On-Chip Thermal Gradients Created by Radiative Cooling of Silicon Nitride Nanomechanical Resonators

ALEXANDRE BOUCHARD

Thesis submitted to the University of Ottawa
in partial fulfillment of the requirements for the degree of

MASTER OF APPLIED SCIENCE IN MECHANICAL ENGINEERING

Department of Mechanical Engineering
Faculty of Engineering
University of Ottawa

PREFACE

A significant portion of this thesis is directly used for the writing of a journal article (by the thesis author) on the same subject. The article is currently in the submission process. See preprint at arXiv:2209.15004 [1].

ABSTRACT

Small scale renewable energy harvesting is an attractive solution to the growing need for power in remote technological applications. For this purpose, localized thermal gradients on-chip—created via radiative cooling—could be exploited to produce microscale renewable heat engines running on environmental heat. This could allow self-powering in small scale portable applications, thus reducing the need for non-renewable sources of electricity and hazardous batteries. In this work, we demonstrate the creation of a local thermal gradient on-chip by radiative cooling of a 90 nm thick freestanding silicon nitride nanomechanical resonator integrated on a silicon substrate at ambient temperature. The reduction in temperature of the thin film is inferred by tracking its mechanical resonance frequency, under high vacuum, using an optical fiber interferometer. Experiments were conducted on 15 different days during fall and summer months, resulting in successful radiative cooling in each case. Maximum temperature drops of 9.3 K and 7.1 K are demonstrated during the day and night, respectively, in close correspondence with our heat transfer model. Future improvements to the experimental setup could enhance the temperature reduction to 48 K for the same membrane, while emissivity engineering potentially yields a maximum theoretical cooling of 67 K with an ideal emitter. This thesis first elaborates a literature review on the field of radiative cooling, along with a theoretical review of relevant thermal radiation concepts. Then, a heat transfer model of the radiative cooling experiment is detailed, followed by the experimental method, apparatus, and procedures. Finally, the experimental and theoretical results are presented, along with future work and concluding remarks.

ACKNOWLEDGEMENTS

This thesis would not have been possible without the help of many individuals, from which I have received a wealth of support and guidance.

First, I would like to extend my deepest gratitude to my supervisor, Professor Raphael St-Gelais, for giving me the opportunity to work on this project and for his continued support during the entire process. Without his generosity, guidance, and teachings, the completion of this thesis would not have been possible.

I am also thankful for my colleagues in the UOMEMS lab at the University of Ottawa. Special thanks to Michel Stephan and Timothy Hodges for designing the membrane mount and helping to build the vacuum chamber. Many thanks to Albert Mu, Chang Zhang, Timothy Hodges, Michel Stephan, and Mathieu Giroux for fabricating the silicon nitride membranes used in this work, and for exchanging interesting discussions which enhanced my understanding of the research topic.

Additionally, I would like to acknowledge Dr. Triantafillos Koukoulas, Dr. Lixue Wu, and Dr. Richard Green from the Metrology Research Centre of the National Research Council of Canada who provided a portion of the equipment used in this work as well as invaluable intellectual input.

Finally, I would like to acknowledge funding by the Natural Sciences and Engineering Research Council of Canada Discovery Program [grant number: RGPIN-2018-04412] and the National Research Council of Canada Ideation Fund: New Beginnings Initiative [grant number: INB-000574-1].

