.00309581201244269
        Power_a29 = 0.00102987614106375
        Power_a30 = 50.0692275133784
        Power_a31 = -8.30924188835276
        Power_a32 = -0.00423872431374436
        Power_a33 = 0.0159194417344668
        Power_a34 = -1.82858066609309
        Power_a35 = 458.74057699975
        Power_a36 = -563.647481994907
        Power_a37 = 314.534554069748
        Power_a38 = -3.46469514871893
        Power_a39 = 1.89229925475327
        Power_a40 = 0.0013654517515994
        Power_a41 = -0.72721656801935
        Power_a42 = 2.32280778835198
        Power_a43 = 0.00172262758239966
        Power_a44 = -0.00370220518976682
        Power_a45 = -1.0253055307542
        Power_a46 = -0.337978469125712
        Power_a47 = 0.0028001738584194
        Power_a48 = -0.00983024845516207
        Power_a49 = 0.127865649339589
        Power_a50 = 31.7369814035295
        Power_a51 = -57.3809184292838
        Power_a52 = 0.0131010849202249
        Power_a53 = 0.013750108898426
```

```
460         Power_a54 = 0.0573928379225005
461         Power_a55 = 0.676975847171639
462
463
464       ! QSource Fit Coefficients--
465         QSource_a1 = 0
466         QSource_a2 = 0
467         QSource_a3 = -3032.01059901012
468         QSource_a4 = -4091.7693356965
469         QSource_a5 = 6.02516980683106
470         QSource_a6 = -14.8834938915157
471         QSource_a7 = 12.5911840344802
472         QSource_a8 = 0.414522728104606
473         QSource_a9 = -0.0204090346885222
474         QSource_a10 = 3924.59682597026
475         QSource_a11 = 267.415121460565
476         QSource_a12 = 0.419459594848725
477         QSource_a13 = 1.15389577394557
478         QSource_a14 = -0.638198631227958
479         QSource_a15 = 0.0200992441359098
480         QSource_a16 = -14.3476202418688
481         QSource_a17 = 0.103952041210189
482         QSource_a18 = -0.0209013202773206
483         QSource_a19 = 0.0224331566839627
484         QSource_a20 = 1864.95321276798
485         QSource_a21 = 0
486         QSource_a22 = 240.134691326062
487         QSource_a23 = -11.9664603743353
488         QSource_a24 = 3.1260547384399
489         QSource_a25 = 0.0175933252905686
490         QSource_a26 = -15.0640566999718
491         QSource_a27 = 17.509335924808
492         QSource_a28 = -0.0270591603760394
493         QSource_a29 = 0.0104030561805099
494         QSource_a30 = 464.545403015573
495         QSource_a31 = -627.673536624565
496         QSource_a32 = 0.252086399744295
497         QSource_a33 = -0.00552938594873994
498         QSource_a34 = 4.99870381183502
499         QSource_a35 = 6481.54744383379
500         QSource_a36 = -6711.26251984639
501         QSource_a37 = 498.058016970524
502         QSource_a38 = 8.24117771019545
503         QSource_a39 = -1.88283317141052
504         QSource_a40 = -0.00958711421906944
505         QSource_a41 = 32.6616569782531
506         QSource_a42 = -35.5839661993736
507         QSource_a43 = -0.00679557980040679
508         QSource_a44 = 0.0194056739161573
509         QSource_a45 = -478.876098704607
510         QSource_a46 = 581.074758304514
511         QSource_a47 = 0.255207503932254
```

```
512         QSource_a48 = -0.109244498047632
513         QSource_a49 = -6.25353924211051
514         QSource_a50 = -649.404479428142
515         QSource_a51 = 503.334399872541
516         QSource_a52 = -0.0127706735771298
517         QSource_a53 = -0.373062419587884
518         QSource_a54 = -3.05278487019105
519         QSource_a55 = 11.3527757134112
520
521
522       !       QLoad Fit Coefficients--
523         QLoad_a1 = 0
524         QLoad_a2 = 0
525         QLoad_a3 = -3136.05801206954
526         QLoad_a4 = 3780.06498904686
527         QLoad_a5 = 433.292022966824
528         QLoad_a6 = 6.62797711720898
529
530         QLoad_a7 = -14.2108867473341
531         QLoad_a8 = -0.286339158949
532         QLoad_a9 = 0.0221224839479144
533         QLoad_a10 = -3029.12177955307
534         QLoad_a11 = -862.198697658889
535         QLoad_a12 = 31.4638676057945
536         QLoad_a13 = 0.913055412142552
537         QLoad_a14 = 0.416266266747177
538         QLoad_a15 = -0.0200423782301436
539         QLoad_a16 = 12.3216107685853
540         QLoad_a17 = -0.412875458020366
541         QLoad_a18 = 0.0183635690482504
542         QLoad_a19 = -0.0186750308543504
543         QLoad_a20 = -8545.95210542819
544         QLoad_a21 = 0
545         QLoad_a22 = -9114.82065448515
546         QLoad_a23 = 31.55691862129
547         QLoad_a24 = -17.9739330638227
548         QLoad_a25 = -0.0164293313112548
549         QLoad_a26 = 33.373423225282
550         QLoad_a27 = -12.7371659237513
551         QLoad_a28 = 0.00120923590803361
552         QLoad_a29 = -0.0504398632878211
553         QLoad_a30 = -1416.0666657021
554         QLoad_a31 = 3054.86283240976
555         QLoad_a32 = -0.0371970517028511
556         QLoad_a33 = 0.350298369787819
557         QLoad_a34 = -71.9605465320487
558         QLoad_a35 = -18429.789627595
559         QLoad_a36 = 11567.4302453532
560         QLoad_a37 = 9159.01491396084
561         QLoad_a38 = 16.5150437382452
562         QLoad_a39 = -23.087509894315
563         QLoad_a40 = 0.0137065810205693
```

```
564         QLoad_a41 = -107.418681912314
565         QLoad_a42 = 105.636667801481
566         QLoad_a43 = -0.00226327162233764
567         QLoad_a44 = -0.0146786202435363
568         QLoad_a45 = 3815.85932563418
569         QLoad_a46 = -4040.35572772849
570         QLoad_a47 = -0.428576312485912
571         QLoad_a48 = 0.465752598861653
572         QLoad_a49 = 9.33032649628423
573         QLoad_a50 = 268.639851337517
574         QLoad_a51 = -286.217279768197
575         QLoad_a52 = 0.125110415533302
576         QLoad_a53 = 0.468981036265427
577         QLoad_a54 = 4.60283450099681
578         QLoad_a55 = -7.9280058949683
579
580
581
582         If ((SourceFlowRate == 0) .OR. (LoadFlowRate == 0) .OR. (HPCall == 0)
583     .OR. (CpSource == 0) .OR. (CpLoad == 0) .OR. (nCompressor < 0.50))Then !   .OR.
584     (TSourceIn < 6)) Then
…           Power = 0
…           QSource = 0.0D0
585         QLoad = 0.0D0
586         COP = 0.0D0
587         EBE = 0.0D0
588         TSourceOut = TSourceIn
589         TLoadOut =  TLoadIn
590
591         else
592         ! Cooerce Inputs to limits of model range if necessary
593             If (TSourceIn <10) Then
594             TSourceIn = 10.0D0
595             else if (TSourceIn > 60) Then
596             TSourceIn = 60.0D0
597             end if
598             If (TLoadIn <10) Then
599             TLoadIn = 10.0D0
600             else if (TSourceIn > 60) Then
601             TLoadIn = 60.0D0
602             end if
603             If (SourceFlowRate < 360) Then
604             SourceFlowRate = 360.0D0
605             else if (SourceFlowRate > 645) Then
606             SourceFlowRate = 645.0D0
607             end if
608             If (LoadFlowRate < 360) Then
609             LoadFlowRate = 360.0D0
610             else if (LoadFlowRate > 645) Then
611             LoadFlowRate = 645.0D0
612             end if
613
```

```
614         !--- Convert Temperatures into Kelvin
615
616         TSourceIn = TSourceIn + 273.15     ! Convert from C to K
617         TLoadIn = TLoadIn + 273.15       ! Convert from C to K
618
619         ! Convert Flow Rates into kg/min
620         SourceFlowRate = SourceFlowRate / 60
621         LoadFlowRate = LoadFlowRate / 60
622
623         ! Evaluate Power
624         Power =
625   power_a1*nCompressor**1+Power_a2*nCompressor**2+Power_a3*nCompressor**3+Power_
626   a4*TLoadIn**1+Power_a5*TLoadIn**1*nCompressor**1+Power_a6*TLoadIn**1*nCompress
…     or**2+Power_a7*TLoadIn**2+Power_a8*TLoadIn**2*nCompressor**1+Power_a9*TLoadIn*
      *3+Power_a10*TSourceIn**1+Power_a11*TSourceIn**1*nCompressor**1+Power_a12*TSou
      rceIn**1*nCompressor**2+Power_a13*TSourceIn**1*TLoadIn**1+Power_a14*TSourceIn*
      *1*TLoadIn**1*nCompressor**1+Power_a15*TSourceIn**1*TLoadIn**2+Power_a16*TSour
      ceIn**2+Power_a17*TSourceIn**2*nCompressor**1+Power_a18*TSourceIn**2*TLoadIn**
      1+Power_a19*TSourceIn**3+Power_a20*LoadFlowRate**1+Power_a21*LoadFlowRate**1*n
      Compressor**1+Power_a22*LoadFlowRate**1*nCompressor**2+Power_a23*LoadFlowRate*
      *1*TLoadIn**1+Power_a24*LoadFlowRate**1*TLoadIn**1*nCompressor**1+Power_a25*Lo
      adFlowRate**1*TLoadIn**2+Power_a26*LoadFlowRate**1*TSourceIn**1+ ...

      Power_a27*LoadFlowRate**1*TSourceIn**1*nCompressor**1+Power_a28*LoadFlowRate**
      1*TSourceIn**1*TLoadIn**1+Power_a29*LoadFlowRate**1*TSourceIn**2+Power_a30*Loa
      dFlowRate**2+Power_a31*LoadFlowRate**2*nCompressor**1+Power_a32*LoadFlowRate**
      2*TLoadIn**1+Power_a33*LoadFlowRate**2*TSourceIn**1+Power_a34*LoadFlowRate**3+
      Power_a35*SourceFlowRate**1+Power_a36*SourceFlowRate**1*nCompressor**1+Power_a
      37*SourceFlowRate**1*nCompressor**2+Power_a38*SourceFlowRate**1*TLoadIn**1+Pow
      er_a39*SourceFlowRate**1*TLoadIn**1*nCompressor**1+Power_a40*SourceFlowRate**1
      *TLoadIn**2+Power_a41*SourceFlowRate**1*TSourceIn**1+Power_a42*SourceFlowRate*
      *1*TSourceIn**1*nCompressor**1+Power_a43*SourceFlowRate**1*TSourceIn**1*TLoadI
      n**1+Power_a44*SourceFlowRate**1*TSourceIn**2+Power_a45*SourceFlowRate**1*Load
      FlowRate**1+Power_a46*SourceFlowRate**1*LoadFlowRate**1*nCompressor**1+Power_a
      47*SourceFlowRate**1*LoadFlowRate**1*TLoadIn**1+Power_a48*SourceFlowRate**1*Lo
      adFlowRate**1*TSourceIn**1+Power_a49*SourceFlowRate**1*LoadFlowRate**2+Power_a
      50*SourceFlowRate**2+Power_a51*SourceFlowRate**2*nCompressor**1+Power_a52*Sour
      ceFlowRate**2*TLoadIn**1+Power_a53*SourceFlowRate**2*TSourceIn**1+Power_a54*So
      urceFlowRate**2*LoadFlowRate**1+Power_a55*SourceFlowRate**3

            ! Evaluate QSource
            QSource =
627   QSource_a1*nCompressor**1+QSource_a2*nCompressor**2+QSource_a3*nCompressor**3+
628   QSource_a4*TLoadIn**1+QSource_a5*TLoadIn**1*nCompressor**1+QSource_a6*TLoadIn*
629   *1*nCompressor**2+QSource_a7*TLoadIn**2+QSource_a8*TLoadIn**2*nCompressor**1+Q
…     Source_a9*TLoadIn**3+QSource_a10*TSourceIn**1+QSource_a11*TSourceIn**1*nCompre
      ssor**1+QSource_a12*TSourceIn**1*nCompressor**2+QSource_a13*TSourceIn**1*TLoad
      In**1+QSource_a14*TSourceIn**1*TLoadIn**1*nCompressor**1+QSource_a15*TSourceIn
      **1*TLoadIn**2+QSource_a16*TSourceIn**2+QSource_a17*TSourceIn**2*nCompressor**
      1+QSource_a18*TSourceIn**2*TLoadIn**1+QSource_a19*TSourceIn**3+QSource_a20*Loa
      dFlowRate**1+QSource_a21*LoadFlowRate**1*nCompressor**1+QSource_a22*LoadFlowRa
      te**1*nCompressor**2+QSource_a23*LoadFlowRate**1*TLoadIn**1+QSource_a24*LoadFl
```

```
owRate**1*TLoadIn**1*nCompressor**1+QSource_a25*LoadFlowRate**1*TLoadIn**2+QSo
urce_a26*LoadFlowRate**1*TSourceIn**1+QSource_a27*LoadFlowRate**1*TSourceIn**1
*nCompressor**1+QSource_a28*LoadFlowRate**1*TSourceIn**1*TLoadIn**1+QSource_a2
9*LoadFlowRate**1*TSourceIn**2+QSource_a30*LoadFlowRate**2+QSource_a31*LoadFlo
wRate**2*nCompressor**1+QSource_a32*LoadFlowRate**2*TLoadIn**1+QSource_a33*Loa
dFlowRate**2*TSourceIn**1+QSource_a34*LoadFlowRate**3+QSource_a35*SourceFlowRa
te**1+QSource_a36*SourceFlowRate**1*nCompressor**1+QSource_a37*SourceFlowRate*
*1*nCompressor**2+QSource_a38*SourceFlowRate**1*TLoadIn**1+QSource_a39*SourceF
lowRate**1*TLoadIn**1*nCompressor**1+QSource_a40*SourceFlowRate**1*TLoadIn**2+
QSource_a41*SourceFlowRate**1*TSourceIn**1+QSource_a42*SourceFlowRate**1*TSour
ceIn**1*nCompressor**1+QSource_a43*SourceFlowRate**1*TSourceIn**1*TLoadIn**1+Q
Source_a44*SourceFlowRate**1*TSourceIn**2+QSource_a45*SourceFlowRate**1*LoadFl
owRate**1+QSource_a46*SourceFlowRate**1*LoadFlowRate**1*nCompressor**1+QSource
_a47*SourceFlowRate**1*LoadFlowRate**1*TLoadIn**1+QSource_a48*SourceFlowRate**
1*LoadFlowRate**1*TSourceIn**1+QSource_a49*SourceFlowRate**1*LoadFlowRate**2+Q
Source_a50*SourceFlowRate**2+QSource_a51*SourceFlowRate**2*nCompressor**1+QSou
rce_a52*SourceFlowRate**2*TLoadIn**1+QSource_a53*SourceFlowRate**2*TSourceIn**
1+QSource_a54*SourceFlowRate**2*LoadFlowRate**1+QSource_a55*SourceFlowRate**3
    ! Evaluate QLoad
    QLoad =
QLoad_a1*nCompressor**1+QLoad_a2*nCompressor**2+QLoad_a3*nCompressor**3+QLoad_
a4*TLoadIn**1+QLoad_a5*TLoadIn**1*nCompressor**1+QLoad_a6*TLoadIn**1*nCompress
or**2+QLoad_a7*TLoadIn**2+QLoad_a8*TLoadIn**2*nCompressor**1+QLoad_a9*TLoadIn*
*3+QLoad_a10*TSourceIn**1+QLoad_a11*TSourceIn**1*nCompressor**1+QLoad_a12*TSou
rceIn**1*nCompressor**2+QLoad_a13*TSourceIn**1*TLoadIn**1+QLoad_a14*TSourceIn*
*1*TLoadIn**1*nCompressor**1+QLoad_a15*TSourceIn**1*TLoadIn**2+QLoad_a16*TSour
ceIn**2+QLoad_a17*TSourceIn**2*nCompressor**1+QLoad_a18*TSourceIn**2*TLoadIn**
1+QLoad_a19*TSourceIn**3+QLoad_a20*LoadFlowRate**1+QLoad_a21*LoadFlowRate**1*n
Compressor**1+QLoad_a22*LoadFlowRate**1*nCompressor**2+QLoad_a23*LoadFlowRate*
*1*TLoadIn**1+QLoad_a24*LoadFlowRate**1*TLoadIn**1*nCompressor**1+QLoad_a25*Lo
adFlowRate**1*TLoadIn**2+QLoad_a26*LoadFlowRate**1*TSourceIn**1+QLoad_a27*Load
FlowRate**1*TSourceIn**1*nCompressor**1+QLoad_a28*LoadFlowRate**1*TSourceIn**1
*TLoadIn**1+QLoad_a29*LoadFlowRate**1*TSourceIn**2+QLoad_a30*LoadFlowRate**2+Q
Load_a31*LoadFlowRate**2*nCompressor**1+QLoad_a32*LoadFlowRate**2*TLoadIn**1+Q
Load_a33*LoadFlowRate**2*TSourceIn**1+QLoad_a34*LoadFlowRate**3+QLoad_a35*Sour
ceFlowRate**1+QLoad_a36*SourceFlowRate**1*nCompressor**1+QLoad_a37*SourceFlowR
ate**1*nCompressor**2+QLoad_a38*SourceFlowRate**1*TLoadIn**1+QLoad_a39*SourceF
lowRate**1*TLoadIn**1*nCompressor**1+QLoad_a40*SourceFlowRate**1*TLoadIn**2+QL
oad_a41*SourceFlowRate**1*TSourceIn**1+QLoad_a42*SourceFlowRate**1*TSourceIn**
1*nCompressor**1+QLoad_a43*SourceFlowRate**1*TSourceIn**1*TLoadIn**1+QLoad_a44
*SourceFlowRate**1*TSourceIn**2+QLoad_a45*SourceFlowRate**1*LoadFlowRate**1+QL
oad_a46*SourceFlowRate**1*LoadFlowRate**1*nCompressor**1+QLoad_a47*SourceFlowR
ate**1*LoadFlowRate**1*TLoadIn**1+QLoad_a48*SourceFlowRate**1*LoadFlowRate**1*
TSourceIn**1+QLoad_a49*SourceFlowRate**1*LoadFlowRate**2+QLoad_a50*SourceFlowR
ate**2+QLoad_a51*SourceFlowRate**2*nCompressor**1+QLoad_a52*SourceFlowRate**2*
TLoadIn**1+QLoad_a53*SourceFlowRate**2*TSourceIn**1+QLoad_a54*SourceFlowRate**
2*LoadFlowRate**1+QLoad_a55*SourceFlowRate**3
    ! Evaluate power
    COP = QLoad / Power

    ! Evaluate EBE
    EBE = QSource + Power - QLoad
```

```
634
635     ! Convert FlowRates to kg/s
636     SourceFlowRate = SourceFlowRate / 60
637     LoadFlowRate = LoadFlowRate / 60
638
639     ! Evaluate Outlet Temperatures
640     TSourceOut = TSourceIn_Stor + QSource/(SourceFlowRate*CpSource*1000)
641     TLoadOut = TLoadIn_Stor + QLoad/(LoadFlowRate*CpLoad*1000)
642
643     ! Convert Temperatures back from K to C
644     !TSourceOut = TSourceOut
645     !TLoadOut = TLoadOut
646     TSourceIn = TSourceIn_Stor
647     TLoadIn = TLoadIn_Stor
648
649     ! Replace Flowrates with the original stored florates to convert to L/h
650     SourceFlowRate = SourceFlowRate_Stor
651     LoadFlowRate = LoadFlowRate_Stor
652
653      end if
654
655       if ((TSourceOut <= 5) .OR. (TLoadOut > 70)) Then
656
657       Power = 0
658       QSource = 0.0D0
659       QLoad = 0.0D0
660       COP = 0.0D0
661       EBE = 0.0D0
662       TSourceOut = TSourceIn_Stor
663       TLoadOut =  TLoadIn_Stor
664       end if
665
666 !-------------------------------------------------------------------------------
667
668 !-------------------------------------------------------------------------------
669 !Set the Outputs from this Model (#,Value)
670         Call SetOutputValue(1, SourceFlowRate) ! SourceFlowRate
671         Call SetOutputValue(2, LoadFlowRate) ! LoadFlowRate
672         Call SetOutputValue(3, QSource) ! QSource
673         Call SetOutputValue(4, QLoad) ! QLoad
674         Call SetOutputValue(5, Power) ! Power
675         Call SetOutputValue(6, COP) ! COP
676         Call SetOutputValue(7, TSourceOut) ! TSourceOut
677         Call SetOutputValue(8, TLoadOut) ! TLoadOut
678         Call SetOutputValue(9, TSourceIn) ! TSourceIn
679         Call SetOutputValue(10, TLoadIn) ! TLoadIn
680         Call SetOutputValue(11, EBE) ! EBE
681
682 !-------------------------------------------------------------------------------
683       Return
684       End
685 !-------------------------------------------------------------------------------
```

Appendix D

Listing of MATLAB Functions and Scripts

D.1 Conventional "Temperature-Based" MATLAB Control Script

The following MATLAB script "ConventionalControllerParametrics_SelectAvailableOperating Modes.m" is called by TRNSYS Type 155 once per time-step of the simulations referred to in Chapter 5.

```
% ConventionalControllerParametrics_SelectAvailableOperatingModes.m
% ------------------------------------------------------------------------
%
% A custom controller for Multi-Mode SAHP that works on Temperature Differentials (M-file called by TRNSYS type 155)
%
% Data passed from / to TRNSYS
% ----------------------------
%
% trnTime (1x1)        : simulation time
% trnInfo (15x1)       : TRNSYS info array
% trnInputs (nIx1)     : TRNSYS inputs
% trnStartTime (1x1)   : TRNSYS Simulation Start time
% trnStopTime (1x1)    : TRNSYS Simulation Stop time
% trnTimeStep (1x1)    : TRNSYS Simulation time step
% mFileErrorCode (1x1) : Error code for this m-file. It is set to 1 by TRNSYS and the m-file should set it to 0 at the
%                        end to indicate that the call was successful. Any non-zero value will stop the simulation
% trnOutputs (nOx1)    : TRNSYS outputs
%
%
% Notes:
% ------
%
% You can use the values of trnInfo(7), trnInfo(8) and trnInfo(13) to identify the call (e.g. first iteration, etc.)
% Real-time controllers (callingMode = 10) will only be called once per time step with trnInfo(13) = 1 (after convergence)
%
% The number of inputs is given by trnInfo(3)
% The number of expected outputs is given by trnInfo(6)
```

```
28 % WARNING: if multiple units of Type 155 are used, the variables passed
   from/to TRNSYS will be sized according to
29 %          the maximum required by all units. You should cope with that
   by only using the part of the arrays that is
30 %          really used by the current m-File. Example: use "nI =
   trnInfo(3); myInputs = trnInputs(1:nI);"
31 %                                                    rather than
   "MyInputs = trnInputs;"
32 %          Please also note that all m-files share the same workspace in
   Matlab (they are "scripts", not "functions") so
33 %          variables like trnInfo, trnTime, etc. will be overwritten at
   each call.
34 %
35 % ------------------------------------------------------------------------
   ------------------------------------------
36 % The component is  iterative
37 %
38 % trnInputs
39 % ---------
40 %
41 % trnInputs(1) : To_COL, degree C, collector outlet temperature
42 % trnInputs(2) : T_DHW_BOT, degree C, DHW Tank Temperature at Bottom
43 % trnInputs(3) : T_DHW_TOP, degree C, DHW Tank Temperature at Top
44 % trnInputs(4) : HP_Source_Flowrate, kg/h, Flow rate on Source side of HP
45 % trnInputs(5) : HP_Load_Flowrate, kg/h, Flow rate on load side of HP
46 % trnInputs(6) : HP_Q_Source, W, Heat Transfer rate on source side of HP
47 % trnInputs(7) : HP_Q_Load, W, Heat Transfer rate on Load side of HP
48 % trnInputs(8) : HP_Power, W, HP Compressor Power Consumption
49 % trnInputs(9) : HP_COP, [dim], HP  Coefficient of Performance
50 % trnInputs(10) : HP_TSource_Out, degree C, HP Souce Side outlet
   temperature
51 % trnInputs(11) : HP_TLoad_Out, degree C, HP Load Side outlet temperature
52 % trnInputs(12) : HP_TSource_In, degree C, HP Souce Side inlet
   temperature
53 % trnInputs(13) : HP_TLoad_In, degree C, HP Load Side inlet temperature
54 % trnInputs(14) : HP_EBE, W, HP Energy balance Error
55 % trnInputs(15) : STC_Useful_EGain, kJ/h, Useful energy gain at STC
56 % trnInputs(16) : DHW_EStore_Rate, kJ/h, DHW Tank Energy Storage rate
57 % trnInputs(17) : T_DHW_MID, degree C, DHW Tank Temperature Middle
58 % trnInputs(18) : T_DHW_AVG, degree C, DHW Tank Temperature Average
59 % trnInputs(19) : DrawTemp, degree C, Temperature of DHW draw
60 % trnInputs(20) : DrawFlow, kg/h, Flow rate of DHW draw
61 % trnInputs(21) : MainsTemp, degree C, Temperature incoming Mains Water
62 % trnInputs(22) : HX_TSourceIn, degree C, HP Souce Side inlet temperature
```

```
% trnInputs(23) : HX_TSourceOut, degree C, HP Source Side outlet temperature
% trnInputs(24) : HX_SourceFlowRate, kg/h, HP Souce Side Flow rate
% trnInputs(25) : HX_T_LoadIn, degree C, HP Load Side inlet temperature
% trnInputs(26) : HX_T_LoadOut, degree C, HP Load Side outlet temperature
% trnInputs(27) : HX_LoadFlowRate, kg/h, HP load Side Flow rate
% trnInputs(28) : DHW_E_Delivery_Draw, kJ/h, DHW Draw Side Energy Delivery rate
% trnInputs(29) : DHW_E_Delivery_Charge, kJ/h, DHW Charge Side Energy Delivery rate
% trnInputs(30) : DHW_AUX_Heating, kJ/h, DHW Auxiliary Heating rate
% trnInputs(31) : T_DHW_AUX_Set, C, Setpoint Temperature for the Aux Charging the DHW Tank
% trnInputs(32) : T_Deadband = ; C, Deadband for DHW Setpoint
% trnInputs(33) : T_HX_MinDiff, C, Minimum Dt for HX operation
% trnInputs(34) : T_HP_Cutin, C, Cutin Source side temperature for the HP
% trnInputs(35) : SunriseSetDelay = h, the number of hours after/before sunrise/set to activate the HX or HP controls
% trnInputs(36) : T_DHW_Thermostat, degree C, DHW  Thermostat Temperature
% trnInputs(37) : AUX_Mode, The mode of operation for the Auxiliary heater.
% trnInputs(38) : T_DHW_Solar_Set, The temperature setpoint for solar charging the DHW Tank.
% trnInputs(39) : AvailableModes, 0=No Restriction, 1= HX Only, 2=HP Only, 3=No Solar Modes.