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

List of Abbreviations

CAD	Computer Aided Design
CPV	Concentrating Photovoltaic
EEH	Emissive Energy Harvester
FFT	Fast Fourier Transform
FNR	Town of Fournier, Ontario, Canada
MODTRAN	Moderate Resolution Atmospheric Transmission
RTD	Resistance Temperature Detector
Si	Silicon
SiN	Silicon Nitride
ZnSe	Zinc Selenide
VKH	Town of Vankleek Hill, Ontario, Canada

List of Latin Letters

a_{CTE}	Thermal expansion coefficient	K^{-1}
B	The normalized total tangential electric field at the front interface of an assembly of layers	-
C	The normalized total tangential magnetic field at the front interface of an assembly of layers	-
c_0	Speed of light in vacuum	$m \cdot s^{-1}$
c_v	Constant volume specific heat	$J \cdot kg^{-1} \cdot K^{-1}$
D_m	Molecular diameter	m

d	Thickness of membrane or distance between fiber tip and membrane	m
E	Young's Modulus	Pa
e	Temperature shift approximation error	%
f	Resonance frequency	Hz
Δf	Resonance frequency shift	Hz
h	Planck constant or heat transfer coefficient	$J \cdot s$ or $W \cdot m^{-2} \cdot K^{-1}$
I_{BB}	Blackbody spectral radiative intensity	$W \cdot m^{-2} \cdot m^{-1} \cdot sr^{-1}$
I_N	Nth-order modified Bessel function of the first kind	-
K_n	Knudsen number	-
k	Thermal conductivity	$W \cdot m^{-1} \cdot K^{-1}$
k_B	Boltzmann constant	$J \cdot K^{-1}$
L	Side length of a square membrane	m
l	Characteristic length	m
(n, j)	Number of antinodes in the vibrational mode shape of the membrane	-
N	Refractive Index	-
P	Pressure	hPa
q	Emissive power	$W \cdot m^{-2}$
R_{air}	Gas constant of air	$J \cdot kg^{-1} \cdot K^{-1}$
S	Seebeck coefficient	$V \cdot K^{-1}$
T	Temperature	K
ΔT	Temperature shift	K
u	Uncertainty	-

V	Voltage	V
x_{cond}	Fraction of heat transfer occurring via conduction	-
ZT	Thermoelectric material figure of merit	-
z	Number of layers in thin film assembly	-

List of Greek Letters

A_T	Thermal accommodation coefficient	-
α	Absorptivity	-
β	NTC thermistor temperature coefficient	K
βr_{eff}	Membrane geometry factor	-
γ	Specific heat ratio	-
δ	Phase thickness of a thin film layer, or declination angle	- or radians
ε	Emissivity	-
η	Admittance or efficiency	-
θ	Zenith angle of the photon leaving/hitting the surface	radians
κ	Electrical conductivity	$\mathrm{S \cdot m^{-1}}$
Λ	Mean free path	m
λ	Wavelength	m
ν	Poisson ratio	-
ρ	Reflectivity or density	- or $\mathrm{kg \cdot m^{-3}}$
σ	Stefan-Boltzmann constant	$\mathrm{W \cdot m^{-2} \cdot K^{-4}}$
ς_0	Membrane tensile stress	Pa

τ	Transmissivity	-
Φ	Latitude	degrees
ϕ	Azimuth angle of the photon leaving/hitting the surface	radians
ω	Hour angle	degrees
Ω	Resistance	Ohm

List of Subscripts

0	Initial or vacuum
$AM1.5$	AM1.5 solar spectrum
BB	Blackbody
$back$	Backside of membrane
atm	Atmosphere
ch	Chamber
$cond$	Conductive
EI-1034	LabJack EI-1034 temperature probe
FM	Free molecular flow
i	Incoming radiation (i.e., irradiation)
inc	Incident medium
m	Membrane, or mean
out	Output medium or substrate
r	Layer r
th	Thermistor

TE	S-polarization (transverse-electric)
TEG	Thermoelectric generator
top	Topside of membrane
TM	P-polarization (transverse-magnetic)
v	Viewport
θ	Directional
λ	Spectral

1 INTRODUCTION

Converting ambient heat to electricity is an attractive solution to the growing need for renewable power in remote locations. More specifically, microscale energy harvesting on-chip could allow self-powering in small scale portable applications, which would reduce the need for non-renewable energy sources and hazardous batteries. However, from a thermodynamic standpoint, harvesting ambient heat is only possible with a heat sink at a colder temperature. Outer space ($T \approx 3$ K) is a sustainable prospect for such a heat sink. Remarkably, objects on Earth can access this heat sink via a passive phenomenon known as radiative cooling.