%
% trnOutputs
% ----------
%
% trnOutputs(1) : Controller_Mode, Mode
% trnOutputs(2) : Pump1_Signal, Pump Signal (Source Side)
% trnOutputs(3) : Pump2_Signal, Pump Signal (Load Side)
% trnOutputs(4) : Diverter1_Signal,Diverter Signal (Source Side)
% trnOutputs(5) : Diverter3_Signal,Diverter Signal (Load Side)
% trnOutputs(6) : HP_CALL, HP Call signal, 1=on, 0=0ff;
% trnOutputs(7) : N_Compressor, HP Compressor Speed, 0=0% or 0 RPM, 1=100%
% or 3600 RPM
% trnOutputs(8) : AuxHeat_Call, 1=Enable DHW Element, 0=Disable DHW Element;

%
% JCH, July 13, 2022
```

```
97  % ------------------------------------------------------------------------
98
99
100 % TRNSYS sets mFileErrorCode = 1 at the beginning of the M-File for error
    detection
101 % This file increments mFileErrorCode at different places. If an error
    occurs in the m-file the last succesful step will
102 % be indicated by mFileErrorCode, which is displayed in the TRNSYS error
    message
103 % At the very end, the m-file sets mFileErrorCode to 0 to indicate that
    everything was OK
104
105 mFileErrorCode = 100;    % Beginning of the m-file
106
107 % --- Controller Parameters-----------------------------------------------
108 % ------------------------------------------------------------------------
109 T_DHW_AUX_Set =trnInputs(31); %50; % Degree C (DHW Tank Setpoint
    Temperature)
110 T_Deadband =trnInputs(32); %2; % Degree C (DHW Tank Deadband)
111 T_HX_MinDiff = trnInputs(33); %2; % Degree C (Minimum Dt for HX
    operation)
112 T_HP_Cutin = trnInputs(34); %10; % Degree C (Cutin Source side
    temperature for the HP)
113 SunriseSetDelay = trnInputs(35); %4;% h, the number of hours after/before
    sunrise/set to activate the HX or HP controls
114 AUX_Mode = trnInputs(37); % 0=Heater locked out during daylight hours, 1,
    Heater locked out when HX and HP are on, 2, Heater locked out when HP is
    on, 3, Heater Never locked out.
115 T_DHW_Solar_Set = trnInputs(38); % T_DHW_Solar_Set, The temperature
    setpoint for solar charging the DHW Tank.
116 AvailableModes = trnInputs(39);  %AvailableModes, 0=No Restriction, 1= HX
    Only, 2=HP Only.
117
118
119 t_timeout =  0.25; % hour (Timeout for next control decision)
    SunriseTimes = [7.709980987 31.71044677 55.71017753 79.70917327
    103.7074347 127.7049633 151.7017612 175.6978313 199.693177  223.6878025
    247.6817125 271.6749123 295.6674078 319.6592053 343.6503117 367.6407344
    391.630481  415.6195598 439.6079794 463.5957486 487.5828767 511.5693733
    535.555248  559.540511  583.5251726 607.5092433 631.4927337 655.4756547
    679.4580173 703.4398325 727.4211117 751.401866  775.3821068 799.3618456
120 823.3410939 847.3198632 871.298165  895.2760108 919.2534122 943.2303806
    967.2069277 991.1830648 1015.158803 1039.134155 1063.10913  1087.083741
    1111.057999 1135.031915 1159.005499 1182.978764 1206.951719 1230.924376
    1254.896745 1278.868837 1302.840663 1326.812233 1350.783557 1374.754646
    1398.72551  1422.696159 1446.666604 1470.636853 1494.606917 1518.576806
    1542.546529 1566.516096 1590.485516 1614.454799 1638.423954 1662.39299
    1686.361916 1710.330742 1734.299477 1758.268129 1782.236708 1806.205222
```

```
1830.17368  1854.142091 1878.110464 1902.078806 1926.047128 1950.015437
1973.983742 1997.952051 2021.920373 2045.888718 2069.857092 2093.825505
2117.793964 2141.76248  2165.73106  2189.699712 2213.668446 2237.63727
2261.606192 2285.575221 2309.544366 2333.513636 2357.483039 2381.452584
2405.422281 2429.392138 2453.362163 2477.332367 2501.302758 2525.273345
2549.244138 2573.215147 2597.186379 2621.157846 2645.129555 2669.101518
2693.073744 2717.046242 2741.019022 2764.992095 2788.965469 2812.939156
2836.913164 2860.887505 2884.862188 2908.837224 2932.812623 2956.788394
2980.764549 3004.741098 3028.71805  3052.695417 3076.673208 3100.651434
3124.630106 3148.609232 3172.588824 3196.568891 3220.549444 3244.530492
3268.512045 3292.494113 3316.476705 3340.459831 3364.443499 3388.427719
3412.4125   3436.397849 3460.383776 3484.370288 3508.357392 3532.345097
3556.333408 3580.322334 3604.311879 3628.302049 3652.292851 3676.284289
3700.276367 3724.269089 3748.262459 3772.25648  3796.251154 3820.246482
3844.242467 3868.239108 3892.236406 3916.234361 3940.23297  3964.232232
3988.232144 4012.232704 4036.233908 4060.235751 4084.238227 4108.241333
4132.245061 4156.249403 4180.254354 4204.259905 4228.266046 4252.27277
4276.280065 4300.287923 4324.296332 4348.30528  4372.314758 4396.324752
4420.33525  4444.346239 4468.357707 4492.369641 4516.382026 4540.394849
4564.408096 4588.421754 4612.435807 4636.450243 4660.465046 4684.480203
4708.495699 4732.51152  4756.527652 4780.544082 4804.560794 4828.577776
4852.595015 4876.612496 4900.630208 4924.648137 4948.66627  4972.684596
4996.703103 5020.721779 5044.740613 5068.759595 5092.778713 5116.797957
5140.817319 5164.836788 5188.856355 5212.876012 5236.895751 5260.915564
5284.935443 5308.955381 5332.975373 5356.995411 5381.015489 5405.035603
5429.055747 5453.075916 5477.096106 5501.116314 5525.136534 5549.156765
5573.177002 5597.197244 5621.217488 5645.237732 5669.257973 5693.278212
5717.298446 5741.318675 5765.338898 5789.359114 5813.379324 5837.399528
5861.419725 5885.439916 5909.460103 5933.480286 5957.500465 5981.520644
6005.540821 6029.561001 6053.581183 6077.601371 6101.621565 6125.641769
6149.661985 6173.682214 6197.702459 6221.722723 6245.743009 6269.763319
6293.783656 6317.804022 6341.824421 6365.844855 6389.865328 6413.885842
6437.906399 6461.927004 6485.947659 6509.968366 6533.98913  6558.009951
6582.030833 6606.051779 6630.072792 6654.093873 6678.115025 6702.13625
6726.157551 6750.178929 6774.200386 6798.221924 6822.243544 6846.265248
6870.287035 6894.308908 6918.330867 6942.352912 6966.375043 6990.397259
7014.419561 7038.441946 7062.464415 7086.486965 7110.509594 7134.532301
7158.555081 7182.577932 7206.600851 7230.623834 7254.646875 7278.66997
7302.693113 7326.7163   7350.739522 7374.762774 7398.786047 7422.809335
7446.832628 7470.855917 7494.879193 7518.902446 7542.925665 7566.948839
7590.971956 7614.995005 7639.017972 7663.040844 7687.063607 7711.086248
7735.108752 7759.131103 7783.153287 7807.175286 7831.197085 7855.218668
7879.240016 7903.261114 7927.281942 7951.302485 7975.322723 7999.342638
8023.362213 8047.381428 8071.400267 8095.418709 8119.436738 8143.454335
8167.471483 8191.488162 8215.504357 8239.520049 8263.535222 8287.549859
8311.563944 8335.577462 8359.590398 8383.602737 8407.614465 8431.625569
8455.636036 8479.645855 8503.655013 8527.663501 8551.671309 8575.678427
8599.684848 8623.690565 8647.695569 8671.699857 8695.703422 8719.706261
8743.70837
121     ];
SunsetTimes = [16.49908377  40.51426113 64.52998696 88.54624482
122 112.5630179 136.5802889 160.5980405 184.6162548 208.6349139 232.6539998
```

	256.6734941 280.6933786 304.7136349 328.7342448 352.7551899 376.7764521 400.7980135 424.8198561 448.8419623 472.8643148 496.8868966 520.9096908 544.932681 568.9558513 592.9791859 617.0026697 641.0262878 665.050026 689.0738705 713.0978077 737.121825 761.1459099 785.1700507 809.194236 833.2184551 857.2426978 881.2669543 905.2912154 929.3154726 953.3397177 977.363943 1001.388142 1025.412307 1049.436432 1073.460513 1097.484544 1121.508519 1145.532436 1169.556289 1193.580077 1217.603795 1241.627442 1265.651015 1289.674512 1313.697933 1337.721275 1361.744538 1385.767722 1409.790827 1433.813852 1457.836799 1481.859667 1505.882457 1529.905171 1553.92781 1577.950375 1601.972868 1625.995291 1650.017646 1674.039934 1698.062158 1722.084321 1746.106424 1770.12847 1794.150462 1818.172401 1842.194292 1866.216136 1890.237937 1914.259696 1938.281417 1962.303102 1986.324753 2010.346374 2034.367968 2058.389535 2082.41108 2106.432604 2130.45411 2154.475599 2178.497075 2202.518538 2226.539991 2250.561435 2274.582872 2298.604303 2322.62573 2346.647154 2370.668575 2394.689994 2418.711412 2442.732829 2466.754244 2490.775659 2514.797071 2538.818481 2562.839888 2586.861289 2610.882685 2634.904072 2658.92545 2682.946814 2706.968163 2730.989494 2755.010803 2779.032086 2803.053339 2827.074558 2851.095737 2875.116872 2899.137957 2923.158986 2947.179953 2971.20085 2995.221671 3019.242408 3043.263053 3067.283597 3091.304033 3115.324351 3139.344541 3163.364593 3187.384498 3211.404244 3235.423821 3259.443217 3283.46242 3307.481419 3331.500201 3355.518754 3379.537065 3403.55512 3427.572906 3451.590411 3475.607619 3499.624517 3523.641092 3547.657328 3571.673212 3595.688729 3619.703865 3643.718606 3667.732938 3691.746846 3715.760317 3739.773336 3763.78589 3787.797965 3811.809548 3835.820626 3859.831186 3883.841217 3907.850705 3931.85964 3955.86801 3979.875805 4003.883015 4027.889629 4051.895639 4075.901036 4099.905812 4123.90996 4147.913473 4171.916344 4195.918569 4219.920142 4243.921058 4267.921314 4291.920908 4315.919836 4339.918097 4363.91569 4387.912614 4411.90887 4435.904458 4459.899379 4483.893635 4507.88723 4531.880165 4555.872444 4579.864072 4603.855053 4627.845392 4651.835094 4675.824165 4699.812613 4723.800442 4747.787662 4771.774277 4795.760298 4819.745731 4843.730585 4867.714869 4891.698591 4915.681761 4939.664387 4963.646479 4987.628047 5011.609101 5035.58965 5059.569703 5083.549273 5107.528367 5131.506998 5155.485174 5179.462907 5203.440206 5227.417082 5251.393546 5275.369607 5299.345277 5323.320564 5347.295481 5371.270037 5395.244241 5419.218106 5443.191639 5467.164853 5491.137756 5515.110358 5539.08267 5563.054702 5587.026462 5610.997961 5634.969208 5658.940213 5682.910985 5706.881533 5730.851867 5754.821996 5778.791929 5802.761675 5826.731243 5850.700642 5874.66988 5898.638967 5922.607911 5946.576721 5970.545405 5994.513972 6018.48243 6042.450788 6066.419053 6090.387236 6114.355342 6138.323382 6162.291363 6186.259294 6210.227182 6234.195037 6258.162865 6282.130676 6306.098478 6330.066278 6354.034086 6378.001909 6401.969756 6425.937636 6449.905556 6473.873525 6497.841551 6521.809644 6545.777811 6569.746062 6593.714405 6617.682848 6641.651402 6665.620075 6689.588875 6713.557813 6737.526896 6761.496136 6785.46554 6809.435119 6833.404881 6857.374838 6881.344998 6905.315372 6929.285969 6953.256799 6977.227873 7001.199202 7025.170794 7049.142662 7073.114815 7097.087264 7121.060021 7145.033095 7169.006497 7192.98024 7216.954334 7240.928789 7264.903618 7288.878832 7312.854442 7336.830459 7360.806895 7384.783762 7408.761071 7432.738833 7456.717061 7480.695766 7504.674959 7528.654652 7552.634856 7576.615584 7600.596846 7624.578653 7648.561018 7672.543951 7696.527464 7720.511566

```
    7744.496269 7768.481584 7792.46752  7816.454088 7840.441298 7864.42916
    7888.417681 7912.406873 7936.396742 7960.387298 7984.378548 8008.370499
    8032.363159 8056.356534 8080.35063  8104.345451 8128.341004 8152.337292
    8176.334319 8200.332087 8224.330599 8248.329856 8272.32986  8296.330609
    8320.332105 8344.334344 8368.337325 8392.341045 8416.345499 8440.350684
    8464.356594 8488.363221 8512.370561 8536.378603 8560.38734  8584.396762
    8608.406859 8632.41762  8656.429033 8680.441086 8704.453765 8728.467058
    8752.480949
123     ]';
124
125
126 HX_Pump_Setting = 0.4651162; % Pump rated at kg/min, at 46.51% Speed = 5
    kg/min
127 HP_Pump_Setting = 1; % Pump rated at kg/min, at 100% Speed
128 HX_Diverter_Setting = 0;
129 HP_Diverter_Setting = 1;
130
131
132 mFileErrorCode = 110;    % After setting parameters
133
134
135 % --- Process Inputs ------------------------------------------------------
    ---------------------------------------------
136 % ------------------------------------------------------------------------
    ---------------------------------------------
137
138 To_COL = trnInputs(1);
139 T_DHW_BOT = trnInputs(2);
140 T_DHW_TOP = trnInputs(3);
141 HP_Source_Flowrate = trnInputs(4);
142 HP_Load_Flowrate = trnInputs(5);
143 HP_Q_Source = 3.6*trnInputs(6);% Convert W to kJ/h
144 HP_Q_Load = 3.6*trnInputs(7); % Convert W to kJ/h
145 HP_Power = 3.6*trnInputs(8);% Convert W to kJ/h
146 HP_COP = trnInputs(9);
147 HP_TSource_Out = trnInputs(10);
148 HP_TLoad_Out = trnInputs(11);
149 HP_TSource_In = trnInputs(12);
150 HP_TLoad_In = trnInputs(13);
151 HP_EBE = 3.6*trnInputs(14); % Convert W to kJ/h
152 STC_Useful_EGain = trnInputs(15);
153 DHW_EStore_Rate = trnInputs(16);
154 T_DHW_MID = trnInputs(17);
```

```
T_DHW_AVG = trnInputs(18);
DrawTemp = trnInputs(19);
DrawFlow = trnInputs(20);
MainsTemp = trnInputs(21);
HX_TSourceIn = trnInputs(22);
HX_TSourceOut = trnInputs(23);
HX_SourceFlowRate = trnInputs(24);
HX_T_LoadIn = trnInputs(25);
HX_T_LoadOut = trnInputs(26);
HX_LoadFlowRate = trnInputs(27);
DHW_E_Dot_Draw = trnInputs(28);
DHW_E_Dot_Charge= -trnInputs(29); % Sign reconciled (Delivery vs. Supply)
DHW_AUX_E_Dot = trnInputs(30);
T_DHW_Thermostat = trnInputs(36);

mFileErrorCode = 120;    % After processing inputs


% --- First call of the simulation: initial time step (no iterations) ---
% ------------------------------------------------------------------------
% (note that Matlab is initialized before this at the info(7) = -1 call,
but the m-file is not called)

if ( (trnInfo(7) == 0) & (trnTime-trnStartTime < 1e-6) )

    % This is the first call (Counter will be incremented later for this
very first call)
    nCall = 0;

    % This is the first time step
    nStep = 1;

    % Initialize history of the variables for plotting at the end of the
simulation
    nTimeSteps = (trnStopTime-trnStartTime)/trnTimeStep + 1;
    history.To_COL= zeros(nTimeSteps,1);
    history.T_DHW_BOT= zeros(nTimeSteps,1);
    history.T_DHW_TOP= zeros(nTimeSteps,1);
    history.HP_Source_Flowrate= zeros(nTimeSteps,1);
    history.HP_Load_Flowrate= zeros(nTimeSteps,1);
```

```
192     history.HP_Q_Source= zeros(nTimeSteps,1);
193     history.HP_Q_Load= zeros(nTimeSteps,1);
194     history.HP_Power= zeros(nTimeSteps,1);
195     history.HP_COP= zeros(nTimeSteps,1);
196     history.HP_TSource_Out= zeros(nTimeSteps,1);
197     history.HP_TLoad_Out= zeros(nTimeSteps,1);
198     history.HP_TSource_In= zeros(nTimeSteps,1);
199     history.HP_TLoad_In= zeros(nTimeSteps,1);
200     history.HP_EBE= zeros(nTimeSteps,1);
201     history.STC_Useful_EGain= zeros(nTimeSteps,1);
202     history.DHW_EStore_Rate= zeros(nTimeSteps,1);
203     history.T_DHW_MID= zeros(nTimeSteps,1);
204     history.T_DHW_AVG= zeros(nTimeSteps,1);
205     history.DrawTemp= zeros(nTimeSteps,1);
206     history.DrawFlow= zeros(nTimeSteps,1);
207     history.MainsTemp= zeros(nTimeSteps,1);
208     history.HX_TSourceIn= zeros(nTimeSteps,1);
209     history.HX_TSourceOut= zeros(nTimeSteps,1);
210     history.HX_SourceFlowRate= zeros(nTimeSteps,1);
211     history.HX_T_LoadIn= zeros(nTimeSteps,1);
212                                 history.HX_T_LoadOut= zeros(nTimeSteps,1);
213     history.HX_LoadFlowRate= zeros(nTimeSteps,1);
214                             history.Controller_Mode = zeros(nTimeSteps,1);
215     history.AuxHeat_Call = zeros(nTimeSteps,1);
216     history.DHW_E_Dot_Draw= zeros(nTimeSteps,1);
217     history.DHW_E_Draw = zeros(nTimeSteps,1); %
218     history.DHW_E_Dot_Charge = zeros(nTimeSteps,1);
219     history.DHW_E_Charge = zeros(nTimeSteps,1);
220     history.DHW_AUX_E_Dot = zeros(nTimeSteps,1);
221     history.DHW_AUX_E = zeros(nTimeSteps,1);
222     history.L_S = zeros(nTimeSteps,1);
223     history.L_A = zeros(nTimeSteps,1);
224     history.HP_Energy = zeros(nTimeSteps,1);
225     history.SolarFraction= zeros(nTimeSteps,1);
226     history.CumulativeSolarFraction = zeros(nTimeSteps,1);
227     history.trnTime = zeros(nTimeSteps,1);
228     history.T_DHW_Thermostat = zeros(nTimeSteps,1);
229     tLastControlChange = 0;
230
```

```
231
232
233     % No return, we will calculate the solar collector performance during
    this call
234     mFileErrorCode = 130;    % After initialization
235
236 end
237
238
239 % --- Very last call of the simulation (after the user clicks "OK"):
240 if ( trnInfo(8) == -1 )
241
242     mFileErrorCode = 1000;
243
244     % Generate an output file to log simulation data.

245 T=table(history.trnTime,mod(history.trnTime,24),history.HP_Source_Flowrat
    e,...
246         history.HP_Load_Flowrate,   history.HP_Q_Source,
    history.HP_Q_Load,...
247         history.HP_Power,   history.HP_COP, history.HP_TSource_Out,...
248         history.HP_TLoad_Out,   history.HP_TSource_In,
    history.HP_TLoad_In,...
249         history.HP_EBE, history.STC_Useful_EGain,
    history.DHW_EStore_Rate,...
250         history.T_DHW_TOP,  history.T_DHW_MID,  history.T_DHW_BOT,
    history.T_DHW_AVG,...
251         history.DrawTemp,   history.DrawFlow,   history.MainsTemp,
    history.HX_TSourceIn,...
252         history.HX_TSourceOut,  history.HX_SourceFlowRate,
    history.HX_T_LoadIn,...
253         history.HX_T_LoadOut,   history.HX_LoadFlowRate,
    history.Controller_Mode,...
254         history.AuxHeat_Call, history.DHW_E_Dot_Draw,
    history.DHW_E_Draw,...
255         history.DHW_E_Dot_Charge, history.DHW_E_Charge,
    history.DHW_AUX_E_Dot, history.DHW_AUX_E, ...
256         history.CumulativeSolarFraction, history.T_DHW_Thermostat);
257     T.Properties.VariableNames = {'SimTime','TOD','HP_Source_Flowrate',
    ...
258         'HP_Load_Flowrate', 'HP_Q_Source',  'HP_Q_Load',    'HP_Power
    [W]', ...
259         'HP_COP',   'HP_TSource_Out',   'HP_TLoad_Out',
    'HP_TSource_In',...
260         'HP_TLoad_In',  'HP_EBE',   'STC_Useful_EGain',
    'DHW_EStore_Rate',...
261         'T_DHW_TOP',    'T_DHW_MID',    'T_DHW_BOT',    'T_DHW_AVG',...
```

```
        'DrawTemp', 'DrawFlow', 'MainsTemp',    'HX_TSourceIn',
'HX_TSourceOut',...
        'HX_SourceFlowRate',    'HX_T_LoadIn',  'HX_T_LoadOut',
'HX_LoadFlowRate',...
        'Controller_Mode', ', AuxHeat_Call', 'DHW Draw Energy Delivery
Rate [kJ/h]',...
        'DHW Draw Energy Delivered [kJ]', 'DHW Charge Energy Delivery
Rate [kJ/h]',...
        'DHW Charge Energy Delivered [kJ]', 'DHW AUX Heating Rate
[kJ/h]',...
        'DHW AUX Heat Delivered[kJ]', 'Cumulative Solar Fraction',
'T_DHW_Thermostat'};
% Calculate Daily Solar Fraction(s) and Daily Totals of Simulation
Variables
numdays=ceil((history.trnTime(end)-history.trnTime(3))/24); % Count the
days in the simulation
if (history.trnTime(end) == history.trnTime(3) &&
length(history.trnTime)>= (8760/trnTimeStep))
    numdays=365;
end
days = [1:1:numdays]'; % An array with one element per day in the
simulation
days=days+floor(history.trnTime(3)/24); % Adjust index to be the Julian
day of each day in the simulation
DailySolarFraction=zeros(length(days),1);
DailyDHWDrawEnergy=zeros(length(days),1);
DailyDHWChargeEnergy= zeros(length(days),1);
DailyDHWAUXEnergy=zeros(length(days),1);
DailyHPEnergy=zeros(length(days),1);
DailyDrawMass=zeros(length(days),1);
DailyInadequateDrawMass=zeros(length(days),1);
                                        DailyNumDraws=zeros(length(days),1);
DailyNumInadequateDraws=zeros(length(days),1);

for i=1:1:length(days)
    dailyBounds=[((i-1)*24/trnTimeStep)+3 ((i)*24/trnTimeStep)+3];
    if i==length(days)
        dailyBounds=[((i-1)*24/trnTimeStep)+3 length(history.trnTime)];
    end