Radiative cooling allows for the temperature of an object to diminish below ambient by coupling it with the coldness of outer space via thermal radiation. This phenomenon is possible due to the substantial temperature difference between the surface of the Earth ($T \approx 300$ K) and outer space ($T \approx 3$ K), as well as coupling between the two enabled by atmospheric transparency ($\tau_{\lambda,atm}$). The atmosphere is especially transparent to radiation in the 8 μm – 13 μm wavelength range which is referred to as the atmospheric transparency window (see Figure 1.1a). This means a significant portion of radiation in the 8 μm – 13 μm wavelength range is transmitted through the atmosphere towards the enormous heat sink that is outer space. Therefore, significant cooling below ambient temperature is possible for objects having a clear view of the sky and possessing the following material properties:

i. High emissivity in the 8 μm – 13 μm wavelength range to maximize radiation to outer space through the atmospheric transparency window,
ii. Weak absorptivity outside of the 8 μm – 13 μm wavelength range to minimize parasitic radiative heating (e.g., from the atmosphere, from the sun) [2], and

iii. Thermal coupling with the environment dominated by radiation to minimize conductive heating.

Radiative cooling has been investigated in applications such as heat extraction (e.g., in buildings [2]–[7] and photovoltaics [8]–[11]) and macroscale energy harvesting [12]–[15], but its use for microscale electricity generation integrated on-chip has not been reported. In the late 20th century, the radiative cooling of bulk materials with intrinsic radiative cooling properties was studied for potential energy savings applications [16]–[22]. Such bulk materials were mostly limited to nighttime radiative cooling, or in some cases daytime radiative cooling with diffuse solar irradiation only. More recently, advancements in material science and photonics have allowed more tuning of the spectral radiative properties of an object. This has made daytime radiative cooling—with direct sunlight exposure—possible by suppressing absorption in the solar spectrum [8], [23]–[36]. Raman et al. first demonstrated radiative cooling under direct sunlight with a 1D multilayer structure in 2014, achieving a temperature drop of 4.9 K below ambient [24]. In 2016, a record temperature drop of 42 K below ambient in peak daylight was demonstrated [26]. Energy harvesting applications harnessing ambient heat via radiative cooling have been demonstrated, but only by cooling of a macroscopic object connected to an external, bulk, thermoelectric generator [13], [14]. Integrating such a heat engine on a chip would require localized cooling—i.e., a thermal gradient on-chip—with integrated thermoelectric circuits. This was not possible in the previously investigated devices where the entire substrates were uniformly cooled.

The objective of this thesis is to demonstrate the first fundamental ingredient of a microscale integrated heat engine: creation of a local thermal gradient on-chip. This is demonstrated by radiative cooling of a 90 nm thick, freestanding silicon nitride (SiN) membrane integrated on a silicon (Si) chip that remains at ambient temperature (see Figure 1.1b). Recent work has shown that the thermal coupling of a SiN membrane with its environment can be dominated by thermal radiation [37]–[39], making it a material platform of choice for thermal radiation sensing experiments [40], [41]. Moreover, the radiative hemispherical emissivity spectrum of freestanding SiN ($\varepsilon_{\lambda,m}$) is naturally well aligned with the atmospheric transparency window while having weak absorptivity elsewhere (see Figure 1.1a), which means freestanding SiN possesses the criteria for significant radiative cooling. SiN membranes are also a robust platform over which it is possible to deposit metal and semiconductors to create planar thermoelectric circuits [42], which could enable micro heat engines on-chip.