DailySolarFraction(i)=sum(history.L_S(dailyBounds(1):dailyBounds(2)))/(su
m(history.L_A(dailyBounds(1):dailyBounds(2)))+sum(history.L_S(dailyBounds
(1):dailyBounds(2))));

DailyDHWDrawEnergy(i)=trnTimeStep*sum(history.DHW_E_Dot_Draw(dailyBounds(
1):dailyBounds(2)));
```

```
292 DailyDHWChargeEnergy(i)=trnTimeStep*sum(history.DHW_E_Dot_Charge(dailyBou
    nds(1):dailyBounds(2)));
293 DailyDHWAUXEnergy(i)=trnTimeStep*sum(history.DHW_AUX_E_Dot(dailyBounds(1)
    :dailyBounds(2)));
294 DailyHPEnergy(i)=trnTimeStep*sum(history.HP_Power(dailyBounds(1):dailyBou
    nds(2))); % HP Energy Consumption in kJ
295 DailyDrawMass(i)=trnTimeStep*sum(history.DrawFlow(dailyBounds(1):dailyBou
    nds(2)));
296
297
298     %Count the number of draw events
299     TempVector1=history.DrawFlow(dailyBounds(1):dailyBounds(2));
300     TempVector2=zeros(length(TempVector1),1);
301     for j=2:1:length(TempVector1)
302         if (TempVector1(j-1)==0 && TempVector1(j)>0)
303             TempVector2(j)=1;
304         end
305     end
306     DailyNumDraws(i)=sum(TempVector2);
307
308     % Divide out inadequate draw flow rates
309     TempVector1=history.DrawFlow(dailyBounds(1):dailyBounds(2));
310     TempVector2=history.DrawTemp(dailyBounds(1):dailyBounds(2));
311     TempVector3=zeros(length(TempVector1),1);
312     for j=2:1:length(TempVector1)
313         if (TempVector1(j)>0 && TempVector2(j)<=35)
314             TempVector3(j)=TempVector1(j);
315         end
316     end
317     DailyInadequateDrawMass(i)=trnTimeStep*sum(TempVector3);
318
319     TempVector2=zeros(length(TempVector1),1);
320     %Count the number of draw events
321     for j=2:1:length(TempVector1)
322         if (TempVector3(j-1)==0 && TempVector3(j)>0)
323             TempVector2(j)=1;
324         end
325
```

```
    end
    DailyNumInadequateDraws(i)=sum(TempVector2);

end

T2=table(days,DailySolarFraction,DailyDHWDrawEnergy,
DailyDHWChargeEnergy, DailyDHWAUXEnergy,
DailyHPEnergy,DailyDrawMass,DailyNumDraws,DailyInadequateDrawMass,DailyNu
mInadequateDraws);
T2.Properties.VariableNames = {'Day','Solar Fraction','DHW Draw Energy
Delivered [kJ]','DHW Charge Energy Delivered [kJ]','DHW AUX Heat
Delivered[kJ]', 'HP Energy Consumed[kJ]', 'Daily DHW Draw Mass [kg]',
'Daily Number of Draws','Daily inadequate DHW Draw Mass [kg]', 'Daily
Number of Inadequate Draws'};

% Cauculate Monthly Solar Fraction And other totals
monthHourDef=[  1 0
                2 744
                3 1416
                4 2160
                5 2880
                6 3624
                7 4344
                8 5088
                9 5832
                10 6552
                11 7296
                12 8016
                13 8760];
if exist('monthsSimulated','var')
    clear monthsSimulated;
end

j=0;
for i = 1:1:12
    if any(history.trnTime(3:end) >=monthHourDef(i,2) &
history.trnTime(3:end) <=monthHourDef(i+1,2))
        j=j+1;
        monthsSimulated(j)=i;
    end
end

```

```
monthlySolarFraction=zeros(length(monthsSimulated),1);
monthlyDHWDrawEnergy=zeros(length(monthsSimulated),1);
monthlyDHWChargeEnergy=zeros(length(monthsSimulated),1);
monthlyDHWAUXEnergy=zeros(length(monthsSimulated),1);
monthlyHPEnergy=zeros(length(monthsSimulated),1);
monthlyDrawMass=zeros(length(monthsSimulated),1);
monthlyInadequateDrawMass=zeros(length(monthsSimulated),1);
monthlyNumDraws=zeros(length(monthsSimulated),1);
monthlyNumInadequateDraws=zeros(length(monthsSimulated),1);


j=0;
for i=monthsSimulated
    j=j+1;
loweridx=find(history.trnTime>=monthHourDef(i,2),1);
upperidx=find(history.trnTime<=monthHourDef(i+1,2),1,"last");

monthlySolarFraction(j)=sum(history.L_S(loweridx:upperidx))/(sum(history.
L_A(loweridx:upperidx))+sum(history.L_S(loweridx:upperidx)));
monthlyDHWDrawEnergy(j)=trnTimeStep*sum(history.DHW_E_Dot_Draw(loweridx:u
pperidx));
monthlyDHWChargeEnergy(j)=trnTimeStep*sum(history.DHW_E_Dot_Charge(loweri
dx:upperidx));
monthlyDHWAUXEnergy(j)=trnTimeStep*sum(history.DHW_AUX_E_Dot(loweridx:upp
eridx));
monthlyHPEnergy(j)=trnTimeStep*sum(history.HP_Power(loweridx:upperidx));
monthlyDrawMass(j)=trnTimeStep*sum(history.DrawFlow(loweridx:upperidx));

%Count the number of draw events
    TempVector1=history.DrawFlow(loweridx:upperidx);
    TempVector2=zeros(length(TempVector1),1);
    for k=2:1:length(TempVector1)
        if (TempVector1(k-1)==0 && TempVector1(k)>0)
            TempVector2(k)=1;
        end
    end
    monthlyNumDraws(j)=sum(TempVector2);

% Divide out inadequate draw flow rates
    TempVector1=history.DrawFlow(loweridx:upperidx);
    TempVector2=history.DrawTemp(loweridx:upperidx);
```

```
397     TempVector3=zeros(length(TempVector1),1);
398     for k=2:1:length(TempVector1)
399         if (TempVector1(k)>0 && TempVector2(k)<=35)
400             TempVector3(k)=TempVector1(k);
401         end
402     end
403     monthlyInadequateDrawMass(j)=trnTimeStep*sum(TempVector3);
404
405     TempVector2=zeros(length(TempVector1),1);
406     %Count the number of draw events
407     for k=2:1:length(TempVector1)
408         if (TempVector3(k-1)==0 && TempVector3(k)>0)
409             TempVector2(k)=1;
410         end
411     end
412     monthlyNumInadequateDraws(j)=sum(TempVector2);
413
414 end
415
416
417
418 monthsSimulated=monthsSimulated';
419 T3=table(monthsSimulated,monthlySolarFraction, monthlyDHWDrawEnergy,
    monthlyDHWChargeEnergy, monthlyDHWAUXEnergy,
    monthlyHPEnergy,monthlyDrawMass,monthlyNumDraws,monthlyInadequateDrawMass
    ,monthlyNumInadequateDraws);
420 T3.Properties.VariableNames = {'Month','Solar Fraction','DHW Draw Energy
    Delivered [kJ]','DHW Charge Energy Delivered [kJ]','DHW AUX Heat
    Delivered[kJ]', 'HP Energy Consumed[kJ]', 'Monthly DHW Draw Mass [kg]',
    'Monthly Number of Draws','Monthly inadequate DHW Draw Mass [kg]',
    'Monthly Number of Inadequate Draws'};
421
422 % Calculate an Annual Solar Fraction
423
424 % Output datafiles
425     if (~exist('RunNo'))
426         RunNo = 1;
427     end
428 writetable(T,sprintf('SAHP_Simulation_StepByStep_Output_Run_%d.csv',RunNo
    ));
```

```
writetable(T2,sprintf('SAHP_Simulation_Daily_Totalized_Output_Run_%d.csv',RunNo));
writetable(T3,sprintf('SAHP_Simulation_Monthly_Totalized_Output_Run_%d.csv',RunNo));
    RunNo = RunNo+1;
    mFileErrorCode = 0; % Tell TRNSYS that we reached the end of the m-file without errors
    return

end


% --- Post convergence calls: store values ------------------------------------------------
% -----------------------------------------------------------------------------------------

if (trnInfo(13) == 1)



            mFileErrorCode = 140;   % Beginning of a post-convergence call

                                                history.To_COL(nStep) = To_COL;
    history.T_DHW_BOT(nStep) = T_DHW_BOT;
    history.T_DHW_TOP(nStep) = T_DHW_TOP;
    history.HP_Source_Flowrate(nStep) = HP_Source_Flowrate;
    history.HP_Load_Flowrate(nStep) = HP_Load_Flowrate;
    history.HP_Q_Source(nStep) = HP_Q_Source;
    history.HP_Q_Load(nStep) = HP_Q_Load;
    history.HP_Power(nStep) = HP_Power;
    history.HP_COP(nStep) = HP_COP;
    history.HP_TSource_Out(nStep) = HP_TSource_Out;
    history.HP_TLoad_Out(nStep) = HP_TLoad_Out;
    history.HP_TSource_In(nStep) = HP_TSource_In;
    history.HP_TLoad_In(nStep) = HP_TLoad_In;
    history.HP_EBE(nStep) = HP_EBE;
    history.STC_Useful_EGain(nStep) = STC_Useful_EGain;
    history.DHW_EStore_Rate(nStep) = DHW_EStore_Rate;
    history.T_DHW_MID(nStep) = T_DHW_MID;
```

```
464     history.T_DHW_AVG(nStep) = T_DHW_AVG;
465                                         history.DrawTemp(nStep) = DrawTemp;
466     history.DrawFlow(nStep) = DrawFlow;
467     history.MainsTemp(nStep) = MainsTemp;
468     history.HX_TSourceIn(nStep) = HX_TSourceIn;
469     history.HX_TSourceOut(nStep) = HX_TSourceOut;
470     history.HX_SourceFlowRate(nStep) = HX_SourceFlowRate;
471     history.HX_T_LoadIn(nStep) = HX_T_LoadIn;
472     history.HX_T_LoadOut(nStep) = HX_T_LoadOut;
473     history.HX_LoadFlowRate(nStep) = HX_LoadFlowRate;
474     history.Controller_Mode(nStep) = Controller_Mode;
475     history.AuxHeat_Call(nStep)= AuxHeat_Call;
476     history.DHW_E_Dot_Draw(nStep)= DHW_E_Dot_Draw;
477     history.DHW_E_Dot_Charge(nStep) = DHW_E_Dot_Charge;
478     history.DHW_AUX_E_Dot(nStep) = DHW_AUX_E_Dot;
479     history. T_DHW_Thermostat(nStep) =  T_DHW_Thermostat;
480     history.trnTime(nStep)=trnTime;
481     history.L_S(nStep) = (DHW_E_Dot_Charge*trnTimeStep)-
(HP_Power*trnTimeStep);
482     history.L_A(nStep) =
(DHW_AUX_E_Dot*trnTimeStep)+(HP_Power*trnTimeStep);
483     history.CumulativeSolarFraction(nStep) =
sum(history.L_S)/(sum(history.L_A)+sum(history.L_S));
484
485
486     if (nStep==1)
487         history.DHW_E_Draw(nStep)= DHW_E_Dot_Draw*trnTimeStep;
488         history.DHW_E_Charge(nStep) = DHW_E_Dot_Charge*trnTimeStep;
489         history.DHW_AUX_E(nStep) = DHW_AUX_E_Dot*trnTimeStep;
490         history.HP_Energy(nStep) = HP_Power*trnTimeStep;
491     else
492         history.DHW_E_Draw(nStep)= history.DHW_E_Draw(nStep-1) +
DHW_E_Dot_Draw*trnTimeStep;
493         history.DHW_E_Charge(nStep) = history.DHW_E_Charge(nStep-1) +
DHW_E_Dot_Charge*trnTimeStep;
494         history.DHW_AUX_E(nStep) = history.DHW_AUX_E(nStep-1) +
DHW_AUX_E_Dot*trnTimeStep;
495         history.HP_Energy(nStep) = history.HP_Energy(nStep-1) +
HP_Power*trnTimeStep;
496     end
497     mFileErrorCode = 0; % Tell TRNSYS that we reached the end of the m-
file without errors
498     return  % Do not update outputs at this call
```

```
499
500 end
501
502
503 % --- All iterative calls ----------------------------------------------
    ------------------------------------------
504 % ------------------------------------------------------------------------
    ------------------------------------------
505
506 % --- If this is a first call in the time step, increment counter ---
507
508 if ( trnInfo(7) == 0 )
509     nStep = nStep+1;
510 end
511
512 % --- Get TRNSYS Inputs ---
513
514 nI = trnInfo(3);      % For bookkeeping
515 nO = trnInfo(6);    % For bookkeeping
516
517 To_COL = trnInputs(1);
518 T_DHW_BOT = trnInputs(2);
519 T_DHW_TOP    = trnInputs(3);
520 T_DHW_Thermostat = trnInputs(36);
521
522 mFileErrorCode = 150;    % After reading inputs
523
524 % --- Make Controls Decisions here ---
525
526 %Solar Charging Mode Decisions
527 if (max([T_DHW_TOP, T_DHW_MID, T_DHW_BOT])>=99) % CSA Safety requirement
    that all charging means are off if water temperature reaches 99C
528     Controller_Mode=0; % System mode=0 ==> HP and HX and Circulators Off.
529     Pump1_Signal =0;
530     Pump2_Signal =0;
531     Diverter1_Signal=0;
532     Diverter2_Signal=0;
533     HP_CALL =0;
534     N_Compressor=0;
535
```

```
536 elseif(T_DHW_Thermostat>(T_DHW_Solar_Set+T_Deadband)) %Tank is above  the solar setpoint, no heat required.
537     Controller_Mode=0; % System mode=0 ==> HP and HX and Circulators Off.
538     Pump1_Signal =0;
539     Pump2_Signal =0;
540     Diverter1_Signal=0;
541     Diverter2_Signal=0;
542     HP_CALL =0;
543     N_Compressor=0;
544
545 elseif(T_DHW_Thermostat>=T_DHW_Solar_Set) %Tank top is within deadband, no control change.
546     % Do Nothing
547 elseif(nnz(trnTime > SunriseTimes+SunriseSetDelay & trnTime < SunsetTimes-SunriseSetDelay)) % Tests if the current simulation time is between the hours of sunrise and sunset for any day of the year.
548     % Calling for heat and within daylight hours
549     % Decide between HX and HP HERE
550     if(trnTime-tLastControlChange>=t_timeout) %Check if enough time has elapsed to change Controller_Mode
551         if((To_COL-T_DHW_BOT>=T_HX_MinDiff) && (AvailableModes==0 ||AvailableModes==1))
552             % Activate HX Mode (Mode1) and set circulators and diverters.
553             if (Controller_Mode~=1)
554                 tLastControlChange = trnTime;
555             end
556             Controller_Mode=1;
557             Pump1_Signal =HX_Pump_Setting;
558             Pump2_Signal =HX_Pump_Setting;
559             Diverter1_Signal=HX_Diverter_Setting;
560             Diverter2_Signal=HX_Diverter_Setting;
561             HP_CALL =0;
562             N_Compressor=0;
563
564         elseif ((To_COL>T_HP_Cutin) && (AvailableModes==0 ||AvailableModes==2))
565             % Activate HP Mode (Mode2) and set circulators and diverters.
566             if (Controller_Mode~=2)
567                 tLastControlChange = trnTime;
568             end
569             Controller_Mode=2;
570             Pump1_Signal =HP_Pump_Setting;
```

```
571             Pump2_Signal =HP_Pump_Setting;
572             Diverter1_Signal=HP_Diverter_Setting;
573             Diverter2_Signal=HP_Diverter_Setting;
574             HP_CALL =1;
575             N_Compressor=1;
576
577         else
578             % Activate HP Warmup Mode (Mode3) and set circulators and
    diverters.
579             if ((Controller_Mode~=3)&&(AvailableModes~=3))
580                 tLastControlChange = trnTime;
581             end
582             Controller_Mode=3;
583             Pump1_Signal =HP_Pump_Setting;
584             Pump2_Signal =0;
585             Diverter1_Signal=HP_Diverter_Setting;
586             Diverter2_Signal=HP_Diverter_Setting;
587             HP_CALL =0;
588             N_Compressor=1;
589         end
590     end
591
592 else % Outside daylight hours
593     Controller_Mode=0;
594     Pump1_Signal =0;
595     Pump2_Signal =0;
596     Diverter1_Signal=0;
597     Diverter2_Signal=0;
598     HP_CALL =0;
599     N_Compressor=0;
600 end
601
602 % AUX Charging Decisions
    % 0=Heater locked out during daylight hours, 1=Heater locked out when HX
603 and HP are on, 2=Heater locked out only when HP is on, 3=Heater Never
    locked out.
604
605 if (max([T_DHW_TOP, T_DHW_MID, T_DHW_BOT])>=99) % CSA Safety requirement
    that all charging means are off if water temperature reaches 99C
606     AuxHeat_Call =0;
607 else
```

```
608     switch AUX_Mode
609         case 0
            if(nnz(trnTime > SunriseTimes+SunriseSetDelay & trnTime <
610 SunsetTimes-SunriseSetDelay)) % Tests if the current simulation time is
    between the hours of sunrise and sunset for any day of the year.
611                 AuxHeat_Call =0;
            elseif (T_DHW_Thermostat>(T_DHW_AUX_Set+T_Deadband)) %Tank is
612 above  the AUX setpoint, no heat required.
613                 AuxHeat_Call =0;
            elseif (T_DHW_Thermostat<T_DHW_AUX_Set) %Tank is below the
614 AUX setpoint, call for heat.
615                 AuxHeat_Call =1;
616             end
617         case 1
            if (Controller_Mode==1 || Controller_Mode==2) % Controller
618 mode is 1 or 2, therefore HX or HP is on.
619                 AuxHeat_Call =0;
            elseif (T_DHW_Thermostat>(T_DHW_AUX_Set+T_Deadband)) %Tank is
620 above  the AUX setpoint, no heat required.
621                 AuxHeat_Call =0;
            elseif (T_DHW_Thermostat<T_DHW_AUX_Set) %Tank is below the
622 AUX setpoint, call for heat.
623                 AuxHeat_Call =1;
624             end
625         case 2
            if (Controller_Mode==2) % Controller mode 2, therefore HP is
626 on.
627                 AuxHeat_Call =0;
            elseif (T_DHW_Thermostat>(T_DHW_AUX_Set+T_Deadband)) %Tank is
628 above  the AUX setpoint, no heat required.
629                 AuxHeat_Call =0;
            elseif (T_DHW_Thermostat<T_DHW_AUX_Set) %Tank is below the
630 AUX setpoint, call for heat.
631                 AuxHeat_Call =1;
632             end
633         case 3
            if (T_DHW_Thermostat>(T_DHW_AUX_Set+T_Deadband)) %Tank is
634 above  the AUX setpoint, no heat required.
635                 AuxHeat_Call =0;
            elseif (T_DHW_Thermostat<T_DHW_AUX_Set) %Tank is below the
636 AUX setpoint, call for heat.
637                 AuxHeat_Call =1;
638             end
639     end
640 end
```

```
641
642 % --- Set outputs ---
643 % trnOutputs(1) : Controller_Mode, Mode
644 % trnOutputs(2) : Pump1_Signal, Pump Signal (Source Side)
645 % trnOutputs(3) : Pump2_Signal, Pump Signal (Load Side)
646 % trnOutputs(4) : Diverter1_Signal,Diverter Signal (Source Side)
647 % trnOutputs(5) : Diverter3_Signal,Diverter Signal (Load Side)
648 % trnOutputs(6) : HP_CALL, HP Call signal, 1=on, 0=0ff;
649 % trnOutputs(7) : N_Compressor, HP Compressor Speed, 0=0% or 0 RPM,
    1=100%
650 % or 3600 RPM
651 % trnOutputs(8) : AuxHeat_Call, 1=Enable DHW Element, 0=Disable DHW
    Element;
652
653 trnOutputs(1) = Controller_Mode;
654 trnOutputs(2) = Pump1_Signal;
655 trnOutputs(3) = Pump2_Signal;
656 trnOutputs(4) = Diverter1_Signal;
657 trnOutputs(5) = Diverter2_Signal;
658 trnOutputs(6) = HP_CALL;
659 trnOutputs(7) = N_Compressor;
660 trnOutputs(8) = AuxHeat_Call;
661
662 mFileErrorCode = 0; % Tell TRNSYS that we reached the end of the m-file
    without errors
663 return
```

D.2 Predictive Control MATLAB Control Scripts

The following MATLAB script "MPC_Controller_IterativelyBuildControlSeries.m" is called by TRNSYS Type 155 once per time-step of the top-level simulation referred to in Chapter 6.

```
% MPC_Controller_IterativelyBuildControlSeries.m
% ------------------------------------------------------------------------------------------------
%
% A custom controller for Multi-Mode SAHP that works by iteratively sub-simulating and adjusting the control series to minimize or maximize a chosen variable, while minimizing fault conditions. (M-file called by TRNSYS type 155)
%
% Data passed from / to TRNSYS
% ----------------------------
%
% trnTime (1x1)        : simulation time
% trnInfo (15x1)       : TRNSYS info array
% trnInputs (nIx1)     : TRNSYS inputs
% trnStartTime (1x1)   : TRNSYS Simulation Start time
% trnStopTime (1x1)    : TRNSYS Simulation Stop time
% trnTimeStep (1x1)    : TRNSYS Simulation time step
% mFileErrorCode (1x1) : Error code for this m-file. It is set to 1 by TRNSYS and the m-file should set it to 0 at the
%                        end to indicate that the call was successful. Any non-zero value will stop the simulation
% trnOutputs (nOx1)    : TRNSYS outputs
%
%
% Notes:
% ------
%
% You can use the values of trnInfo(7), trnInfo(8) and trnInfo(13) to identify the call (e.g. first iteration, etc.)
% Real-time controllers (callingMode = 10) will only be called once per time step with trnInfo(13) = 1 (after convergence)
%
% The number of inputs is given by trnInfo(3)
% The number of expected outputs is given by trnInfo(6)
% WARNING: if multiple units of Type 155 are used, the variables passed from/to TRNSYS will be sized according to
%          the maximum required by all units. You should cope with that by only using the part of the arrays that is
```

```
30 %          really used by the current m-File. Example: use "nI =
   trnInfo(3); myInputs = trnInputs(1:nI);"
31 %                                                    rather than
   "MyInputs = trnInputs;"
32 %          Please also note that all m-files share the same workspace in
   Matlab (they are "scripts", not "functions") so
33 %          variables like trnInfo, trnTime, etc. will be overwritten at
   each call.
34 %
35 % ------------------------------------------------------------------------
   ---------------------------------------------
36 % The component is  iterative
37 %
38 % trnInputs
39 % ---------
40 %
41 % trnInputs(1) : To_COL, degree C, collector outlet temperature
42 % trnInputs(2) : T_DHW_BOT, degree C, DHW Tank Temperature at Bottom
43 % trnInputs(3) : T_DHW_TOP, degree C, DHW Tank Temperature at Top
44 % trnInputs(4) : HP_Source_Flowrate, kg/h, Flow rate on Source side of HP
45 % trnInputs(5) : HP_Load_Flowrate, kg/h, Flow rate on load side of HP
46 % trnInputs(6) : HP_Q_Source, W, Heat Transfer rate on source side of HP
47 % trnInputs(7) : HP_Q_Load, W, Heat Transfer rate on Load side of HP
48 % trnInputs(8) : HP_Power, W, HP Compressor Power Consumption
49 % trnInputs(9) : HP_COP, [dim], HP  Coefficient of Performance
50 % trnInputs(10) : HP_TSource_Out, degree C, HP Souce Side outlet
   temperature
51 % trnInputs(11) : HP_TLoad_Out, degree C, HP Load Side outlet temperature
52 % trnInputs(12) : HP_TSource_In, degree C, HP Souce Side inlet
   temperature
53 % trnInputs(13) : HP_TLoad_In, degree C, HP Load Side inlet temperature
54 % trnInputs(14) : HP_EBE, W, HP Energy balance Error
55 % trnInputs(15) : STC_Useful_EGain, kJ/h, Useful energy gain at STC
56 % trnInputs(16) : DHW_EStore_Rate, kJ/h, DHW Tank Energy Storage rate
57 % trnInputs(17) : T_DHW_MID, degree C, DHW Tank Temperature Middle
58 % trnInputs(18) : T_DHW_AVG, degree C, DHW Tank Temperature Average
59 % trnInputs(19) : DrawTemp, degree C, Temperature of DHW draw
60 % trnInputs(20) : DrawFlow, kg/h, Flow rate of DHW draw
61 % trnInputs(21) : MainsTemp, degree C, Temperature incoming Mains Water
62 % trnInputs(22) : HX_TSourceIn, degree C, HP Souce Side inlet
   tempertureHC
63 % trnInputs(23) : HX_TSourceOut, degree C, HP Source Side outlet
   temperature
64 % trnInputs(24) : HX_SourceFlowRate, kg/h, HP Souce Side Flow rate
```