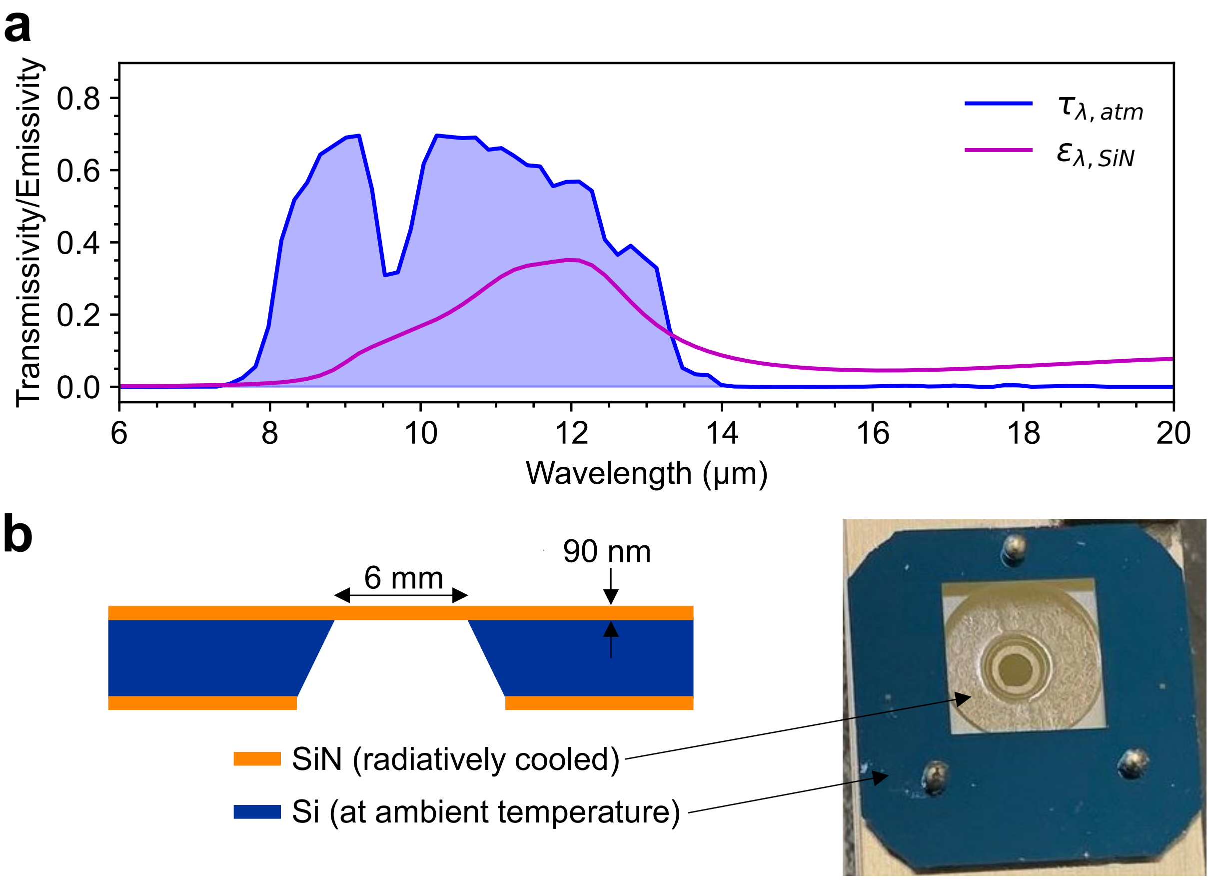


Figure 1.1. a) $\tau_{\lambda,atm}$: atmospheric transmissivity at normal incidence for mid-latitude summer climate conditions computed by the Moderate Resolution Atmospheric Transmission (MODTRAN) model [43], [44]. $\varepsilon_{\lambda,m}$: hemispherical emissivity of a 90 nm thick SiN membrane. b) SiN chip used in this work. The membrane is semitransparent, so the membrane mount and optical fiber tip (centered black dot) are visible. Note that the bottom layer of SiN is a fabrication byproduct [45].