```
% trnInputs(25) : HX_T_LoadIn, degree C, HP Load Side inlet temperature
% trnInputs(26) : HX_T_LoadOut, degree C, HP Load Side outlet temperature
% trnInputs(27) : HX_LoadFlowRate, kg/h, HP load Side Flow rate
% trnInputs(28) : DHW_E_Delivery_Draw, kJ/h, DHW Draw Side Energy
Delivery rate
% trnInputs(29) : DHW_E_Delivery_Charge, kJ/h, DHW Charge Side Energy
Delivery rate
% trnInputs(30) : DHW_AUX_Heating, kJ/h, DHW Auxiliary Heating rate
% trnInputs(31) : T_DHW_AUX_Set, C, Setpoint Temperature for the Aux
Charging the DHW Tank
% trnInputs(32) : T_Deadband = ; C, Deadband for DHW Setpoint
% trnInputs(33) : T_HX_MinDiff, C, Minimum Dt for HX operation
% trnInputs(34) : T_HP_Cutin, C, Cutin Source side temperature for the HP
% trnInputs(35) : SunriseSetDelay = h, the number of hours after/before
sunrise/set to activate the HX or HP controls
% trnInputs(36) : T_DHW_Thermostat, degree C, DHW  Thermostat Temperature
% trnInputs(37) : AUX_Mode, The mode of operation for the Auxiliary
heater.
% trnInputs(38) : T_DHW_Solar_Set, The temperature setpoint for solar
charging the DHW Tank.
% trnInputs(39) : AvailableModes, 0=No Restriction, 1= HX Only, 2=HP
Only, 3=No Solar Modes.
% trnInputs(40) : ModulateCompressor, 0=No, 1= 3 Stage based on T in.
% trnInputs(41) : T_DHW_Node_2
% trnInputs(42) : T_DHW_Node_3
% trnInputs(43) : T_DHW_Node_4
% trnInputs(44) : T_DHW_Node_5
% trnInputs(45) : T_DHW_Node_6
% trnInputs(46) : T_DHW_Node_8
% trnInputs(47) : T_DHW_Node_9
% trnInputs(48) : T_DHW_Node_11
% trnInputs(49) : T_DHW_Node_12
% trnInputs(50) : T_DHW_Node_13
% trnInputs(51) : T_DHW_Node_14
% trnInputs(52) : T_DHW_Node_15
% trnInputs(53) : T_DHW_Node_16
% trnInputs(54) : T_DHW_Node_17
% trnInputs(55) : T_DHW_Node_18
% trnInputs(56) : T_DHW_Node_19
% trnInputs(57) : TotalTiltedRadiation
%
% trnOutputs
% ----------
```

```
%
% trnOutputs(1) : Controller_Mode, Mode
% trnOutputs(2) : Pump1_Signal, Pump Signal (Source Side)
% trnOutputs(3) : Pump2_Signal, Pump Signal (Load Side)
% trnOutputs(4) : Diverter1_Signal,Diverter Signal (Source Side)
% trnOutputs(5) : Diverter3_Signal,Diverter Signal (Load Side)
% trnOutputs(6) : HP_CALL, HP Call signal, 1=on, 0=0ff;
% trnOutputs(7) : N_Compressor, HP Compressor Speed, 0=0% or 0 RPM, 1=100%
% or 3600 RPM
% trnOutputs(8) : AuxHeat_Call, 1=Enable DHW Element, 0=Disable DHW Element;
%
% JCH, July 13, 2022
% ------------------------------------------------------------------------------------------------
% TRNSYS sets mFileErrorCode = 1 at the beginning of the M-File for error detection
% This file increments mFileErrorCode at different places. If an error occurs in the m-file the last succesful step will
% be indicated by mFileErrorCode, which is displayed in the TRNSYS error message
% At the very end, the m-file sets mFileErrorCode to 0 to indicate that everything was OK
mFileErrorCode = 100;    % Beginning of the m-file
% --- Controller Parameters-----------------------------------------------------------------------
% ------------------------------------------------------------------------------------------------
T_DHW_AUX_Set =trnInputs(31); %50; % Degree C (DHW Tank Setpoint Temperature)
T_Deadband =trnInputs(32); %2; % Degree C (DHW Tank Deadband)
T_HX_MinDiff = trnInputs(33); %2; % Degree C (Minimum Dt for HX operation)
T_HP_Cutin = trnInputs(34); %10; % Degree C (Cutin Source side temperature for the HP)
SunriseSetDelay = trnInputs(35); %4;% h, the number of hours after/before sunrise/set to activate the HX or HP controls
AUX_Mode = trnInputs(37); % 0=Heater locked out during daylight hours, 1, Heater locked out when HX and HP are on, 2, Heater locked out when HP is on, 3, Heater Never locked out.
T_DHW_Solar_Set = trnInputs(38); % T_DHW_Solar_Set, The temperature setpoint for solar charging the DHW Tank.
AvailableModes = trnInputs(39);  %AvailableModes, 0=No Restriction, 1= HX Only, 2=HP Only.
t_timeout =  0.25; % hour (Timeout for next control decision)
SunriseTimes = [7.709980987
```

```
    ]';
SunsetTimes = [16.49908377
    ]';
Cp=4.19; % The specific heat of water in kJ/kgK
HX_Pump_Setting = 0.4651162; % Pump rated at kg/min, at 46.51% Speed = 5 kg/min
HP_Pump_Setting = 1; % Pump rated at kg/min, at 100% Speed
HX_Diverter_Setting = 0;
HP_Diverter_Setting = 1;
toPlot=true;
mFileErrorCode = 110;    % After setting parameters
% --- Process Inputs ------------------------------------------------------------------------------
% -------------------------------------------------------------------------------------------------
To_COL = trnInputs(1);
T_DHW_BOT = trnInputs(2);
T_DHW_TOP = trnInputs(3);
HP_Source_Flowrate = trnInputs(4);
HP_Load_Flowrate = trnInputs(5);
% HP_Q_Source = 3.6*trnInputs(6);% Convert W to kJ/h
% HP_Q_Load = 3.6*trnInputs(7); % Convert W to kJ/h
% HP_Power = 3.6*trnInputs(8);% Convert W to kJ/h
HP_Q_Source = trnInputs(6);% Convert W to kJ/h
HP_Q_Load = trnInputs(7); % Convert W to kJ/h
HP_Power = trnInputs(8);% Convert W to kJ/h
HP_COP = trnInputs(9);
HP_TSource_Out = trnInputs(10);
HP_TLoad_Out = trnInputs(11);
HP_TSource_In = trnInputs(12);
HP_TLoad_In = trnInputs(13);
HP_EBE = 3.6*trnInputs(14); % Convert W to kJ/h
STC_Useful_EGain = trnInputs(15);
DHW_EStore_Rate = trnInputs(16);
T_DHW_MID = trnInputs(17);
T_DHW_AVG = trnInputs(18);
DrawTemp = trnInputs(19);
DrawFlow = trnInputs(20);
MainsTemp = trnInputs(21);
HX_TSourceIn = trnInputs(22);
HX_TSourceOut = trnInputs(23);
```

```
HX_SourceFlowRate = trnInputs(24);
HX_T_LoadIn = trnInputs(25);
HX_T_LoadOut = trnInputs(26);
HX_LoadFlowRate = trnInputs(27);
DHW_E_Dot_Draw = trnInputs(28);
DHW_E_Dot_Charge= -trnInputs(29); % Sign reconciled (Delivery vs. Supply)
DHW_AUX_E_Dot = trnInputs(30);
T_DHW_Thermostat = trnInputs(36);
ModulateCompressor = trnInputs(40);
T_SolarSource = HX_TSourceIn;
%T_SolarSource = To_COL;
T_DHW_Profile =[trnInputs(3)
    trnInputs(43)
    trnInputs(46)
    trnInputs(49)
    trnInputs(53)
    trnInputs(2) ];
% T_DHW_Node_1=trnInputs(3);
% T_DHW_Node_2=trnInputs(41);
% T_DHW_Node_3=trnInputs(42);
% T_DHW_Node_4=trnInputs(43);
% T_DHW_Node_5=trnInputs(44);
% T_DHW_Node_6=trnInputs(45);
% T_DHW_Node_7=trnInputs(36);
% T_DHW_Node_8=trnInputs(46);
% T_DHW_Node_9=trnInputs(47);
% T_DHW_Node_10=trnInputs(17);
% T_DHW_Node_11=trnInputs(48);
% T_DHW_Node_12=trnInputs(49);
% T_DHW_Node_13=trnInputs(50);
% T_DHW_Node_14=trnInputs(51);
% T_DHW_Node_15=trnInputs(52);
% T_DHW_Node_16=trnInputs(53);
% T_DHW_Node_17=trnInputs(54);
% T_DHW_Node_18=trnInputs(55);
% T_DHW_Node_19=trnInputs(56);
% T_DHW_Node_20=trnInputs(2);
[DHWTankEnergy, DHWAvailableEnergy, DHWUnavailableEnergy] =
TankState80gal(T_DHW_Profile);
TotalTiltedRadiation = trnInputs(57);
```

```
mFileErrorCode = 120;    % After processing inputs
% --- First call of the simulation: initial time step (no iterations) ---
% ------------------------------------------------------------------------
% (note that Matlab is initialized before this at the info(7) = -1 call,
but the m-file is not called)

if ( (trnInfo(7) == 0) & (trnTime-trnStartTime < 1e-6) )

    % This is the first call (Counter will be incremented later for this
very first call)
    nCall = 0;
    % This is the first time step
    nStep = 1;
    nStep_LastPlot = 3;
    % Initialize history of the variables for plotting at the end of the
simulation
    nTimeSteps = (trnStopTime-trnStartTime)/trnTimeStep + 1;
    history.To_COL= zeros(nTimeSteps,1);
    history.T_DHW_BOT= zeros(nTimeSteps,1);
    history.T_DHW_TOP= zeros(nTimeSteps,1);
    history.HP_Source_Flowrate= zeros(nTimeSteps,1);
    history.HP_Load_Flowrate= zeros(nTimeSteps,1);
    history.HP_Q_Source= zeros(nTimeSteps,1);
    history.HP_Q_Load= zeros(nTimeSteps,1);
    history.HP_Power= zeros(nTimeSteps,1);
    history.HP_COP= zeros(nTimeSteps,1);
    history.HP_TSource_Out= zeros(nTimeSteps,1);
    history.HP_TLoad_Out= zeros(nTimeSteps,1);
    history.HP_TSource_In= zeros(nTimeSteps,1);
    history.HP_TLoad_In= zeros(nTimeSteps,1);
    history.HP_EBE= zeros(nTimeSteps,1);
    history.STC_Useful_EGain= zeros(nTimeSteps,1);
    history.DHW_EStore_Rate= zeros(nTimeSteps,1);
    history.T_DHW_MID= zeros(nTimeSteps,1);
    history.T_DHW_AVG= zeros(nTimeSteps,1);
    history.DrawTemp= zeros(nTimeSteps,1);
    history.DrawFlow= zeros(nTimeSteps,1);
    history.MainsTemp= zeros(nTimeSteps,1);
    history.HX_TSourceIn= zeros(nTimeSteps,1);
    history.HX_TSourceOut= zeros(nTimeSteps,1);
```

```
245     history.HX_SourceFlowRate= zeros(nTimeSteps,1);
246     history.HX_T_LoadIn= zeros(nTimeSteps,1);
247     history.HX_T_LoadOut= zeros(nTimeSteps,1);
248     history.HX_LoadFlowRate= zeros(nTimeSteps,1);
249     history.Controller_Mode = zeros(nTimeSteps,1);
250     history.AuxHeat_Call = zeros(nTimeSteps,1);
251     history.DHW_E_Dot_Draw= zeros(nTimeSteps,1);
252     history.DHW_E_Draw = zeros(nTimeSteps,1);
253     history.DHW_E_Dot_Charge = zeros(nTimeSteps,1);
254     history.DHW_E_Charge = zeros(nTimeSteps,1);
255     history.DHW_AUX_E_Dot = zeros(nTimeSteps,1);
256     history.DHW_AUX_E = zeros(nTimeSteps,1);
257     history.L_S = zeros(nTimeSteps,1);
258     history.L_A = zeros(nTimeSteps,1);
259     history.HP_Energy = zeros(nTimeSteps,1);
260     history.SolarFraction= zeros(nTimeSteps,1);
261     history.CumulativeSolarFraction = zeros(nTimeSteps,1);
262     history.trnTime = zeros(nTimeSteps,1);
263     history.T_DHW_Thermostat = zeros(nTimeSteps,1);
264     tLastControlChange = 0;
265     history.DHWTankEnergy = zeros(nTimeSteps,1);
266     history.DHWAvailableEnergy = zeros(nTimeSteps,1);
267     history.DHWUnavailableEnergy = zeros(nTimeSteps,1);
268     history.TotalTiltedRadiation = zeros(nTimeSteps,1);
269     Controller_Mode = 0;
270     Pump1_Signal =0;
271     Pump2_Signal =0;
272     Diverter1_Signal=0;
273     Diverter2_Signal=0;
274     HP_CALL =0;
275     N_Compressor=0;
276     AuxHeat_Call =0;
277     simHourLastSubSim = -1;
278     afterFirstSubSim = false;
279     rng('shuffle');
280     % No return, we will calculate the solar collector performance during
    this call
281     mFileErrorCode = 130;    % After initialization
282
283 end
```

```
% --- Very last call of the simulation (after the user clicks "OK"):
if ( trnInfo(8) == -1 )
    mFileErrorCode = 1000;
    % Generate an output file to log simulation data.

T=table(history.trnTime,mod(history.trnTime,24),history.HP_Source_Flowrate,...
        history.HP_Load_Flowrate,
        history.HP_Power,
        history.HP_TLoad_Out,
        history.HP_EBE,
        history.T_DHW_TOP,
        history.DrawTemp,
        history.HX_TSourceOut,
        history.HX_T_LoadOut,
        history.AuxHeat_Call, history.DHW_E_Dot_Draw, history.DHW_E_Draw,...
        history.DHW_E_Dot_Charge, history.DHW_E_Charge, history.DHW_AUX_E_Dot, history.DHW_AUX_E, ...
        history.CumulativeSolarFraction, history.T_DHW_Thermostat,...
        history.DHWTankEnergy, history.DHWAvailableEnergy, history.DHWUnavailableEnergy, history.TotalTiltedRadiation);
    T.Properties.VariableNames = {'SimTime','TOD','HP_Source_Flowrate',
        'HP_Load_Flowrate',
        'HP_COP',
        'HP_TLoad_In',
        'T_DHW_TOP',
        'DrawTemp',
        'HX_SourceFlowRate',
        'Controller_Mode', 'AuxHeat_Call', 'DHW Draw Energy Delivery Rate [kJ/h]',...
        'DHW Draw Energy Delivered [kJ]', 'DHW Charge Energy Delivery Rate [kJ/h]',...
        'DHW Charge Energy Delivered [kJ]', 'DHW AUX Heating Rate [kJ/h]',...
        'DHW AUX Heat Delivered[kJ]', 'Cumulative Solar Fraction', 'T_DHW_Thermostat',...
        'DHW Tank Energy [kJ]','DHW Available Energy [kJ]', 'DHW Unavailable Energy', 'TotalTiltedRadiation'};
    % Calculate Daily Solar Fraction(s) and Daily Totals of Simulation Variables
    numdays=ceil((history.trnTime(end)-history.trnTime(3))/24); % Count the days in the simulation
    if (history.trnTime(end) == history.trnTime(3) && length(history.trnTime)>= (8760/trnTimeStep))
```

```
316         numdays=365;
317     end
318     days = [1:1:numdays]'; % An array with one element per day in the
simulation
319     days=days+floor(history.trnTime(3)/24); % Adjust index to be the
Julian day of each day in the simulation
320     DailySolarFraction=zeros(length(days),1);
321     DailyDHWDrawEnergy=zeros(length(days),1);
322     DailyDHWChargeEnergy= zeros(length(days),1);
323     DailyDHWAUXEnergy=zeros(length(days),1);
324     DailyHPEnergy=zeros(length(days),1);
325     DailyDrawMass=zeros(length(days),1);
326     DailyInadequateDrawMass=zeros(length(days),1);
327     DailyNumDraws=zeros(length(days),1);
328     DailyNumInadequateDraws=zeros(length(days),1);
329     for i=1:1:length(days)
330         dailyBounds=[((i-1)*24/trnTimeStep)+3 ((i)*24/trnTimeStep)+3];
331         if i==length(days)
332             dailyBounds=[((i-1)*24/trnTimeStep)+3
length(history.trnTime)];
333         end

334 DailySolarFraction(i)=sum(history.L_S(dailyBounds(1):dailyBounds(2)))/(su
m(history.L_A(dailyBounds(1):dailyBounds(2)))+sum(history.L_S(dailyBounds
(1):dailyBounds(2))));

335 DailyDHWDrawEnergy(i)=trnTimeStep*sum(history.DHW_E_Dot_Draw(dailyBounds(
1):dailyBounds(2)));

336 DailyDHWChargeEnergy(i)=trnTimeStep*sum(history.DHW_E_Dot_Charge(dailyBou
nds(1):dailyBounds(2)));

337 DailyDHWAUXEnergy(i)=trnTimeStep*sum(history.DHW_AUX_E_Dot(dailyBounds(1)
:dailyBounds(2)));

338 DailyHPEnergy(i)=trnTimeStep*sum(history.HP_Power(dailyBounds(1):dailyBou
nds(2))); % HP Energy Consumption in kJ

339 DailyDrawMass(i)=trnTimeStep*sum(history.DrawFlow(dailyBounds(1):dailyBou
nds(2)));
340         %Count the number of draw events
341         TempVector1=history.DrawFlow(dailyBounds(1):dailyBounds(2));
342         TempVector2=zeros(length(TempVector1),1);
343         for j=2:1:length(TempVector1)
344             if (TempVector1(j-1)==0 && TempVector1(j)>0)
```

```
345                 TempVector2(j)=1;
346             end
347         end
348         DailyNumDraws(i)=sum(TempVector2);
349         % Divide out inadequate draw flow rates
350         TempVector1=history.DrawFlow(dailyBounds(1):dailyBounds(2));
351         TempVector2=history.DrawTemp(dailyBounds(1):dailyBounds(2));
352         TempVector3=zeros(length(TempVector1),1);
353         for j=2:1:length(TempVector1)
354             if (TempVector1(j)>0 && TempVector2(j)<=35)
355                 TempVector3(j)=TempVector1(j);
356             end
357         end
358         DailyInadequateDrawMass(i)=trnTimeStep*sum(TempVector3);
359         TempVector2=zeros(length(TempVector1),1);
360         %Count the number of draw events
361         for j=2:1:length(TempVector1)
362             if (TempVector3(j-1)==0 && TempVector3(j)>0)
363                 TempVector2(j)=1;
364             end
365         end
366         DailyNumInadequateDraws(i)=sum(TempVector2);
367     end
    T2=table(days,DailySolarFraction,DailyDHWDrawEnergy,
368 DailyDHWChargeEnergy, DailyDHWAUXEnergy,
DailyHPEnergy,DailyDrawMass,DailyNumDraws,DailyInadequateDrawMass,DailyNu
mInadequateDraws);
    T2.Properties.VariableNames = {'Day','Solar Fraction','DHW Draw
Energy Delivered [kJ]','DHW Charge Energy Delivered [kJ]','DHW AUX Heat
369 Delivered[kJ]', 'HP Energy Consumed[kJ]', 'Daily DHW Draw Mass [kg]',
'Daily Number of Draws','Daily inadequate DHW Draw Mass [kg]', 'Daily
Number of Inadequate Draws'};
370     % Cauculate Monthly Solar Fraction And other totals
371     monthHourDef=[  1 0
372         2 744
373         3 1416
374         4 2160
375         5 2880
376         6 3624
377         7 4344
378         8 5088
379         9 5832
```

```
380         10 6552
381         11 7296
382         12 8016
383         13 8760];
384     if exist('monthsSimulated','var')
385         clear monthsSimulated;
386     end
387     j=0;
388     for i = 1:1:12
389         if any(history.trnTime(3:end) >=monthHourDef(i,2) &
history.trnTime(3:end) <=monthHourDef(i+1,2))
390             j=j+1;
391             monthsSimulated(j)=i;
392         end
393     end
394     monthlySolarFraction=zeros(length(monthsSimulated),1);
395     monthlyDHWDrawEnergy=zeros(length(monthsSimulated),1);
396     monthlyDHWChargeEnergy=zeros(length(monthsSimulated),1);
397     monthlyDHWAUXEnergy=zeros(length(monthsSimulated),1);
398     monthlyHPEnergy=zeros(length(monthsSimulated),1);
399     monthlyDrawMass=zeros(length(monthsSimulated),1);
400     monthlyInadequateDrawMass=zeros(length(monthsSimulated),1);
401     monthlyNumDraws=zeros(length(monthsSimulated),1);
402     monthlyNumInadequateDraws=zeros(length(monthsSimulated),1);
403     j=0;
404     for i=monthsSimulated
405         j=j+1;
406         loweridx=find(history.trnTime>=monthHourDef(i,2),1);
407         upperidx=find(history.trnTime<=monthHourDef(i+1,2),1,"last");

408 monthlySolarFraction(j)=sum(history.L_S(loweridx:upperidx))/(sum(history.
    L_A(loweridx:upperidx))+sum(history.L_S(loweridx:upperidx)));

409 monthlyDHWDrawEnergy(j)=trnTimeStep*sum(history.DHW_E_Dot_Draw(loweridx:u
    pperidx));

410 monthlyDHWChargeEnergy(j)=trnTimeStep*sum(history.DHW_E_Dot_Charge(loweri
    dx:upperidx));

411 monthlyDHWAUXEnergy(j)=trnTimeStep*sum(history.DHW_AUX_E_Dot(loweridx:upp
    eridx));

412 monthlyHPEnergy(j)=trnTimeStep*sum(history.HP_Power(loweridx:upperidx));
```

```
413
monthlyDrawMass(j)=trnTimeStep*sum(history.DrawFlow(loweridx:upperidx));
414         %Count the number of draw events
415         TempVector1=history.DrawFlow(loweridx:upperidx);
416         TempVector2=zeros(length(TempVector1),1);
417         for k=2:1:length(TempVector1)
418             if (TempVector1(k-1)==0 && TempVector1(k)>0)
419                 TempVector2(k)=1;
420             end
421         end
422         monthlyNumDraws(j)=sum(TempVector2);
423         % Divide out inadequate draw flow rates
424         TempVector1=history.DrawFlow(loweridx:upperidx);
425         TempVector2=history.DrawTemp(loweridx:upperidx);
426         TempVector3=zeros(length(TempVector1),1);
427         for k=2:1:length(TempVector1)
428             if (TempVector1(k)>0 && TempVector2(k)<=35)
429                 TempVector3(k)=TempVector1(k);
430             end
431         end
432         monthlyInadequateDrawMass(j)=trnTimeStep*sum(TempVector3);
433         TempVector2=zeros(length(TempVector1),1);
434         %Count the number of draw events
435         for k=2:1:length(TempVector1)
436             if (TempVector3(k-1)==0 && TempVector3(k)>0)
437                 TempVector2(k)=1;
438             end
439         end
440         monthlyNumInadequateDraws(j)=sum(TempVector2);
441     end
442     monthsSimulated=monthsSimulated';
    T3=table(monthsSimulated,monthlySolarFraction, monthlyDHWDrawEnergy,
443 monthlyDHWChargeEnergy, monthlyDHWAUXEnergy,
monthlyHPEnergy,monthlyDrawMass,monthlyNumDraws,monthlyInadequateDrawMass
,monthlyNumInadequateDraws);
    T3.Properties.VariableNames = {'Month','Solar Fraction','DHW Draw
Energy Delivered [kJ]','DHW Charge Energy Delivered [kJ]','DHW AUX Heat
444 Delivered[kJ]', 'HP Energy Consumed[kJ]', 'Monthly DHW Draw Mass [kg]',
'Monthly Number of Draws','Monthly inadequate DHW Draw Mass [kg]',
'Monthly Number of Inadequate Draws'};
445
446     % Calculate an Annual Solar Fraction
```

```
447
448     % Output datafiles
449     if (~exist('RunNo'))
450         RunNo = 1;
451     end
452 writetable(T,sprintf('SAHP_Simulation_StepByStep_Output_Run_%d.csv',RunNo
    ));
453 writetable(T2,sprintf('SAHP_Simulation_Daily_Totalized_Output_Run_%d.csv'
    ,RunNo));
454 writetable(T3,sprintf('SAHP_Simulation_Monthly_Totalized_Output_Run_%d.cs
    v',RunNo));
455     system('move R:\MPC_Output\*.* "C:\Users\Julian\Documents\Waterloo
    School Work\SAHP\MATLAB_TRNSYS\MPC_Controller\Archived MPC Outputs"');
456 save(append('IterativeSimulations\',string(datetime("today")),'_',string(
    trnStartTime),'_',string(trnStopTime)))
457     RunNo = RunNo+1;
458     mFileErrorCode = 0; % Tell TRNSYS that we reached the end of the m-
    file without errors
459     return
460 end
461 % --- Post convergence calls: store values ----------------------------
    ---------------------------------------------
462 % ----------------------------------------------------------------------
    ---------------------------------------------
463 if (trnInfo(13) == 1)
464     mFileErrorCode = 140;   % Beginning of a post-convergence call
465
466     history.To_COL(nStep) = To_COL;
467     history.T_DHW_BOT(nStep) = T_DHW_BOT;
468     history.T_DHW_TOP(nStep) = T_DHW_TOP;
469     history.HP_Source_Flowrate(nStep) = HP_Source_Flowrate;
470     history.HP_Load_Flowrate(nStep) = HP_Load_Flowrate;
471     history.HP_Q_Source(nStep) = HP_Q_Source;
472     history.HP_Q_Load(nStep) = HP_Q_Load;
473     history.HP_Power(nStep) = HP_Power;
474     history.HP_COP(nStep) = HP_COP;
475     history.HP_TSource_Out(nStep) = HP_TSource_Out;
476     history.HP_TLoad_Out(nStep) = HP_TLoad_Out;
477     history.HP_TSource_In(nStep) = HP_TSource_In;
478     history.HP_TLoad_In(nStep) = HP_TLoad_In;
```

```
    history.HP_EBE(nStep) = HP_EBE;
    history.STC_Useful_EGain(nStep) = STC_Useful_EGain;
    history.DHW_EStore_Rate(nStep) = DHW_EStore_Rate;
    history.T_DHW_MID(nStep) = T_DHW_MID;
    history.T_DHW_AVG(nStep) = T_DHW_AVG;
    history.DrawTemp(nStep) = DrawTemp;
    history.DrawFlow(nStep) = DrawFlow;
    history.MainsTemp(nStep) = MainsTemp;
    history.HX_TSourceIn(nStep) = HX_TSourceIn;
    history.HX_TSourceOut(nStep) = HX_TSourceOut;
    history.HX_SourceFlowRate(nStep) = HX_SourceFlowRate;
    history.HX_T_LoadIn(nStep) = HX_T_LoadIn;
    history.HX_T_LoadOut(nStep) = HX_T_LoadOut;
    history.HX_LoadFlowRate(nStep) = HX_LoadFlowRate;
    history.Controller_Mode(nStep) = Controller_Mode;
    history.AuxHeat_Call(nStep)= AuxHeat_Call;
    history.DHW_E_Dot_Draw(nStep)= DHW_E_Dot_Draw;
    history.DHW_E_Dot_Charge(nStep) = DHW_E_Dot_Charge;
    history.DHW_AUX_E_Dot(nStep) = DHW_AUX_E_Dot;
    history.T_DHW_Thermostat(nStep) =  T_DHW_Thermostat;
    history.trnTime(nStep)=trnTime;
    history.L_S(nStep) = (DHW_E_Dot_Charge*trnTimeStep)-
(HP_Power*trnTimeStep);
    history.L_A(nStep) =
(DHW_AUX_E_Dot*trnTimeStep)+(HP_Power*trnTimeStep);
    history.CumulativeSolarFraction(nStep) =
sum(history.L_S)/(sum(history.L_A)+sum(history.L_S));
    history.DHWTankEnergy(nStep) = DHWTankEnergy;
    history.DHWAvailableEnergy(nStep) = DHWAvailableEnergy;
    history.DHWUnavailableEnergy(nStep) = DHWUnavailableEnergy;
    history.TotalTiltedRadiation(nStep)=TotalTiltedRadiation;
    if (nStep==1)
        history.DHW_E_Draw(nStep)= DHW_E_Dot_Draw*trnTimeStep;
        history.DHW_E_Charge(nStep) = DHW_E_Dot_Charge*trnTimeStep;
        history.DHW_AUX_E(nStep) = DHW_AUX_E_Dot*trnTimeStep;
        history.HP_Energy(nStep) = HP_Power*trnTimeStep;
    else
        history.DHW_E_Draw(nStep)= history.DHW_E_Draw(nStep-1) +
DHW_E_Dot_Draw*trnTimeStep;
        history.DHW_E_Charge(nStep) = history.DHW_E_Charge(nStep-1) +
DHW_E_Dot_Charge*trnTimeStep;
```

```
515         history.DHW_AUX_E(nStep) = history.DHW_AUX_E(nStep-1) +
DHW_AUX_E_Dot*trnTimeStep;
516         history.HP_Energy(nStep) = history.HP_Energy(nStep-1) +
HP_Power*trnTimeStep;
517     end
518     mFileErrorCode = 0; % Tell TRNSYS that we reached the end of the m-
file without errors
519     return  % Do not update outputs at this call
520 end
521 % --- All iterative calls ----------------------------------------------
-----------------------------------------------
522 % ------------------------------------------------------------------------
-----------------------------------------------
523 % --- If this is a first call in the time step, increment counter ---
524 if ( trnInfo(7) == 0 )
525     nStep = nStep+1;
526     % --- Get TRNSYS Inputs ---
527     nI = trnInfo(3);     % For bookkeeping
528     nO = trnInfo(6);   % For bookkeeping
529     To_COL = trnInputs(1);
530     T_DHW_BOT = trnInputs(2);
531     T_DHW_TOP   = trnInputs(3);
532     T_DHW_Thermostat = trnInputs(36);
533     mFileErrorCode = 150;   % After reading inputs
534     simHour=floor(trnTime);
        if(simHour~=0 && simHour~=floor(trnStopTime) && mod(simHour,5)==0
535 && simHour~=simHourLastSubSim && nStep>2) %Runs every 5 hours on the
hour.
536         LengthOfSubsim = 10;
537         simHourLastSubSim = simHour;
538         TimeBounds = [trnTime, trnTime+LengthOfSubsim];
539         if TimeBounds(2)>=trnStopTime
540             TimeBounds(2)=trnStopTime;
541         end
542         % Scheme for SubSimControlSeries:
543         % SubSimControlSeries(:,1) = Time of Year of each timestep.
544         % SubSimControlSeries(:,2) = Are Solar Modes Permissible
545         % (i.e. ==1 during daylight and ==0 outside of daylight hours.
546         % SubSimControlSeries(:,3) = The mode to demand of the controller
547         % for the timestep with 0 == System Off, 1 == HX Charging,
548         % 2 == HP Charging, 3 == HP Warmup
549         % SubSimControlSeries(:,4) = Whether the AUX heater is called for
```

```
550         % the timestep.
551         % SubSimControlSeries(:,5) = Whether the HP is locked out due to
552         % undertemp (1 = HP not locked out, 0 = locked out)
553         SubSimControlSeries = zeros(((TimeBounds(2)-
TimeBounds(1))/trnTimeStep)+1,5);
554         SubSimControlSeries(:,1) =
(TimeBounds(1):trnTimeStep:TimeBounds(2))';
555         SubSimControlSeries(:,5) = 1; % Initially, HP is unlocked.
556         %Test if the entire subsimulation is outside daylight hours
557         julianDay=floor(trnTime/24)+1;
            if ((TimeBounds(1)<SunriseTimes(julianDay)+SunriseSetDelay &&
TimeBounds(2)<SunriseTimes(julianDay)+SunriseSetDelay) ||
558 (TimeBounds(1)>SunsetTimes(julianDay)-SunriseSetDelay &&
TimeBounds(2)>SunsetTimes(julianDay)-SunriseSetDelay &&
TimeBounds(2)<SunriseTimes(julianDay+1)+SunriseSetDelay))
            %~(nnz(TimeBounds(1) > SunriseTimes+SunriseSetDelay &
559 TimeBounds(1) < SunsetTimes-SunriseSetDelay)) && (nnz(TimeBounds(2) >
SunriseTimes+SunriseSetDelay & TimeBounds(2) < SunsetTimes-
SunriseSetDelay))
560             SubSimControlSeries(:,2) = 0; % Solar Modes not permissible
561             %Test if the beginning of the subsimulation is outside
daylight hours
562         elseif ~(nnz(TimeBounds(1) > SunriseTimes+SunriseSetDelay &
TimeBounds(1) < SunsetTimes-SunriseSetDelay))
563             ind=find(SunriseTimes>TimeBounds(1),1); % Find the sunrize
immediately after the time in TimeBounds(1)

564 ind=find(SubSimControlSeries(:,1)>=SunriseTimes(ind)+SunriseSetDelay,1);%
Find the subsimulation time step corresponding to sunrize
565             SubSimControlSeries(ind:end,2) = 1; % Solar Modes Permissible
after Sunrise.
566             %Test if the end of the subsimulation is outside daylight
hours
567         elseif ~(nnz(TimeBounds(2) > SunriseTimes+SunriseSetDelay &
TimeBounds(2) < SunsetTimes-SunriseSetDelay))
568             ind=find(SunsetTimes>TimeBounds(2),1)-1; % Find the sunset
immediately before the time in TimeBounds(2)
569             ind=find(SubSimControlSeries(:,1)>=SunsetTimes(ind)-
SunriseSetDelay,1);
570             SubSimControlSeries(1:ind,2) = 1; % Solar Modes Permissible
Before Sunset.
571             %Otherwise, the entire subsimulation will be inside of
daylight hours.
572         else
573             SubSimControlSeries(:,2) = 1; % Solar Modes Permissible
574         end
575         %Run a subsimulation with the HX to use as a benchmark
576         SubSimControlSeries(:,3)=SubSimControlSeries(:,2)*1;% If Solar
Modes are possible, run the HX
```

```
        SubSimDeck1(TimeBounds,T_DHW_Profile, T_SolarSource,
577 'R:\SAHP_Matlab_MPC_Type1163_Sub-Sim_WithPlaceholders.dck'); % Make
    required Substitutions in the placeholder deck.
        ForecastInputFiles1(TimeBounds); %Pull the relevant weather and
578 draw profile information.

579 writematrix(SubSimControlSeries(:,3:4),'R:\InputFiles\ExtractedControls.c
    sv'); % Put SubSimControlSeries into a temporary TRNSYS input file.
        status=system('"C:\Trnsys18\Exe\TrnEXE64.exe"
580 "R:\SAHP_Matlab_MPC_Type1163_Sub-Sim_WithPlaceholders_Replaced.dck" /n');
    % Call a new instance of TRNSYS to run the sub-simulation deck.
581     mFileErrorCode = 155.1;

    [CurrentHXSubSimResults,~,~,~,~]=Post_Process1('R:\MPC_Output\MPC_Output.
582 csv','R:\MPC_Output\DHW_Temps.csv', trnTimeStep, history.L_S,
    history.L_A, SubSimControlSeries,false); % PostProcess the sub-simulation
    output calculating integrals, detecting faults, etc.

583 destination=sprintf('R:\\MPC_Output\\MPC_Output_%d_HX.csv',floor(trnTime)
    ); % Set a path to save the output file from the sub-simulation.
        movefile('R:\MPC_Output\MPC_Output.csv', destination); % Save the
584 sub-simulation output file to a unique path.
        movefile('R:\MPC_Output\MPC_Output.mat',
585 strrep(destination,'.csv','.mat')); % Save the sub-simulation output file
    to a unique path.
586     if(status~=0)
587         error("error in most recent TRNSYS CALL");
588     end
589     %Run a subsimulation with the HP to use as a benchmark
        SubSimControlSeries(:,3)=SubSimControlSeries(:,2)*2;% If Solar
590 Modes are possible, run the HP
        SubSimDeck1(TimeBounds,T_DHW_Profile, T_SolarSource,
591 'R:\SAHP_Matlab_MPC_Type1163_Sub-Sim_WithPlaceholders.dck'); % Make
    required Substitutions in the placeholder deck.
        ForecastInputFiles1(TimeBounds); %Pull the relevant weather and
592 draw profile information.

593 writematrix(SubSimControlSeries(:,3:4),'R:\InputFiles\ExtractedControls.c
    sv'); % Put SubSimControlSeries into a temporary TRNSYS input file.
        status=system('"C:\Trnsys18\Exe\TrnEXE64.exe"
594 "R:\SAHP_Matlab_MPC_Type1163_Sub-Sim_WithPlaceholders_Replaced.dck" /n');
    % Call a new instance of TRNSYS to run the sub-simulation deck.
595     mFileErrorCode = 155.2;

    [CurrentHPSubSimResults,~,~,~,~]=Post_Process1('R:\MPC_Output\MPC_Output.
596 csv','R:\MPC_Output\DHW_Temps.csv', trnTimeStep, history.L_S,
    history.L_A, SubSimControlSeries,false); % PostProcess the sub-simulation
    output calculating integrals, detecting faults, etc.
```

```
597 destination=sprintf('R:\\MPC_Output\\MPC_Output_%d_HP.csv',floor(trnTime)
    ); % Set a path to save the output file from the sub-simulation.
598         movefile('R:\MPC_Output\MPC_Output.csv', destination); % Save the
    sub-simulation output file to a unique path.
599         movefile('R:\MPC_Output\MPC_Output.mat',
    strrep(destination,'.csv','.mat')); % Save the sub-simulation output file
    to a unique path.
600         if(status~=0)
601             error("error in most recent TRNSYS CALL");
602         end
603         % First run of this loop will be with an empty control series if
    it
604         % is the very first subsimulation, if not, then the next
605         % subsimulation will import the part of
606         % OldSubSimControlSeries that overlaps.
607         if (afterFirstSubSim == true)
608             OverlapHours =
    min([SubSimControlSeries(end,1),OldSubSimControlSeries(end,1)])-
    max([SubSimControlSeries(1,1),OldSubSimControlSeries(1,1)]);
609             OverlapTimesteps = floor(OverlapHours/trnTimeStep);
610             SubSimControlSeries(:,[3,4])=0;
611             SubSimControlSeries(:,5)=1;

612 SubSimControlSeries(1:OverlapTimesteps+1,[3,4,5])=OldSubSimControlSeries(
    end-OverlapTimesteps:end,[3,4,5]);
613             %         figure(2)
614             %
    plot(SubSimControlSeries(:,1),SubSimControlSeries(:,3))
615             %         hold all;
616             %
    plot(SubSimControlSeries(:,1),SubSimControlSeries(:,4))
617             %
    plot(OldSubSimControlSeries(:,1),OldSubSimControlSeries(:,3))
618             %
    plot(OldSubSimControlSeries(:,1),OldSubSimControlSeries(:,4))
619             %         figure(1)
620         else
621             OldSubSimControlSeries= zeros(((TimeBounds(2)-
    TimeBounds(1))/trnTimeStep)+1,5);
622             SubSimControlSeries(:,3)=0;
623         end
624         % Each Iteration of the loop will add blocks to the Control
    Series
625         % to address the "Fault" Conditions that are detected in "Post
626         % Process"
```

```
627         i=0;
628         converged = false;
629         while converged == false
                SubSimDeck1(TimeBounds,T_DHW_Profile, T_SolarSource,
630 'R:\SAHP_Matlab_MPC_Type1163_Sub-Sim_WithPlaceholders.dck'); % Make
    required Substitutions in the placeholder deck.
                ForecastInputFiles1(TimeBounds); %Pull the relevant weather
631 and draw profile information.

632 writematrix(SubSimControlSeries(:,3:4),'R:\InputFiles\ExtractedControls.c
    sv'); % Put SubSimControlSeries into a temporary TRNSYS input file.
                status=system('"C:\Trnsys18\Exe\TrnEXE64.exe"
633 "R:\SAHP_Matlab_MPC_Type1163_Sub-Sim_WithPlaceholders_Replaced.dck" /n');
    % Call a new instance of TRNSYS to run the sub-simulation deck.
634             mFileErrorCode = 155.3;
635             if(status~=0)
636                 error("error in most recent TRNSYS CALL");
637             end

    [CurrentSubSimResults,StagnationErrors,HPUndertempErrors,SystemOvertempEr
    rors,UndertempDrawErrors]=Post_Process1('R:\MPC_Output\MPC_Output.csv','R
638 :\MPC_Output\DHW_Temps.csv', trnTimeStep, history.L_S, history.L_A,
    SubSimControlSeries,toPlot); % PostProcess the sub-simulation output
    calculating integrals, detecting faults, etc.
639             if(toPlot)
640                 figure(1);
641                 %subplot(2,1,1);

642 plot(CurrentSubSimResults.TIME,CurrentSubSimResults.T_DHW_AVG,'--')

643 %plot(CurrentSubSimResults.TIME,CurrentSubSimResults.DHW_Tank_Energy_kJ_,
    '--')
644                 hold on
645                 %subplot(2,1,2)
                    %
646 plot(CurrentSubSimResults.TIME,max(CurrentSubSimResults.HP_Source_Flowrat
    e,CurrentSubSimResults.HX_SourceFlowRate),'--')
647                 %hold on
648                 %subplot(2,1,1)
649                 figure(2)
650                 subplot(2,1,1)

651 plot(CurrentSubSimResults.TIME,CurrentSubSimResults.Controller_Mode,'--')
652                 hold on;
653                 subplot(2,1,2)
```

```
654
plot(CurrentSubSimResults.TIME,CurrentSubSimResults.AuxHeat_Call,'--')
655                 hold on;
656                 figure(3)

657 plot(CurrentSubSimResults.TIME,CurrentSubSimResults.DHW_Available_Energy_
    kJ_,'--')
658             end

659 destination=sprintf('R:\\MPC_Output\\MPC_Output_%d_%d.csv',floor(trnTime)
    ,i); % Set a path to save the output file from the sub-simulation.
660             movefile('R:\MPC_Output\MPC_Output.csv', destination); % Save
    the sub-simulation output file to a unique path.
                movefile('R:\MPC_Output\MPC_Output.mat',
661 strrep(destination,'.csv','.mat')); % Save the sub-simulation output file
    to a unique path.
662             % Update the control series
663             OldSubSimControlSeries=SubSimControlSeries;
664             mFileErrorCode = 155.4;
665             %Replace Stagnating times in control series with HX
    operation:
666             if(~isempty(StagnationErrors))

667 SubSimControlSeries(StagnationErrors(:,1):StagnationErrors(:,2),3)=1;
668             end
669             % Parse HPUndertempErrors and lock the HP out during periods
670             % identified.
671             if(~isempty(HPUndertempErrors))

672 SubSimControlSeries(HPUndertempErrors(:,1):HPUndertempErrors(:,2),5)=0; %
    Lock the HP out.

673 SubSimControlSeries(:,3)=SubSimControlSeries(:,3).*SubSimControlSeries(:,
    5); % Apply the lock to the current control series.
674             end
675             % Parse the Cold Draw Event to determine how energy deficient
676             % it is, and then with assumptions about HX and HP
    Performance,
677             % decide how LONG of a HP or HX run should be added BEFORE
    the
678             % cold draw.
679             mFileErrorCode = 155.5;
680             if(~isempty(UndertempDrawErrors))
681                 additionalHeat = +5000;
682                 EnergyDeficit =
DrawEnergyDeficit(CurrentSubSimResults,UndertempDrawErrors);
```

```
683 %SubSimControlSeries(UndertempDrawErrors(:,1):UndertempDrawErrors(:,2),3)
    =SubSimControlSeries(UndertempDrawErrors(:,1):UndertempDrawErrors(:,2),3)
    *2;
                HX_Edot_Addition =
684 CurrentHXSubSimResults.HX_LoadFlowRate.*Cp.*(CurrentHXSubSimResults.HX_T_
    LoadOut-CurrentHXSubSimResults.HX_T_LoadIn);
                HP_Edot_Addition =
685 CurrentHPSubSimResults.HP_Load_Flowrate.*Cp.*(CurrentHPSubSimResults.HP_T
    Load_Out-CurrentHPSubSimResults.HP_TLoad_In);
686             DeltaEdot_HPHX=HP_Edot_Addition-HX_Edot_Addition;
687             % Apply HP Lockout to DeltaEdot_HPHX
688             DeltaEdot_HPHX = DeltaEdot_HPHX
    .*SubSimControlSeries(:,5);
689             % out timesteps where HP is already on.
                DeltaEdot_HPHX(SubSimControlSeries(:,3)==2)=0; % Examine
690 DeltaEdot and the previous controlseries and zero out timesteps where HP
    is already on
691             mFileErrorCode = 155.6;
692             for l=1:1:size(UndertempDrawErrors,1)
                    AdditionalAuxPotential=
693 10735*trnTimeStep*(UndertempDrawErrors(l)-
    sum(SubSimControlSeries(1:UndertempDrawErrors(l),4)));
                    currentDeltaEdot =
694 flip(DeltaEdot_HPHX(1:UndertempDrawErrors(l)));
                    currentDeltat =
695 flip(CurrentHPSubSimResults.TIME(1:UndertempDrawErrors(l)));
                    currentDeltaE = -
696 cumtrapz(currentDeltat,currentDeltaEdot);
                    HP_Sufficient =
697 max(currentDeltaE)>(EnergyDeficit(l)+additionalHeat);
                    HPandAUX_Sufficient = (max(currentDeltaE) +
698 AdditionalAuxPotential)>(EnergyDeficit(l)+additionalHeat);
699                 mFileErrorCode = 155.7;
700                 if(HP_Sufficient)
                        numHPTimesteps =
701 find(currentDeltaE>(EnergyDeficit(l)+additionalHeat),1);
702                     numAUXTimesteps = 0;
703                 elseif(HPandAUX_Sufficient)
                        [HPPotential,numHPTimesteps] =
704 max(currentDeltaE,[],'omitnan'); %The maximum extra energy that HP can
    deliver.
705                     if HPPotential<=0
706                         numHPTimesteps=0;
707                     end
                        numAUXTimesteps =
708 ((EnergyDeficit(l)+additionalHeat)-HPPotential)/(10735*trnTimeStep) ;
    %The rest of the energy needs to come from AUXHeat
```

```
709                             numAUXTimesteps = ceil(numAUXTimesteps);
710                         else
                                [HPPotential,numHPTimesteps] =
711 max(currentDeltaE,[],'omitnan'); %The maximum extra energy that HP can
    deliver.
712                             if HPPotential<=0
713                                 numHPTimesteps=0;
714                             end
                                numAUXTimesteps = UndertempDrawErrors(l)-1 ; % If
715 HP and AUX are not sufficient, then AUX should be on for the whole sim
    and HP for whenever it can run.
716                         end
717                         mFileErrorCode = 155.8;
718                         if(numHPTimesteps>=UndertempDrawErrors(l))
719                             numHPTimesteps = UndertempDrawErrors(l)-1;
720                         end
721                         if(numAUXTimesteps>=UndertempDrawErrors(l))
722                             numAUXTimesteps = UndertempDrawErrors(l)-1;
723                         end
724                         mFileErrorCode = 155.9;
725                         if(numHPTimesteps~=0)
                                SubSimControlSeries(UndertempDrawErrors(l)-
726 numHPTimesteps:UndertempDrawErrors(l),3)=2; % Replace the requisite
    number of timesteps with HP operation.

    SubSimControlSeries(:,3)=SubSimControlSeries(:,3).*SubSimControlSeries(:,
727 2).*SubSimControlSeries(:,5); % Continue to restrict HP operation to
    daylight hours (and any other restrictions that may be imposed elsewhere
    by adjusting SubSimControlSeries(:,2).
728                         end
729                         if(numAUXTimesteps~=0)

    if(all(SubSimControlSeries(UndertempDrawErrors(l)-
730 numAUXTimesteps:UndertempDrawErrors(l),4) == 0)) % Check if the whole
    block of contiguous AUX control series up to the undertemp error are 0
    (and therefore can be easily replaced with 1).
                                    SubSimControlSeries(UndertempDrawErrors(l)-
731 numAUXTimesteps:UndertempDrawErrors(l),4)=1; % Simply drop the required #
    of 1s into the control series
732                             else % Need to find 0's to replace with 1s.
                                    targetAUXSteps =
733 find(SubSimControlSeries(1:UndertempDrawErrors(l),4)==0,numAUXTimesteps,'
    last');
734                                 SubSimControlSeries(targetAUXSteps,4)=1;
735                             end
```

```
736                     end
737                     mFileErrorCode = 155.101;
738                 end
739                 mFileErrorCode = 155.102;
740                 %                 for l=1:1:size(UndertempDrawErrors,1)
741 %                     currentDeltaEdot =
flip(DeltaEdot_HPHX(1:UndertempDrawErrors(l)));
742 %                     currentDeltat =
flip(CurrentHPSubSimResults.TIME(1:UndertempDrawErrors(l)));
743 %                     currentDeltaE = -
cumtrapz(currentDeltat,currentDeltaEdot);
744 %                     numTimestepstoReplace =
find(currentDeltaE>EnergyDeficit(l),1)+60;
745 %
746 %
747 %
replacementStart=UndertempDrawErrors(l)-numTimestepstoReplace; % Start at
the begining of the fault - the number of HP steps required
748 %                     replacementEnd =
UndertempDrawErrors(l);
749 %                     AuxStart=[];
750 %                     AuxEnd = [];
751 %                     % Check if there are enough
elements to replace before
752 %                     % the beginning of the simulation
753 %
DaylightStart=find(SubSimControlSeries(:,2)==1,1);
754 %
DaylightEnd=find(SubSimControlSeries(:,2)==0,1,'last');
755 %
756 %                     if isempty(numTimestepstoReplace) %
The HP will likely not meet the load.
757 %
758 %                     numTimestepstoReplace =
length(currentDeltaE)
759 %
760 %
761 %                     end
762 %
763 %                     if DaylightEnd<DaylightStart % This
means that the simulation spans a sunrise but not a sunset.
764 %                         DaylightEnd = [];
765 %                     end
766 %
767 %                     if (~isempty(replacementStart)) %
If there is a need for additional heat (HP or AUX)
```

```
768                 %                               if(isempty(DaylightStart)) % If
    the entire simulation is outside daylight
769                 %                                   AuxStart =
    replacementStart; % replace with AUX instead of HP
770                 %
    AuxEnd=UndertempDrawErrors(l);
771                 %                                   replacementStart = [];
772                 %                                   replacementEnd = [];
773                 %                               end
774                 %
775                 %                               if replacementStart<DaylightEnd
                    %
776 if(UndertempDrawErrors(l)>DaylightEnd) % Required HP spans outside of
    daylight hours.
                    %                                   replacementStart =
777 replacementStart  - (UndertempDrawErrors(l)-DaylightEnd); % Shift HP
    Collection earlier in the day.
778                 %                                   replacementEnd =
    DaylightEnd;
779                 %                                   AuxStart =[];
780                 %                                   AuxEnd=[];
781                 %                                   end
782                 %                               end
783                 %
                    %                               if replacementStart<1 % if the
784 required start of HP is before the beginning of the subsim, then HP and
    AUX must overlap
                    %                                   AuxStart =
785 max([UndertempDrawErrors(l)-abs(replacementStart);1]);
786                 %                                   AuxEnd
    =UndertempDrawErrors(l);
787                 %                                   replacementStart = 1;
788                 %                                   replacementEnd =
    UndertempDrawErrors(l);
789                 %                               end
790                 %
                    %                               if
791 replacementStart<=DaylightStart % if the required start of HP is before
    the beginning of the Daylight then some additional AUX is required
792                 %                                   AuxStart =
    UndertempDrawErrors(l)-(DaylightStart-replacementStart);
793                 %                                   AuxEnd =
    UndertempDrawErrors(l);
794                 %                                   replacementStart =
    DaylightStart;
795                 %                                   replacementEnd =
    UndertempDrawErrors(l);
796                 %                               end
```

```
797                 %
798                 %
799                 %
800                 %
801                 %
802                 %                           if ~isempty(replacementStart)
                    %
803 SubSimControlSeries(replacementStart:replacementEnd,3)=SubSimControlSerie
    s(replacementStart:replacementEnd,2)*2;
804                 %                           end
805                 %                           if ~isempty(AuxStart)
                    %
806 SubSimControlSeries(AuxStart:AuxEnd,4)=1;
807                 %                           end
808                 %
809                 %                       end
810                 %
811                 %
812                 %                   end
813                 mFileErrorCode = 155.103;
814             end
815             mFileErrorCode = 155.104;
816             i=i+1; % Increment Counter
817             %             if(toPlot)
818             %         figure(2);
                    %
819 plot(SubSimControlSeries(:,1),SubSimControlSeries(:,3), '.b');
820             %         hold on;
                    %
821 plot(SubSimControlSeries(:,1),SubSimControlSeries(:,4), '.m');
822             %             end
                if i>20 ||
823 isequal(OldSubSimControlSeries,SubSimControlSeries) % Test for
    Convergence or exceeding max iterations
824                 converged = true;
825                 afterFirstSubSim = true;
826             end
827         end
828         if(toPlot)
829             figure(1);
830             %subplot(2,1,1);
```

```
831             plot(history.trnTime(nStep_LastPlot:nStep-1),history.T_DHW_AVG(nStep_LastPlot:nStep-1),'-b','DisplayName','Main Simulation Path');
832             %plot(history.trnTime(nStep_LastPlot:nStep-1),history.DHWTankEnergy(nStep_LastPlot:nStep-1),'-b','DisplayName','Main Simulation Path');
833
834             %plot(history.trnTime(nStep_LastPlot:nStep-1),history.DHWAvailableEnergy(nStep_LastPlot:nStep-1),'-b');
835             %plot(history.trnTime(nStep_LastPlot:nStep-1),history.HP_Power(nStep_LastPlot:nStep-1),'-b');
836             %plot(history.trnTime(nStep_LastPlot:nStep-1),history.CumulativeSolarFraction(nStep_LastPlot:nStep-1),'-b');
837             hold on;
838             set(gca,'ColorOrderIndex',1)
839             figure(2)
840             subplot(2,1,1)
841             plot(history.trnTime(nStep_LastPlot:nStep-1),history.Controller_Mode(nStep_LastPlot:nStep-1),'-b');
842             hold on;
843             set(gca,'ColorOrderIndex',1)
844             subplot(2,1,2)
845             plot(history.trnTime(nStep_LastPlot:nStep-1),history.AuxHeat_Call(nStep_LastPlot:nStep-1),'-b');
846             hold on;
847             set(gca,'ColorOrderIndex',1)
848             figure(3)
849             plot(history.trnTime(nStep_LastPlot:nStep-1),history.DHWAvailableEnergy(nStep_LastPlot:nStep-1),'-b');
850             hold on;
851             set(gca,'ColorOrderIndex',1)
852         end
853         %     legend('Main Simulation Path', 'Sub Sim Control 1','Sub Sim Control 2',...
854         %         'Sub Sim Control 3','Sub Sim Control 4','Sub Sim Control 5',...
855         %         'Sub Sim Control 6','Sub Sim Control 7','Stagnation', 'HP Undertemp',...
856         %         'System Overtemp', 'Undertemp Draw');
857         %         drawnow();
858         nStep_LastPlot = nStep;
859         %SubSimControlSeries(:,3)=SubSimControlSeries(:,2)*1;
860         ControlArray = SubSimControlSeries(:,[1,3,4]);
861     end
862     if (~exist('ControlArray'))
```

```
863         mFileErrorCode = 155.1;
864         Controller_Mode=0; % System mode=0 ==> HP and HX and Circulators
Off.
865         Pump1_Signal =0;
866         Pump2_Signal =0;
867         Diverter1_Signal=0;
868         Diverter2_Signal=0;
869         HP_CALL =0;
870         N_Compressor=0;
871         AuxHeat_Call =0;
872     else
873         % Find the current time index relative to control array
874
875         [~,TimeIndex] = min(abs(trnTime-(ControlArray(:,1))));
876         % --- Make Controls Decisions here ---
877         %Essentially, all of the control logic below will need to be
removed or
878         %commented out, and array index lookups will need to be done
(possilby
879         %interpolating time steps) to deliver once-per-timestep control
signals
880         %that are stored in the array that resulted from the sorting.
881         %Charging Mode Decisions
882         if ((ControlArray(TimeIndex,2)==0) &&
(SubSimControlSeries(TimeIndex,2)==1) && ((HX_TSourceIn< (T_DHW_BOT+2))
|| (HP_TSource_In < 10)))
883             Demanded_Mode = 3; % Call for HP Warmup in the above
conditions.
884         else
885             Demanded_Mode = ControlArray(TimeIndex,2); % Use the control
array that was assembled.
886         end
887         AuxHeat_Enable = ControlArray(TimeIndex,3);
888         if (max([T_DHW_TOP, T_DHW_MID, T_DHW_BOT])>=95) % CSA Safety
requirement that all charging means are off if water temperature reaches
99C
889             Controller_Mode=0; % System mode=0 ==> HP and HX and
Circulators Off.
890             Pump1_Signal =0;
891             Pump2_Signal =0;
892             Diverter1_Signal=0;
893             Diverter2_Signal=0;
894             HP_CALL =0;
895             N_Compressor=0;
```

```
896             %
elseif(T_DHW_Thermostat>(T_DHW_Solar_Set+T_Deadband)) %Tank is above  the
solar setpoint, no heat required.
897             %             Controller_Mode=0; % System mode=0 ==> HP and
HX and Circulators Off.
898             %             Pump1_Signal =0;
899             %             Pump2_Signal =0;
900             %             Diverter1_Signal=0;
901             %             Diverter2_Signal=0;
902             %             HP_CALL =0;
903             %             N_Compressor=0;
904             %         elseif(T_DHW_Thermostat>=T_DHW_Solar_Set) %Tank top
is within deadband, no control change.
905             %             % Do Nothing
                %         %elseif(nnz(trnTime > SunriseTimes+SunriseSetDelay
906 & trnTime < SunsetTimes-SunriseSetDelay)) % Tests if the current
simulation time is between the hours of sunrise and sunset for any day of
the year.
907         else
908             % Calling for heat and within daylight hours
909             % Decide on mode based on the control array:
                if(trnTime-tLastControlChange>=t_timeout) ||
910 (Controller_Mode==2 && Demanded_Mode~=2) %Check if enough time has
elapsed to change Controller_Mode or if the required change is to turn HP
Off
911                 if(Demanded_Mode==1)
912                     % Activate HX Mode (Mode1) and set circulators and
diverters.
913                     if (Controller_Mode~=1)
914                         tLastControlChange = trnTime;
915                     end
916                     Controller_Mode=1;
917                     Pump1_Signal =HX_Pump_Setting;
918                     Pump2_Signal =HX_Pump_Setting;
919                     Diverter1_Signal=HX_Diverter_Setting;
920                     Diverter2_Signal=HX_Diverter_Setting;
921                     HP_CALL =0;
922                     N_Compressor=0;
923                 elseif (Demanded_Mode==2)
924                     % Activate HP Mode (Mode2) and set circulators and
diverters.
925                     if (Controller_Mode~=2)
926                         tLastControlChange = trnTime;
927                     end
```

```
928                     Controller_Mode=2;
929                     Pump1_Signal =HP_Pump_Setting;
930                     Pump2_Signal =HP_Pump_Setting;
931                     Diverter1_Signal=HP_Diverter_Setting;
932                     Diverter2_Signal=HP_Diverter_Setting;
933                     HP_CALL =1;
934                     if (ModulateCompressor==0)
935                         N_Compressor=1;
936                     else
937                         if (To_COL<10)
938                             N_Compressor=0.625;
939                         elseif (To_COL>=10 && To_COL<20)
940                             N_Compressor=0.75;
941                         else
942                             N_Compressor=1;
943                         end
944                     end
945                 elseif  ((Demanded_Mode==3) && (T_SolarSource<95))
946                     % Activate HP Warmup Mode (Mode3) and set circulators
and diverters.
947                     if (Controller_Mode~=3)
948                         tLastControlChange = trnTime;
949                     end
950                     Controller_Mode=3;
951                     Pump1_Signal =HP_Pump_Setting;
952                     Pump2_Signal =0;
953                     Diverter1_Signal=HP_Diverter_Setting;
954                     Diverter2_Signal=HP_Diverter_Setting;
955                     HP_CALL =0;
956                     N_Compressor=1;
957                 else % Everything Off
958                     if (Controller_Mode~=0)
959                         tLastControlChange = trnTime;
960                     end
961                     Controller_Mode=0; % System mode=0 ==> HP and HX and
Circulators Off.
962                     Pump1_Signal =0;
963                     Pump2_Signal =0;
964                     Diverter1_Signal=0;
965                     Diverter2_Signal=0;
```

```
966                     HP_CALL =0;
967                     N_Compressor=0;
968                 end
969             end
970         end
971         % AUX Charging Decisions
972         % 0=Heater locked out, 1=Heater enabled
973         if (max([T_DHW_TOP, T_DHW_MID, T_DHW_BOT])>=95) % CSA Safety requirement that all charging means are off if water temperature reaches 99C
974             AuxHeat_Call =0;
975         elseif(AuxHeat_Enable==0)
976             AuxHeat_Call =0;
977         elseif (AuxHeat_Enable==1)
978             if (T_DHW_Thermostat>(T_DHW_AUX_Set+T_Deadband)) %Tank is above  the AUX setpoint, no heat required.
979                 AuxHeat_Call =0;
980             elseif (T_DHW_Thermostat<T_DHW_AUX_Set) %Tank is below the AUX setpoint, call for heat.
981                 AuxHeat_Call =1;
982             end
983         end
984     end
985     trnOutputs(1) = Controller_Mode;
986     trnOutputs(2) = Pump1_Signal;
987     trnOutputs(3) = Pump2_Signal;
988     trnOutputs(4) = Diverter1_Signal;
989     trnOutputs(5) = Diverter2_Signal;
990     trnOutputs(6) = HP_CALL;
991     trnOutputs(7) = N_Compressor;
992     trnOutputs(8) = AuxHeat_Call;
993 end
994 %plot(history.trnTime(nStep_LastPlot:nStep),history.DHWTankEnergy(nStep_LastPlot:nStep),'-b','DisplayName','Main Simulation Path');
995 %nStep_LastPlot = nStep;
996 mFileErrorCode = 0; % Tell TRNSYS that we reached the end of the m-file without errors
997 return
```

The following MATLAB function “TankState80gal” is called by “MPC_Controller_IterativelyBuildControlSeries.m” to perform the integration of tank nodes to determine the amount od energy contained in the DHW tank and how much of that energy is stored in water above 35°C.

```
1   function [TankEnergy, AvailableEnergy, UnavailableEnergy] =
    TankState80gal(TempProfile)
2   %TankState80Gal is a function that inspects the temperature
    distribution of
3   %an 80 US-gallon DHW Storage tank and dermines the quantity of water
    above
4   %35C and the z-coordinate that corresponds to that temperature. The
5   %funcction then spatially interpolates the temperature profile to
    return 3
6   %energy quantities: Total energy, AvailableEnergy (The energy
    contained in
7   %the water above 35C, and UnavailableEnergy, the energy contained in
    the
8   %water below 35C.
9
10  % Physical Dimensions of the tank
11  V= 0.3028; % The volime of the tank in m^3
12  h = 1.524; % The height of the tank in m
13  A = V/h; % The Cross Sectional Area in m^2
14  z_coords=linspace(0,h,length(TempProfile));
15  % Assume constant properties for Water evaluated at 25C and 1 atm.
16
17  Cp=4.19; % The specific heat of water in kJ/kgK
18  rho=1000; % The density of water in kg/m^3
19
20
21  TankEnergy = trapz(z_coords,TempProfile)*rho*Cp*A;
22  ThresholdNode= find(TempProfile>35,1,"last");
23  if (isempty(ThresholdNode) || ThresholdNode == 1 || ThresholdNode ==
    0)
24  AvailableEnergy=0;
25  else
26  AvailableEnergy =
    trapz(z_coords(1:ThresholdNode),TempProfile(1:ThresholdNode))*rho*Cp*A;
27  end
28  UnavailableEnergy = TankEnergy-AvailableEnergy;
29
30  end
```

The following MATLAB function “SubSimDeck1.m” is called by “MPC_Controller_IterativelyBuildControlSeries.m” to arrange a TRNSYS deck file for each sub-simulation with initial conditions that match the current state of the top-level simulation.

```
function [] = SubSimDeck1(TimeBounds, T_DHW_Profile, T_SolarSource, DeckFile_With_Placeholders)
%SUBSIMDECK1 Accepts a 20 node DHW Tank Temperature profile vector
%'T_DHW_Profile' and the filename of a TRNSYS deck with suitable
%placeholders inserted 'DeckFile_With_Placeholders'.  The function
%replacess the placeholders in the deck file with the temperatures from the
%input profile.

fid = fopen(DeckFile_With_Placeholders,'rt') ;
X = fread(fid) ;
fclose(fid) ;
X = char(X.') ;
Y = strrep(X, '[StartTime]', sprintf('%f',TimeBounds(1))) ;
Y = strrep(Y, '[StopTime]', sprintf('%f',TimeBounds(2))) ;
Y = strrep(Y, '[DHW T_init_1]', sprintf('%f',T_DHW_Profile(1))) ;
Y = strrep(Y, '[DHW T_init_2]', sprintf('%f',T_DHW_Profile(2))) ;
Y = strrep(Y, '[DHW T_init_3]', sprintf('%f',T_DHW_Profile(3))) ;
Y = strrep(Y, '[DHW T_init_4]', sprintf('%f',T_DHW_Profile(4))) ;
Y = strrep(Y, '[DHW T_init_5]', sprintf('%f',T_DHW_Profile(5))) ;
Y = strrep(Y, '[DHW T_init_6]', sprintf('%f',T_DHW_Profile(6))) ;
Y = strrep(Y, '[DHW T_init_7]', sprintf('%f',T_DHW_Profile(7))) ;
Y = strrep(Y, '[DHW T_init_8]', sprintf('%f',T_DHW_Profile(8))) ;
Y = strrep(Y, '[DHW T_init_9]', sprintf('%f',T_DHW_Profile(9))) ;
Y = strrep(Y, '[DHW T_init_10]', sprintf('%f',T_DHW_Profile(10))) ;
Y = strrep(Y, '[DHW T_init_11]', sprintf('%f',T_DHW_Profile(11))) ;
Y = strrep(Y, '[DHW T_init_12]', sprintf('%f',T_DHW_Profile(12))) ;
Y = strrep(Y, '[DHW T_init_13]', sprintf('%f',T_DHW_Profile(13))) ;
Y = strrep(Y, '[DHW T_init_14]', sprintf('%f',T_DHW_Profile(14))) ;
Y = strrep(Y, '[DHW T_init_15]', sprintf('%f',T_DHW_Profile(15))) ;
Y = strrep(Y, '[DHW T_init_16]', sprintf('%f',T_DHW_Profile(16))) ;
Y = strrep(Y, '[DHW T_init_17]', sprintf('%f',T_DHW_Profile(17))) ;
Y = strrep(Y, '[DHW T_init_18]', sprintf('%f',T_DHW_Profile(18))) ;
Y = strrep(Y, '[DHW T_init_19]', sprintf('%f',T_DHW_Profile(19))) ;
```

```
33 Y = strrep(Y, '[DHW T_init_20]', sprintf('%f',T_DHW_Profile(20))) ;
34 Y = strrep(Y, '[T_SolarSource]', sprintf('%f',T_SolarSource)) ;
35 fid2 = fopen(strrep(DeckFile_With_Placeholders,'.dck','_Replaced.dck')
   ,'wt') ;
36 fwrite(fid2,Y) ;
37 fclose (fid2) ;
38
39
40 end
```

Appendix E
Tank State Estimation and Kalman Filtering

E.1 Objectives and Scope

The control of a solar-assisted-heat-pump system relies on accurate knowledge of both the state of the solar resource defined in terms of the amount and the character of solar irradiation, as well as the state of charge of a stratified domestic hot water tank defined by the temperature profile contained within the tank. The majority of the research in this thesis relies on stratified storage tank models in the TRNSYS system model discussed previously. While the previously discussed models have their application in research and development, an alternative to the TRNSYS tank model that requires less computational power, and without software licensing restrictions might be required if a SAHP system controller were to be commercialized. This motivates the work presented here, which investigates an approach to tank state estimation using an Unscented Kalman Filter (UKF). The use of a UKF for tank state estimation has been presented by Kreuzinger et al. [53] in conference proceedings.

E.2 Background

The coefficient of performance of a heat pump is known to vary as a function of the inlet temperatures and mass flow rates of the working fluid on the load side at its condenser, and the source side at its evaporator. Statistical and analytical models of solar irradiation, as well as weather forecast data and direct observation of current solar conditions can be used to satisfy that half of the control decision inputs, however, the temperature profile of a thermal storage tank must be estimated based on all of the previous conditions and activities that have led up to the present moment.

The goal of the current study is to produce an estimate of the state of thermal charge of a thermal storage tank using the fewest number of measurements and relatively little computation power. As such, solving this problem with a sophisticated CFD model of the tank, or a large array of temperature measurements are eliminated as possible solutions. This situation mimics the real-word design constraint of a consumer SDHW system, which would need to be as low-cost as possible and may not be designed to provide for multiple temperature measurements within the tank. State estimation will be achieved by combining a simplified analytical model of the tank behavior under charging conditions with observations that are made in real-time to reduce the error, and the uncertainty of the state predictions that would otherwise be made with the tank model.

E.2.1 Stratified Thermal Storage Model

A common characteristic of solar-thermal systems is a high degree of stratification in the storage tanks. That is to say, mixing of the water inside of the tank is not promoted. As a result, the tank temperature is highest at the top, and lowest at the bottom. It is also the case, that the temperature gradient within the tank is non-constant, so the temperature profile is not uniformly distributed in the z-direction.

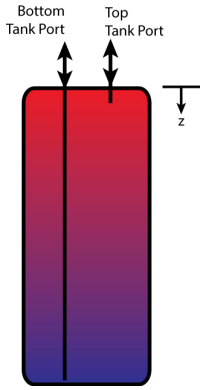


Figure E-1: Stratified Thermal Storage

Equation E-1 below is a partial differential equation that relates the differential temperature, time, and the z-position throughout a thermal storage tank such as the one pictured in Figure E-1 [53].

$$\rho(T)c_p(T)\frac{\partial T}{\partial t} = \lambda_w(T)\frac{\partial^2 T}{\partial z^2} - \rho(T)c_p(T)u(t)\frac{\partial T}{\partial z} - \frac{4k_w(T,u)}{d_i}\Delta T_{amb} \qquad (E-1)$$

In the above equation, ρ represents the density of water in tank [kg/m^3], c_p is the Specific Heat Capacity of water kJ/kg, u is the Plug Flow Velocity m/s which is calculated as the volumetric flow rate of water in the system, divided by the area of an axial slice of the tank. The other parameters include, λ_w, which is the thermal conductivity of Water $W/m°C$, k_w, the effective conduction coefficient of tank walls $W/m°C$, and ΔT_{amb}, the temperature difference across the tank wall $°C$.

In this work, tank charging is modeled by a positive plug flowrate convention, with higher temperature water being introduced at the tank's top port, and relatively cooler water being removed from the tank's bottom port.

In order to conveniently treat this problem numerically, the PDE in equation E-1 can be discretized in both the time and the z-domain, yielding a system of nonlinear ODEs as shown in Equation E-2 [53]. Note that below, $T_i(t_k) = T(z_i, t_k)$, where the tank is divided into N elements and $i = 1, \ldots, N$.

$$\frac{dT_i(t=t_k)}{dt} = \frac{\lambda_w}{\rho c_p}\frac{T_{i+1}(t) - 2T_i(t) + 2T_{i-1}}{\Delta z^2} - u(t)\frac{T_{i+\frac{1}{2}}(t) - T_{i-\frac{1}{2}}(t)}{\Delta z} - \frac{4k_w}{d_i \rho c_p}(T_i(t) - T_{amb}) (E-2)$$

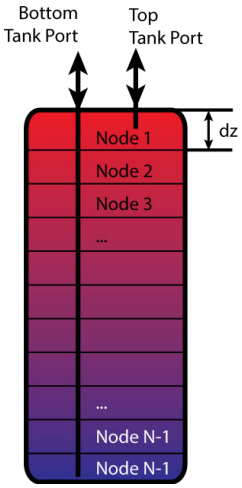


Figure E-2: Discretized tank model showing N nodes

As discussed in Kreuzinger et al., in order to recover sharp changes in the spatial distribution of temperature within the tank, a slope limiting scheme known as the "Superbee limiters" were applied to the differential equation tank model above. This is seen by the incorporation of the $T_{i+\frac{1}{2}}(t)$ and the $T_{i-\frac{1}{2}}(t)$ terms. These terms are calculated as follows for u>0 [53]:

$$T_{i-\frac{1}{2}} = T_{i-1} + \frac{\Phi_{i-\frac{1}{2}}}{2}(T_i - T_{i-1}),$$
$$T_{i+\frac{1}{2}} = T_i + \frac{\Phi_{i+\frac{1}{2}}}{2}(T_{i+1} - T_i) \quad (E-3)$$

where Φ is defined as:

$$\Phi(\theta) = \max(0, \min(1, 2\theta), \min(2, \theta))$$

and θ is defined as:

$$\theta_{i-\frac{1}{2}} = \frac{T_{i-1}(t) - T_{i-2}(t)}{T_i(t) - T_{i-1}(t)}, \qquad \theta_{i+\frac{1}{2}} = \frac{T_i(t) - T_{i-1}(t)}{T_{i+1}(t) - T_i(t)} \tag{E-4}$$

E.2.2 Unscented Kalman Filter

Kalman filtering is a special application of Bayesian updating to a non-stationary problem. The general operating principle of such schemes is to provide recursive calculations that predict and update the state of a system given time inputs, state transition models, measurement models, and observations. Figure E-3 below shows the general sequence of operation of a Bayesian filter as described in Kaipio and Somersalo [54].

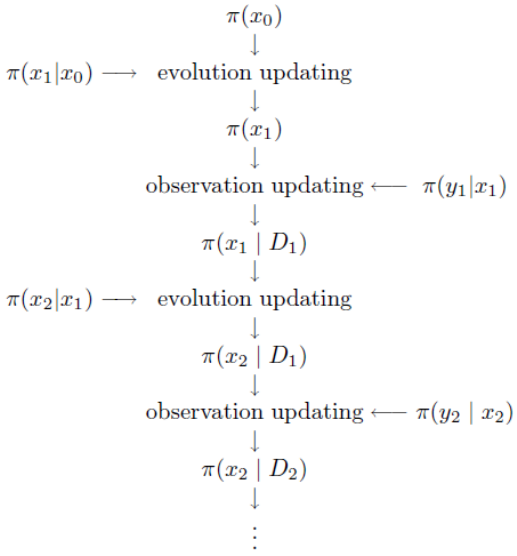


Figure E-3: Bayesian filter sequence of operation [54]

The original Kalman filtering (KF) algorithm relies of the following assumptions to be true [54]:

1. The state equations are linear with additive noise processes.
2. The noise vectors are Gaussian with known means and covariances.
3. The noise vectors are mutually independent.
4. The probability distribution of the state vector is known and Gaussian.

The first condition in the above list does not hold for the system being studied, since the system of ODEs is non-linear. Extended Kalman filters (EKF) and Particle filters (PF) are techniques that can be used on non-linear systems, and the latter can be used when the probability distributions are not known. EKFs require the system to be expressed in closed-form equations so that Jacobians can be calculated at each state point. PFs can be used more generally, but they are more computationally costly to implement as they use a number of sample "particles" to represent the probability distribution. Unscented Kalman Filters (UKF) are an alternative to EKF and PF that uses a set of "Sigma Points" to track the mean and covariance of the probability distribution of the state without providing information about the actual shape of the distribution.

Chapter 7 of *Kalman Filtering and Neural Networks* by Wan and van der Merwe [55], as well as the video lecture by James McNames [56] provide the following description of the UKF algorithm:

First, initialize with $\widehat{\boldsymbol{x}}_0$, an initial estimate of the state vector, and P_0, an initial estimate of the covariance of the state.

$$\widehat{\boldsymbol{x}}_0 = E[\boldsymbol{x}_0] \tag{E-5}$$

$$P_0 = \left\|\boldsymbol{x}_0 - \widehat{\boldsymbol{x}}_{0|0}\right\|^2 \tag{E-6}$$

Next, for $k = 1,2,3, \ldots$

calculate the sigma points as follows:

$$\boldsymbol{\mathcal{X}}_{\mathrm{k-1}} = \left[\widehat{\boldsymbol{x}}_{k-1}\ \widehat{\boldsymbol{x}}_{k-1} + \gamma\sqrt{\boldsymbol{P}_{k-1}}\ \widehat{\boldsymbol{x}}_{k-1} - \gamma\sqrt{\boldsymbol{P}_{k-1}}\right] \tag{E-7}$$

perform the time Update (Prediction Step):

$$\boldsymbol{\mathcal{X}}^{\star}_{\mathrm{k|k-1}} = F(\boldsymbol{\mathcal{X}}_{k-1}, u_{k-1}) \tag{E-8}$$

where F is the state transition function (externally defined and specific to the system), u is a vector of additional inputs that are not a part of the state vector.

$$\widehat{\boldsymbol{x}}_{\boldsymbol{k}}^{-} = \sum_{i=0}^{2L} W_i^m\, \mathcal{X}^{\star}_{\mathrm{i,k|k-1}} \tag{E-9}$$

$$\boldsymbol{P}_k^{-} = \sum_{i=0}^{2L} W_i^c \left(\mathcal{X}^{\star}_{\mathrm{i,k|k-1}} - \widehat{\boldsymbol{x}}_k^{-}\right)\left(\mathcal{X}^{\star}_{\mathrm{i,k|k-1}} - \widehat{\boldsymbol{x}}_k^{-}\right)^T + \boldsymbol{R^v} \tag{E-10}$$

augment Sigma Points after transition to new time-step:

$$\boldsymbol{\mathcal{X}}_{\mathrm{k|k-1}} = \left[\boldsymbol{\mathcal{X}}^{\star}_{\mathrm{k|k-1}} \;\; \boldsymbol{\mathcal{X}}^{\star}_{\mathrm{0|k-1}} + \gamma\sqrt{\boldsymbol{R^{v}}} \;\; \boldsymbol{\mathcal{X}}^{\star}_{\mathrm{0|k-1}} - \gamma\sqrt{\boldsymbol{R^{v}}}\;\right] \qquad (E-11)$$

$$\boldsymbol{\mathcal{Y}}_{\mathrm{k|k-1}} = \boldsymbol{H}\left(\boldsymbol{\mathcal{X}}_{k|k-1}\right) \qquad (E-12)$$

$$\widehat{\boldsymbol{y}}_{\boldsymbol{k}}^{-} = \sum_{i=0}^{2L} W_i^m \mathcal{Y}_{i,k|k-1} \qquad (E-13)$$

perform Measurement-Update (Correction Step):

$$\boldsymbol{P}_{\tilde{\boldsymbol{y}}_{\boldsymbol{k}}\tilde{\boldsymbol{y}}_{\boldsymbol{k}}} = \sum_{i=0}^{2L} W_i^c\left(\mathcal{Y}_{i,k|k-1} - \widehat{\boldsymbol{y}}_{\boldsymbol{k}}^{-}\right)\left(\mathcal{Y}_{i,k|k-1} - \widehat{\boldsymbol{y}}_{\boldsymbol{k}}^{-}\right)^{T} + \boldsymbol{R^{n}} \qquad (E-14)$$

$$\boldsymbol{P}_{\boldsymbol{x}_{\boldsymbol{k}}\boldsymbol{y}_{\boldsymbol{k}}} = \sum_{i=0}^{2L} W_i^c\left(\mathcal{X}_{i,k|k-1} - \widehat{\boldsymbol{x}}_{\boldsymbol{k}}^{-}\right)\left(\mathcal{Y}_{i,k|k-1} - \widehat{\boldsymbol{y}}_{\boldsymbol{k}}^{-}\right)^{T} \qquad (E-15)$$

$$\mathcal{K}_k = \boldsymbol{P}_{\boldsymbol{x}_{\boldsymbol{k}}\boldsymbol{y}_{\boldsymbol{k}}}\boldsymbol{P}^{-\mathbf{1}}_{\tilde{\boldsymbol{y}}_{\boldsymbol{k}}\tilde{\boldsymbol{y}}_{\boldsymbol{k}}} \qquad (E-16)$$

$$\widehat{\boldsymbol{x}}_{\boldsymbol{k}} = \widehat{\boldsymbol{x}}_{\boldsymbol{k}}^{-} + \mathcal{K}_k(\boldsymbol{y}_{\boldsymbol{k}} - \widehat{\boldsymbol{y}}_{\boldsymbol{k}}^{-}) \qquad (E-17)$$

$$P_k = P_k^{-} - \mathcal{K}_k\boldsymbol{P}_{\tilde{\boldsymbol{y}}_{\boldsymbol{k}}\tilde{\boldsymbol{y}}_{\boldsymbol{k}}}\mathcal{K}_k^{T} \qquad (E-18)$$

In the above, $\gamma = \sqrt{L+\lambda}$, λ is the composite scaling parameter, L is the dimension of the state, $\boldsymbol{R^{v}}$ is the process-noise covariance, $\boldsymbol{R^{n}}$ is the measurement-noise covariance, and W_i are the weights calculated in equation E-19 below:

$$W_i^m = W_i^c = \frac{1}{2(L+\lambda)} \qquad (E-19)$$

The Unscented Kalman Filter is implemented in MATLAB R2019b as a part of the control system toolbox. For the remainder of this study, this implementation was used.

E.3 Method

E.3.1 Experimentation

The ETU was reconfigured for the current study. The DHW storage tank was coupled directly to an immersion heater through a short plumbing loop with a variable speed pump, flow meter, and immersed thermocouples inline with the flow. Within the storage tank are 5 thermocouples spaced

evenly from the top to the bottom. Figure E-4 shows a schematic of the apparatus that was used to conduct the tank charging experiment.

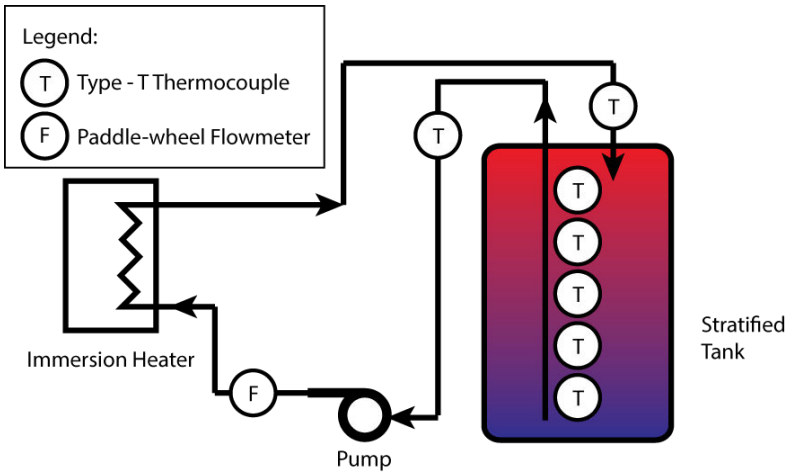


Figure E-4: Schematic representation of the experimental apparatus

An experiment was conducted by filling the system with water at approximately $9°C$ initially. The water in the system was thoroughly mixed by running the circulator pump at maximum speed for 20 minutes with the heater deactivated. The water was then allowed to settle for a further 20 minutes before the test was commenced. An automatic data acquisition system was set to record all of the thermocouple temperatures, as well as the flow rate of the water at 30 second intervals.

At the commencement of the test period, the circulation pump was activated at a flow rate of approximately 3 kg/min. The circulation heater was activated at full power delivering approximately 4100 W of heat to the water. The system was allowed to run for 3 hours and 38 minutes with data acquisition taking place as the tank charged.

The data collected during the experiment included the following data intended to be used in making tank state predictions:

- Incoming Flow Rate $\dot{m}$ [kg/min]
- 1 Mid-stratification Field Tank Temperature [$°C$]
- Incoming and Outgoing Flow Temperature [$°C$]

As well as the following data that were reserved for validation of the tank state predictions:

- Array of 5 Temperatures spaced within the stratification field.

In addition to the experiment described above, a numerical simulation of the tank charging from the same initial conditions and the same "u" inputs was conducted and will be described below.

E.3.2 Simulation

A numerical simulation of the tank charging from approximately 9°C and with the same flow-rate and inlet temperatures per time-step as were observed in the experiment described above. The simulation was based on the time and z-discretized model of the tank that was presented in Equation E-2. A simple finite-difference approach was employed to arrange Equation E-2 as a state transition function which is required by the UKF algorithm. The state transition function is a function that takes as inputs, the previous state $\widehat{\boldsymbol{x}}_{\boldsymbol{k-1}}$ of the system at time t_{k-1}, as well as the u input from the previous time-step and returns as an output the predicted state of the system, $\widehat{\boldsymbol{x}}_{\boldsymbol{k}}$, at the current time ($t_k$). The finite-difference state equation can be seen below in Equation E-20, where $T_{i,k} = T(z = z_i, t = t_k)$:

$$T_{i,k} = T_{i,k-1} + \Delta t \frac{\lambda_w}{\rho c_p} \frac{T_{i+1,k-1} - 2T_{i,k-1} + 2T_{i-1,k-1}}{\Delta z^2} - u(t) \frac{T_{i+\frac{1}{2},k-1} - T_{i-\frac{1}{2},k-1}}{\Delta z} - \frac{4k_w}{d_i \rho c_p}\left(T_{i,k-1} - T_{amb}\right) (E-20)$$

Prior to the application of the UKF, the finite difference simulation was run for the purposes of testing the model. The simulation was run with 30 second time steps to match the sampling frequency of the experiment, from t=0s to t=13140s or approximately 3 hours, 36 minutes.

E.3.3 Data Processing

As the main analysis method for this study, the UKF was applied to the state estimation problem discussed above. The implementation of UKF for this problem is represented in the flowchart shown in Figure E-5 below.

The temperature profiles were stored as an array of the form: $T = \begin{bmatrix} T_{z_1,t_1} & T_{z_1,t_2} & \cdots & T_{z_1,t_k} \\ T_{z_2,t_1} & T_{z_2,t_2} & \cdots & T_{z_2,t_k} \\ \vdots & \vdots & \ddots & \vdots \\ T_{z_N,t_1} & T_{z_N,t_2} & \vdots & T_{z_N,t_k} \end{bmatrix}$, where column-vectors from the array represent the state of the tank at a given time-step.

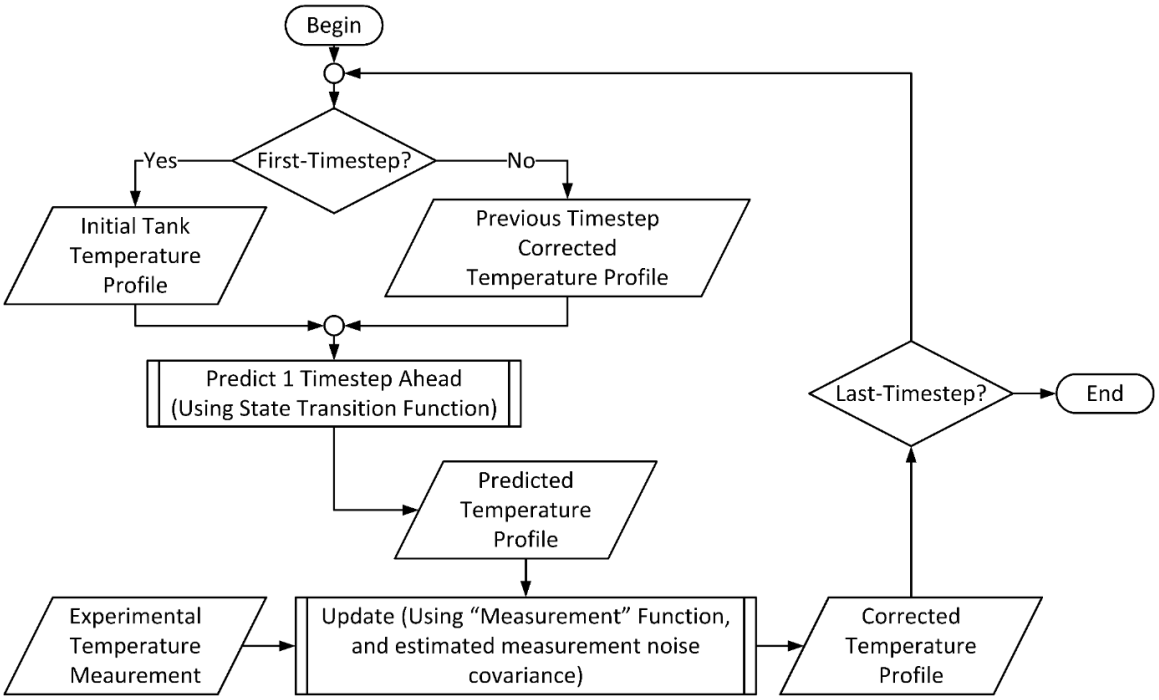


Figure E-5: Calculation process using UKF to estimate the state of a thermal storage tank

The prediction step shown in the flowchart above calls a custom MATLAB function, "StateTransition.m", that implements Equations E-3, E-4, and E-20. The update step also calls the UKF object in MATLAB, which generates and transforms the sigma points, tracking the mean and covariance changes as the state transitions from one time step to another.

The update step shown in the flowchart above calls a custom MATLAB function entitled "measurement.m", which interpolates between the temperatures contained in the tank elements to find the approximate temperature at a location 0.435m below the top of the tank. This location corresponds to the placement location of the internal tank thermocouple that is used to update the prediction made by the earlier step of the algorithm. It also calls the UKF object in MATLAB which uses the covariance matrices and calculates weightings using a method similar to Equation E-19. In the update step, the predicted state is combined with an experimental temperature measurement to bring the estimate of the state closer to the actual state of the tank at that juncture.

Following the update step, the data output is saved in a new column in the *T* array before being passed back to the beginning of the recursion loop where the prediction step is executed again and the

algorithm continues until it reaches the endpoint of the supplied data. All of the source code used in the data analysis for this report can be seen in the appendix to this document.

E.4 Results

The temperature profiles that were generated as a result of both the simple finite difference simulation, as well as the UKF combined simulation-experimental data are displayed in the surface plots in Figure E-6.

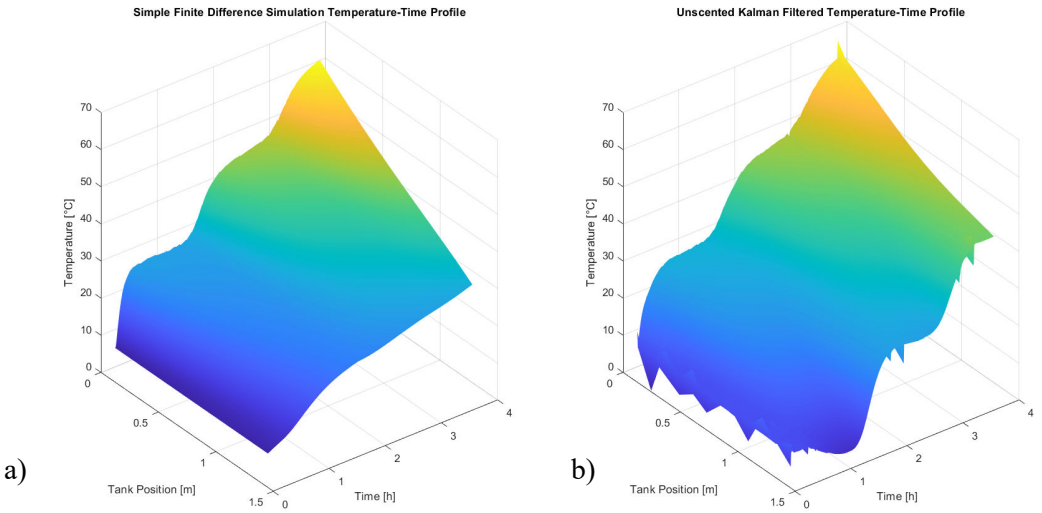


Figure E-6: Surface plots of tank temperature profile generated by a) Simple finite difference simulation; and b) UKF combination of simulation and experimentally measured temperatures

Comparing the figures above, it can be seen that the temperature at the very top of the tank is relatively unchanged between the unfiltered and filtered results. This is likely because the top tank node is where the inlet water is supplied. This water has spent the least amount of time in the tank compared with the water in any other node in the tank. As a result, there has been less time for error in the predicted temperature of the top-node water to accumulate from the time where it was last measured at the tank inlet.

On the other hand, it can be seen that the farther down within the tank, and also the farther forward in time, the greater the effect that the UKF filtering has on the temperature profile. Take, for example the bottom tank temperature, represented by the edge of the plots parallel to the time axis, and closest

to the observer. In these profiles, it can be seen that after approximately the first 1 hour of simulation, the profile predicted by the unfiltered scheme seems to have over predicted the diffusion within the tank and is therefore over predicting the tank bottom node temperature. On the other hand, the UKF filtered results appear to be better characterizing the stratification of the water, keeping the energy concentrated near the top of the tank and predicting a cooler bottom temperature.

At approximately 1.5 hours of simulation, and again at approximately 3.5 hours of simulation, the gradient of temperature in the tank with respect to time is very steep. This can be attributed to a front of hot water moving down at the plug-flow velocity and replacing the relatively cooler bottom-tank water which is flowing out of the outlet port at the bottom of the tank. The UKF filtered results better capture these hot and cold fronts moving down through the tank.

An animation was generated to show a time-step by time-step comparison of the filtered and unfiltered results. It can be seen at: https://youtu.be/98CZisFkg-M

A second animation was generated that demonstrates the development of error in the predicted state of the tank as compared with temperatures measured using the 5 thermocouples distributed inside of the tank. It can be viewed at: https://youtu.be/Kb7QdkyIX_0

The final frame from the error tracking animation can be seen below for the purposes of discussion in Figure E-7.

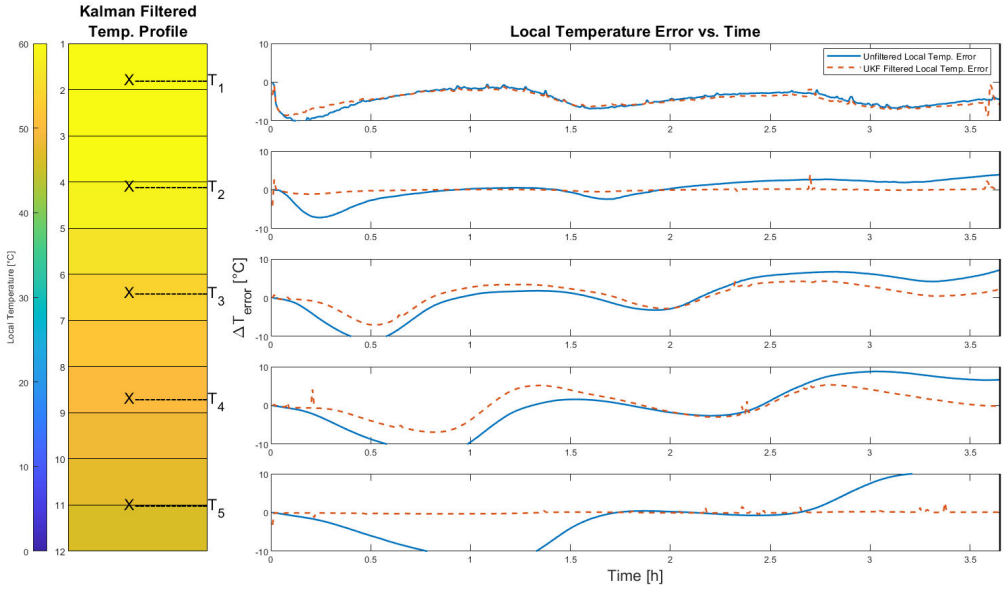


Figure E-7: Local Temperature Error vs. Time at 5 stations within the stratification field

In the above figure, it can be seen that the top station for temperature comparison sees very little improvement as a result of the UKF filtering. It is suggested here that the Kalman filter's inability to affect the temperature nearest to the top of the tank is due to the fact that the tank top temperature is largely set by the continuously varying inlet flow temperature, which is not a part of the state vector.

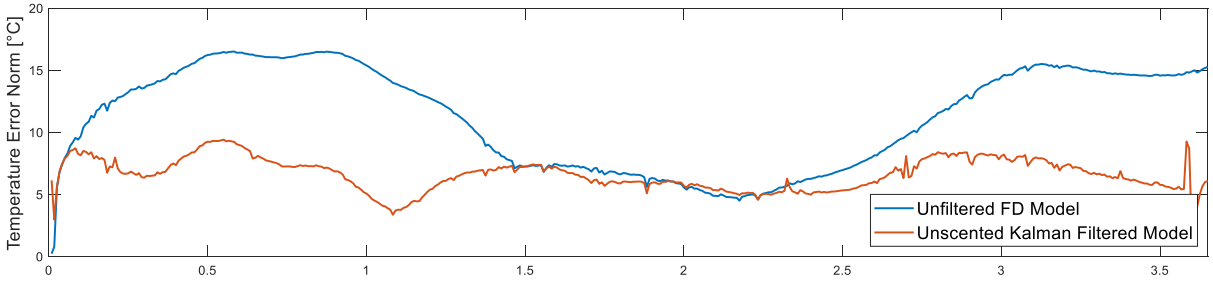
At temperature station 2, it can be seen that the UKF has an impact on the error in predicted temperature, reducing the error from on the order of $5°C$ at t=3.5 hours down to essentially $0°C$. The UKF is most effective at this temperature measurement station because this is the location where the one mid-field experimental temperature measurement update point is located that is incorporated into the estimated temperature profile by the UKF. The selection of this location was arbitrary, but it had to coincide with one of the thermocouples that was already a part of the ETU.

Farther down in the tank, at stations 3 and 4, it can be seen that the temperature error is that of an under prediction of the local tank temperature for the first half of the time domain investigated, followed by an over-prediction of the tank temperature in the second half of the time-range. In both cases, it can be seen that the UKF predicted local temperatures deviate from the actual temperatures by less than $\pm 5°C$, while the unfiltered predictions go out of range on the plots, indicating that there are local errors of greater than $\pm 10°C$. For the unfiltered data, it can clearly be seen that the amount of local error increases greatly as the z-position within in the tank increases (higher towards the bottom of the tank). Finally, at temperature measurement station 5, the UKF filtered predictions have a relative error down at nearly $0°C$ similar to that seen at station 2. This is due to the UKF taking into account the measured outlet tank temperature as a second "update" input. This result clearly demonstrates that the error in the prediction of local temperature within the tank varies inversely with the proximity of the prediction location to the measurement "update" location. The amount of error that accumulates as the prediction location moves farther away from the measurement "update" is a function of how well the state transition model captures the physics of the problem. In the case of this project, it is likely that the model is missing terms related to natural convection within the tank, as well as plumes and mixing that develop at the inlet/outlet as a result of abrupt velocity changes. The discretization of the model into 12 lumped nodes also likely contributes a significant error to the prediction that the UKF cannot correct with the limited information available in the measurement update step.

Figure E-8 below shows a) the norm of the local temperature errors and b) the average of the local temperature errors that were shown individually in Figure E-7. Figure E-8 a) shows that the unfiltered temperature predictions have an error norm that oscillates between approximately $5°C$ and $15°C$. With the UKF filtering the results, it can be seen that the error norm of the temperatures at the validation points is consistently below $10°C$, and averages around $6.6°C$ over the 3.63 hour test.

Figure E-8 b) shows that the spatially-averaged temperature error for the unfiltered results is first an under-prediction, and after approximately 2 hours, becomes an over prediction of tank temperatures. This sort of trend where the internal tank temperatures grow larger with each time-step can be attributed to the model under predicting the heat loss from the tank, for example, by overestimating the thermal resistance of the tank walls. This sort of error appears to be very well compensated-for by the UKF implemented in this project. Throughout the time-range studied, the average temperature error observed in the UKF filtered state estimates ranges from $-3.3°C$ to $2.3°C$. This level of spatially averaged error represents approximately a 50% reduction in the range of the temperature errors.

a)



b)

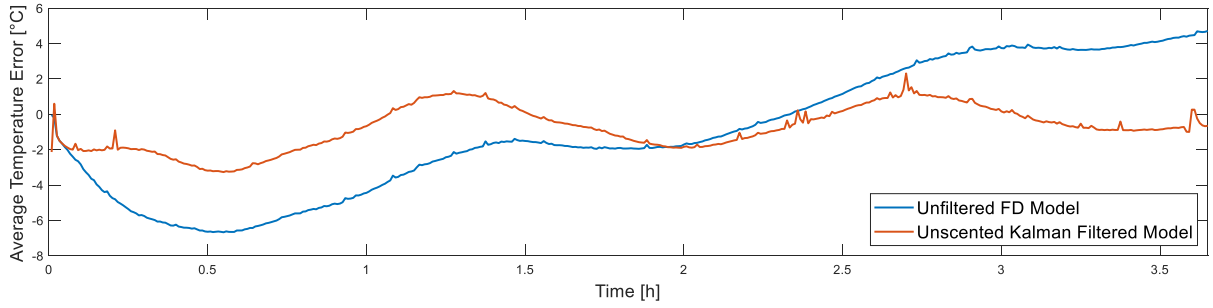


Figure E-8: a) Norm of the local temperature errors vs time; b) Average of the local temperature errors vs. time

E.5 Conclusions

In summary, the problem of state estimation for a stratified thermal storage tank is important for the design of a SAHP control system that would make control decisions based on both incident irradiation on a collector and the state of thermal charge of a DHW tank. A system to measure and estimate the state of thermal charge would likely be restricted in the number of temperature sensors that could be implemented. The Unscented Kalman Filter, a method that can be used to combine measured and predicted data into a better estimate of the state of a system has been investigated in respect of its application to an SAHP-DHW system.

It was seen in the results section that there is significant error that develops when a finite-difference simulation is carried out without any updating. The errors accumulate both in time as the simulation progresses and also in the z-dimension of the tank as the location of interest moves away from the tank inlet.

In the case of an UKF filtered simulation, where a small amount of experimental data is used to update the prediction of the model, the error can be significantly reduced throughout the entire domain, however the benefits of the "updating" are noticed most prevalently near the locations where the experimental measurement updates are made. Reductions in local temperature error of more than $15°C$ were observed at some time instants and tank locations, but more often the benefits were on the order of reductions by $3°C$. These differences represent percent error reductions of 25% and 5% respectively.

There are multiple sources of error inherent in the model used for the state transition function which include both discretization errors, as well as assumptions that neglect modes of heat transfer and simplify the flow-physics of the problems. The UKF, with its experimental measurement updates is able to compensate for some errors, particularly those that are additive within the time-domain. On the other hand, the UKF does a poor job of compensating for errors that accumulate in the z-domain, and as a result, the error in the state estimate is larger as the location of interest moves farther from the location where update measurements were made.

Recommendations for future work include refinement of the state transition model so that it may better capture the problem physics. In particular, the model was discretized into 12 nodes in the z-direction for this project. This was done because of a limitation in the plug-flow model imposed by the plug-flow velocity and the time step of the simulation. If the size of the nodes, Δz, is smaller than

the distance that a "plug" of flowing fluid would travel down in the tank during one time-step, then the simulation diverges from a physical answer. Interpolating the supplied data in time is one refinement that could be used to increase the number of tank nodes, then a grid dependence study could be performed to determine the size of node, Δz, at which the simulation is no longer sensitive to mesh density. The same could be performed for the size of time step used in the simulations.

A second recommendation for future work would be to vary the location, and potentially the number of experimental update measurements within the tank. The arbitrary location selected for the update measurement in this project was likely not optimal and a parametric study could be conducted to find the best location for the sensor in a production system.

In conclusion, the UKF has been demonstrated to have a positive effect in the reduction of error in the estimated state of a stratified thermal storage tank. The average error in the predicted state may be sufficiently small to base control decisions on, however, there is significant room for improvement in the state estimates provided by this method. Some means of improving these estimates were listed above.

E.6 Listing of MATLAB Functions and Scripts for Kalman Filtering Study

UKFTank.m

```
% UFKTank.m
% This script imports data taken from the ETU in ERC3009, creates an UKF
% object using the Matlab implementation of the unscented kalman filter,
% and then proceeds to predict, update, and recurse through each timestep
% contained within the data to generate a dataset of temperature profiles
% distributed in time.  This script should be run before
% Tank_Finite_Difference.m and ErrorTrack.m
% Nonstationary Estimation
% Author: Julian Howarth
% email: julian.howarth@uwaterloo.ca
% Created: 2019-11-21; Last revision: 2019-11-25
clc;clear; close all;
%% Import Experimental Data
% Mass Flow 1 Average (kg/min)  Ambient Air  (C)    DTank HotOut  (C)
DTank ColdIn (C)    DTank ValveOut  (C) DTank HotIn (C) DTank ColdOut
(C)  DTank 5 Top  (C)    DTank 4  (C)    DTank 3 Middle  (C) DTank 2 (C)
DTank 1 Bottom  (C)
```

```
data=xlsread('Inverse_Project\Charging 9C to 60C at3 kg-min_Extra_cols_removed.xls');
m_dot = data(:,1)'./60;
T_amb = data(:,2)';
T_inl= data(:,3)';
T_outl= data(:,7)';
T_obs = data(:,8:end)';
max_k = length(T_obs);

%% Set up Finite Difference Simulation
di = 0.51; % Inner diameter of the tank [m]
A=(pi*di^2)/4;
z = 1.45; % Inner Height of the tank [m]
kw= 0.1;% Heat conduction coefficient of the walls of the tank.
P = 5; % Absolute Pressure,  [bar].
numel = 12;
dz=z/numel;
dt = 30;% s;
T_Prof_initial = ones(numel,1).*9; % Initial temperature profile in tank is uniform 9C;
T_Prof1 = zeros(numel, max_k);

%% Simulate (And Animate Simulation)

obj = unscentedKalmanFilter(@StateTransition,@Measurement,T_Prof_initial,'ProcessNoise',0.1,'MeasurementNoise',0.005);%'ProcessNoise',.1,'MeasurementNoise',.0001);
%obj.StateCovariance=obj.StateCovariance.*0;
for k=1:1:max_k
  %correct(obj,T_obs(2,k),dz);

%correct(obj,[T_obs(1,k);T_obs(2,k);T_obs(3,k);T_obs(4,k);T_obs(5,k)],dz,T_inl(k),T_outl(k));
  correct(obj,[T_obs(2,k);T_obs(5,k)],dz,T_inl(k),T_outl(k));
  predict(obj,m_dot(k),T_inl(k),T_outl(k),T_amb(k),numel,P,dt,dz,di,kw);
  T_Prof1(:,k)= obj.State;
  T_Cov(k)=norm(obj.StateCovariance);

end
```

Tank_Finite_Difference.m

```
%Tank_Finite_Difference.m
% This script is used to plot (and record video animation) of the
% devlopment of the temperature profile in a stratified theremal storage
% tank being analysed.  The script simulates the charging of the tank
using
% a simple finite difference simulation.  It also uses a set of
temperature
% profile-time data generated by UKFTank.m and a set of 5 temperature
% measurements from within an experimental tank setup to plot a
% size-by-size comparison that is saved in the frame-animation.
% Author: Julian Howarth
% email: julian.howarth@uwaterloo.ca
% Created: 2019-11-21; Last revision: 2019-11-25
clc;close all;%clear variables;
%close(myVideo);
% Set whether or not to make video or plot;
Plot = true;
Video = true;
%% Import Experimental Data
% Mass Flow 1 Average (kg/min)  Ambient Air  (C)    DTank HotOut  (C)
DTank ColdIn (C)    DTank ValveOut  (C) DTank HotIn (C) DTank ColdOut
(C)  DTank 5 Top  (C)    DTank 4  (C)    DTank 3 Middle  (C) DTank 2 (C)
DTank 1 Bottom  (C)
data=xlsread('Inverse_Project\Charging 9C to 60C at3 kg-
min_Extra_cols_removed.xls');
m_dot = data(:,1)'./60;
T_amb = data(:,2)';
T_inl= data(:,3)';
T_outl= data(:,7)';
T_obs = data(:,8:end)';
max_k = length(T_obs);
%% Animate Data
%for i=1:1+length(T_obs)
% figure(1);
% h=imshow(T_obs(1,:)')
% colormap(ax1,hot)
%end
%% Set up Finite Difference Simulation
di = 0.51; % Inner diameter of the tank [m]
A=(pi*di^2)/4;
```

```
z = 1.45; % Inner Height of the tank [m]
kw= 0.1;% Heat conduction coefficient of the walls of the tank.
P = 5; % Absolute Pressure,  [bar].
numel = 12;
dz=z/numel;
dt = 30;% s;
T_Prof_initial = ones(numel,1).*9; % Initial temperature profile in tank is uniform 9C;
T_Prof = zeros(numel, max_k);

%% Simulate (And Animate Simulation)
if (Plot == true && Video == true)
myVideo = VideoWriter('Tanks.mp4','MPEG-4'); %open video file
myVideo.FrameRate = 30;  %can adjust this, 5 - 10 works well for me
open(myVideo)
end
for k=1:1:max_k
    if k==1
        T_prof_last_k = T_Prof_initial;
    else
        T_prof_last_k = T_Prof(:,k-1);
    end
    T_Prof(:,k) = StateTransition(T_prof_last_k,m_dot(k),T_inl(k),T_outl(k),T_amb(k),numel,P,dt,dz,di,kw);
    u=m_dot(k)/(XSteam('rho_pT',P,T_inl(k))*(A));

    for i=1:1:numel
        cp= XSteam('Cp_pT',P,T_prof_last_k(i));% Specific Heat Capacity of liquid water at atmospheric pressure.
        rho= XSteam('rho_pT',P,T_prof_last_k(i));% density of liquid water at atmospheric pressure.
        lambda= XSteam('tc_pT',P,T_prof_last_k(i));% Heat conduction coefficient of liquid water at atmospheric pressure.
        if i==1
            [T_plus_half,T_minus_half] = T_plus_minus_half(T_prof_last_k,i,numel,u,T_inl(k),T_outl(k));
            T_Prof(i,k)= T_prof_last_k(i) + dt*(-u*((T_plus_half-T_minus_half)/dz)-((4*kw)/(di*rho*cp))*(T_prof_last_k(i)-T_amb(k)));

        elseif i == numel
            [T_plus_half,T_minus_half] = T_plus_minus_half(T_prof_last_k,i,numel,u,T_inl(k),T_outl(k));
```

```
            T_Prof(i,k)= T_prof_last_k(i) + dt*(-u*((T_plus_half-
69  T_minus_half)/dz)-((4*kw)/(di*rho*cp))*(T_prof_last_k(i)-T_amb(k)));
70
71          else
                [T_plus_half,T_minus_half] =
72  T_plus_minus_half(T_prof_last_k,i,numel,u,T_inl(k),T_outl(k));
            T_Prof(i,k)= T_prof_last_k(i) + dt*(
    (lambda/(rho*cp))*((T_prof_last_k(i+1)-
    2*T_prof_last_k(i)+T_prof_last_k(i-1))/dz^2)-u*((T_plus_half-
73  T_minus_half)/dz)-((4*kw)/(di*rho*cp))*(T_prof_last_k(i)-T_amb(k)));
74          end
75
76      end
77      if Plot == true
78      figure(1);
        set(gcf, 'Units', 'Normalized', 'OuterPosition', [0, 0.04, .7,
79  0.96]);    hold on;
80      subplot(1,3,1)
81      pcolor([T_Prof(:,k),T_Prof(:,k)]);
82      set(gca, 'YDir','reverse');
83      set(gca,'xtick',[]);
84      colorbar();
85      caxis([0 60])
86      title({'Finite Difference','Simulation'},'FontSize',14 )
87      ylabel('Z-Node # in Tank', 'FontSize',14)
88      subplot(1,3,2)
89      pcolor([T_obs(:,k),T_obs(:,k)]);
90      set(gca, 'YDir','reverse');
91      set(gca,'xtick',[]);
92      colorbar();
93      caxis([0 60])
94      title({'Observed','Temp. Profile'},'FontSize',14 )
95      subplot(1,3,3)
96      pcolor([T_Prof1(:,k),T_Prof1(:,k)]);
97      set(gca, 'YDir','reverse');
98      set(gca,'xtick',[]);
99      h=colorbar();
100     caxis([0 60])
101     title({'Kalman Filtered','Temp. Profile'},'FontSize',14 )
102     h = colorbar;
103     ylabel(h, ['Local Temperature [',char(176),'C]'],'FontSize',14)
104     if k==1
```

```
    ylimits=get(gca,'ylim');
    xlimits=get(gca,'xlim');
    t=text(xlimits(2)-1.5,ylimits(2)+0.5,{['','','Time =
',num2str(dt*k),' sec']},'FontSize',14 );    set(t,
'horizontalAlignment', 'left');
    hold on;
    elseif(mod(k,4)==0)
    if ((dt*k)<=59)
    t.String={['','','Time = ',num2str(dt*k),' sec']};
    elseif ((dt*k)<=3559)
    mins=floor(dt*k/60);
    secs=(dt*k)-(mins*60);
    t.String={['','','Time = ',num2str(mins),' min ']};%,num2str(secs),'
sec']};
    else
    hrs=floor(dt*k/3600);
    mins=floor(dt*k/60-hrs*60);
    secs=(dt*k)-(hrs*3600)-(mins*60);
    t.String={['','','Time = ',num2str(hrs),' hrs ',num2str(mins),' min
']};
    end
    end
    drawnow()
    if Video==true
    frame = getframe(gcf); %get frame
    writeVideo(myVideo, frame);
    end
    end
end
if (Plot == true && Video == true)
close(myVideo)
end
```

StateTransition.m

```
function [T_Prof] =
StateTransition(T_prof_last_k,m_dot,T_inl,T_outl,T_amb,numel,P,dt,dz,di,kw)
%STATETRANSITION State transition function of a z-discrete stratified
%thermal storage tank to be used with an UKF and discrete-time data.
% StateTransition.m
% email: julian.howarth@uwaterloo.ca
% Created: 2019-11-21; Last revision: 2019-11-25

```

```
    u=m_dot/(XSteam('rho_pT',P,T_inl)*((pi*di^2)/4));

for i=1:1:numel
    cp= XSteam('Cp_pT',P,T_prof_last_k(i));% Specific Heat Capacity of
liquid water at atmospheric pressure.
    rho= XSteam('rho_pT',P,T_prof_last_k(i));% density of liquid water at
atmospheric pressure.
    lambda= XSteam('tc_pT',P,T_prof_last_k(i));% Heat conduction
coefficient of liquid water at atmospheric pressure.
    if i==1
        [T_plus_half,T_minus_half] =
T_plus_minus_half(T_prof_last_k,i,numel,u,T_inl,T_outl);
        T_Prof(i)= T_prof_last_k(i) + dt*(-u*((T_plus_half-
T_minus_half)/dz)-((4*kw)/(di*rho*cp))*(T_prof_last_k(i)-T_amb));



    elseif i == numel
        [T_plus_half,T_minus_half] =
T_plus_minus_half(T_prof_last_k,i,numel,u,T_inl,T_outl);
        T_Prof(i)= T_prof_last_k(i) + dt*(-u*((T_plus_half-
T_minus_half)/dz)-((4*kw)/(di*rho*cp))*(T_prof_last_k(i)-T_amb));

    else
        [T_plus_half,T_minus_half] =
T_plus_minus_half(T_prof_last_k,i,numel,u,T_inl,T_outl);
        T_Prof(i)= T_prof_last_k(i) + dt*(
(lambda/(rho*cp))*((T_prof_last_k(i+1)-2*T_prof_last_k(i)+T_prof_last_k(i-
1))/dz^2)-u*((T_plus_half-T_minus_half)/dz)-
((4*kw)/(di*rho*cp))*(T_prof_last_k(i)-T_amb));
    end
    %T_Prof = T_prof_last_k;
end
end
```

Plot3dfigs.m

```
% Plot3dfigs.m
% This script takes processed data from UKFTank.m and
% Tank_Finite_Difference.m and produces 3d temperature vs tank height vs
% time profiles.
% email: julian.howarth@uwaterloo.ca
% Created: 2019-12-02; Last revision: 2019-12-02
close all;
set(gcf, 'Units', 'Normalized', 'OuterPosition', [0, 0.04, 1, 0.96]);
hold on;
subplot(1,2,1)
```

```
hold off;
surf([dt:dt:dt*max_k]./3600,[dz:dz:z],T_Prof)
title('Simple Finite Difference Simulation Temperature-Time Profile')
zlabel(['Temperature [',char(176),'C]']);
ylabel('Tank Position [m]');
set(gca, 'YDir','reverse')
xlabel('Time [h]')
shading interp
subplot(1,2,2)
hold off;
surf([dt:dt:dt*max_k]./3600,[dz:dz:z],T_Prof1)
zlabel(['Temperature [',char(176),'C]']);
ylabel('Tank Position [m]');
set(gca, 'YDir','reverse')
xlabel('Time [h]')
title('Unscented Kalman Filtered Temperature-Time Profile')
```

Measurement.m

```
function [Tmeas] = Measurement(T_predict,dz,T_inl,T_outl)
%MEASUREMENT Measurement update function of a z-discrete stratified
%thermal storage tank to be used with an UKF and discrete-time data.
% Measurement.m
% Author: Julian Howarth
% email: julian.howarth@uwaterloo.ca
% Created: 2019-11-21; Last revision: 2019-11-25
zs=[0.145 0.435 0.725 1.0125 1.305];
numel=length(T_predict);
Tmeas = [];
for j=[2,5]
pos=(zs(j)-(dz/2))/dz;
nodeUp = floor(pos);
nodeDn = ceil(pos);
weightDn= pos-nodeUp;
weightUp = 1-weightDn;
if nodeUp <1
Tmeas = [Tmeas, T_inl*weightUp+T_predict(nodeDn)*weightDn];
elseif nodeDn > numel
Tmeas = [Tmeas,T_predict(nodeUp)*weightUp+T_outl*weightDn];
else
Tmeas = [Tmeas,T_predict(nodeUp)*weightUp+T_predict(nodeDn)*weightDn];
```

```
end
end

Tmeas=Tmeas';
end
```

ErrorTrack.m

```
% ErrorTrack.m
% This script is used to plot (and record video animation) of the error
% between local temperatures estimated at 5 stations throughout a
% stratified thermal storage tank, and the 5 validation temperatures in
% the supplied dataset.  UFKTank.m and Tank_Finite_Difference.m should be
% run before this script.
% Author: Julian Howarth
% email: julian.howarth@uwaterloo.ca
% Created: 2019-11-21; Last revision: 2019-11-25
close all;
Tmeas_UKF = zeros(5, max_k);
Tmeas_FD = zeros(5, max_k);
zs=[0.145 0.435 0.725 1.0125 1.305];
for k = 1:1:max_k

for j=1:1:5
pos=(zs(j)-(dz/2))/dz;
nodeUp = floor(pos);
nodeDn = ceil(pos);
weightDn= pos-nodeUp;
weightUp = 1-weightDn;
if nodeUp <1
Tmeas_UKF(j,k) = T_inl(k)*weightUp+T_Prof1(nodeDn,k)*weightDn;
elseif nodeDn > numel
Tmeas_UKF(j,k) = T_Prof1(nodeUp,k)*weightUp+T_outl(k)*weightDn;
else
Tmeas_UKF(j,k) = T_Prof1(nodeUp,k)*weightUp+T_Prof1(nodeDn,k)*weightDn;
end
end

end

T_Error_UKF=T_obs-Tmeas_UKF;
```

```
34
35  for k = 1:1:max_k
36
37  for j=1:1:5
38  pos=(zs(j)-(dz/2))/dz;
39  nodeUp = floor(pos);
40  nodeDn = ceil(pos);
41  weightDn= pos-nodeUp;
42  weightUp = 1-weightDn;
43  if nodeUp <1
44  Tmeas_FD(j,k) = T_inl(k)*weightUp+T_Prof(nodeDn,k)*weightDn;
45  elseif nodeDn > numel
46  Tmeas_FD(j,k) = T_Prof(nodeUp,k)*weightUp+T_outl(k)*weightDn;
47  else
48  Tmeas_FD(j,k) = T_Prof(nodeUp,k)*weightUp+T_Prof(nodeDn,k)*weightDn;
49  end
50  end
51
52  end
53
54  T_Error_FD=T_obs-Tmeas_FD;
55
56  for k=1:1:max_k
57      T_Error_Norm_UKF(k) = norm(T_Error_UKF(:,k));
58      T_Error_Norm_FD(k)  = norm(T_Error_FD(:,k));
59      T_Error_mean_UKF(k) = mean(T_Error_UKF(:,k));
60      T_Error_mean_FD(k)  = mean(T_Error_FD(:,k));
61  end
62  clf
63  figure(1)
    set(gcf, 'Units', 'Normalized', 'OuterPosition', [0, 0.04, 1, 0.96]);
64  hold on;
65  subplot(2,1,1)
66  plot([1:1:k]*dt/3600,T_Error_Norm_FD,'-','LineWidth',1.5);
67  xlim([0 max_k*dt/3600]);
68  hold on;
69  plot([1:1:k]*dt/3600,T_Error_Norm_UKF,'-','LineWidth',1.5);
    legend({'Unfiltered FD Model', 'Unscented Kalman Filtered
70  Model'},'Location','southeast','FontSize',14)
71  ylabel(['Temperature Error Norm [',char(176),'C]'],'FontSize',14)
```

```
subplot(2,1,2)
plot([1:1:k]*dt/3600,T_Error_mean_FD,'-','LineWidth',1.5);
xlim([0 max_k*dt/3600]);
hold on;
plot([1:1:k]*dt/3600,T_Error_mean_UKF,'-','LineWidth',1.5);
ylabel(['Average Temperature Error [',char(176),'C]'],'FontSize',14)
xlabel('Time [h]','FontSize',14)
legend({'Unfiltered FD Model', 'Unscented Kalman Filtered Model'},'Location','southeast','FontSize',14)
figure(3)
myVideo = VideoWriter('Error_track.mp4','MPEG-4'); %open video file
myVideo.FrameRate = 30;  %can adjust this, 5 - 10 works well for me
open(myVideo)
for k=1:1:max_k
set(gcf, 'Units', 'Normalized', 'OuterPosition', [0, 0.04, 1, 0.96]);
hold on;
subplot(5,4,[1 17])
pcolor([T_Prof1(:,k),T_Prof1(:,k)]);
    set(gca, 'YDir','reverse');
    set(gca,'xtick',[]);
    h=colorbar('location','westoutside');
    caxis([0 60])
    ylabel(h, ['Local Temperature [',char(176),'C]'],'FontSize',14)
    title({'Kalman Filtered','Temp. Profile'},'FontSize',14 )
    text(1.4,1.85,'X--------------T_1','VerticalAlignment','middle','HorizontalAlignment','left','FontSize',14)
    text(1.4,4.15,'X--------------T_2','VerticalAlignment','middle','HorizontalAlignment','left','FontSize',14)
    text(1.4,6.45,'X--------------T_3','VerticalAlignment','middle','HorizontalAlignment','left','FontSize',14)
    text(1.4,8.75,'X--------------T_4','VerticalAlignment','middle','HorizontalAlignment','left','FontSize',14)
    text(1.4,11.05,'X--------------T_5','VerticalAlignment','middle','HorizontalAlignment','left','FontSize',14)

    subplot(5,4,[2 4])
hold off;
plot([1:1:k]*dt/3600,T_Error_FD(1,1:k),'-','linewidth',1.5);
xlim([0 max_k*dt/3600]);
```

```
ylim([-10 10]);
hold on;
plot([1:1:k]*dt/3600,T_Error_UKF(1,1:k),'--','linewidth',1.5);
plot([k*dt/3600 k*dt/3600],[-10 10],'-k', 'linewidth',1.5)
title('Local Temperature Error vs. Time','FontSize',14)
legend('Unfiltered Local Temp. Error','UKF Filtered Local Temp. Error')
subplot(5,4,[6 8])
hold off;
plot([1:1:k]*dt/3600,T_Error_FD(2,1:k),'-', 'linewidth',1.5);
xlim([0 max_k*dt/3600]);
ylim([-10 10]);
hold on;
plot([1:1:k]*dt/3600,T_Error_UKF(2,1:k),'--', 'linewidth',1.5);
plot([k*dt/3600 k*dt/3600],[-10 10],'-k', 'linewidth',1.5)
subplot(5,4,[10 12])
hold off;
plot([1:1:k]*dt/3600,T_Error_FD(3,1:k),'-', 'linewidth',1.5);
xlim([0 max_k*dt/3600]);
ylim([-10 10]);
hold on;
plot([1:1:k]*dt/3600,T_Error_UKF(3,1:k),'--', 'linewidth',1.5);
ylabel(['\DeltaT_{error} [',char(176),'C]'],'FontSize',14)
plot([k*dt/3600 k*dt/3600],[-10 10],'-k', 'linewidth',1.5)
subplot(5,4,[14 16])
hold off;
plot([1:1:k]*dt/3600,T_Error_FD(4,1:k),'-', 'linewidth',1.5);
xlim([0 max_k*dt/3600]);
ylim([-10 10]);
hold on;
plot([1:1:k]*dt/3600,T_Error_UKF(4,1:k),'--', 'linewidth',1.5);
plot([k*dt/3600 k*dt/3600],[-10 10],'-k', 'linewidth',1.5)
subplot(5,4,[18 20])
hold off;
plot([1:1:k]*dt/3600,T_Error_FD(5,1:k),'-', 'linewidth',1.5);
xlim([0 max_k*dt/3600]);
ylim([-10 10]);
```