This thesis first elaborates a literature review on the field of radiative cooling, along with a theoretical review of relevant thermal radiation concepts, in section 2. A heat transfer model of the radiative cooling experiment is elaborated in section 3. The experimental method, apparatus, and procedures are described in section 4. Finally, the results are discussed in section 5, and the future work and concluding remarks are outlined in section 6.

2 LITERATURE REVIEW

Although the study of radiative cooling began in the 20th century, humans have been harnessing the benefits of radiative cooling since ancient times. Over a millennium ago, the creation of ice in the deserts of Iran was achieved via radiative cooling [3]. During winter nights, at temperatures above the freezing point, shallow water-filled ponds would gain heat via ground conduction and environmental convection but would lose heat to the sky via radiation (i.e., radiative cooling). Shielding along the sides of the ponds would reduce convective heating, which allowed sufficient heat loss for the creation of ice. Much later—from the 1970s onwards—this passive cooling phenomenon was studied for potential energy savings applications. Section 2.1 elaborates on important thermal radiation concepts relevant to radiative cooling, and sections 2.2 and 2.3 reviews the radiative cooling materials and applications in the literature, respectively.

2.1 THERMAL RADIATION OVERVIEW

Important thermal radiation concepts—relevant to radiative cooling—are elaborated in this section. Specifically, the concepts of blackbody radiation and emissive power are elaborated in section 2.1.1, and the notion of radiative emission from real surfaces is detailed in section 2.1.2.

2.1.1 Blackbody Concept & Emissive Power

Thermal radiation is emitted from any object with a temperature above absolute zero in the form of electromagnetic waves (i.e., photons) [46]. The maximum possible radiative power emitted by an object is described by the emissive power of an ideal radiator—i.e., a blackbody. The spectral radiative intensity (in $W{\cdot}m^{-2}{\cdot}m^{-1}{\cdot}sr$) of a blackbody is given by [46]:

$$I_{BB}(\lambda,T) = \frac{2hc_0^2}{\lambda^5\left(e^{\frac{hc_0}{\lambda k_B T}} - 1\right)}, \tag{2.1}$$

where λ is the wavelength of radiation, $h = 6.626 \times 10^{-34}$ J·s is the Planck constant, $k_B = 1.381 \times 10^{-23}$ J·K^{-1} is the Boltzmann constant, $c_0 = 2.998 \times 10^{8}$ m·s^{-1} is the speed of light in vacuum, and T is the temperature of the blackbody (in K). The blackbody is used as a reference to quantify the radiative properties of real surfaces. For instance, the efficiency at which a surface emits radiative energy is referred to as its emissivity (ε). Since a blackbody is an ideal radiator, its emissivity always unity, and the emissivity of a real surface is always $0 < \varepsilon < 1$ (i.e., relative to a blackbody).

The rate at which radiation leaves a real surface (in W·m^{-2}) can be determined by weighing the emissivity of the real surface with the emissive power of a blackbody at the same temperature (T). Therefore, the total hemispherical emissive power (i.e., the rate at which radiation is leaving a real surface integrated over all wavelengths and all directions) is given by [46]:

$$q = \int_0^{\infty}\int_0^{2\pi}\int_0^{\frac{\pi}{2}} \varepsilon_{\lambda,\theta}(\lambda,\phi,\theta) I_{BB}(\lambda,T)\cos\theta\sin\theta d\theta d\phi d\lambda, \tag{2.2}$$

where λ, θ and ϕ are the wavelength, azimuth angle and zenith angle of the photon leaving the surface, respectively, and $\varepsilon_{\lambda,\theta}$ is the spectral directional emissivity of the real surface. Reciprocally, the rate at which incoming radiation (i.e., irradiation) from a large surrounding with emissivity $\varepsilon_{\lambda,\theta,i}$ and temperature T_i is incident upon a real surface is given by [46]:

$$q_i = \int_0^{\infty}\int_0^{2\pi}\int_0^{\frac{\pi}{2}} \varepsilon_{\lambda,\theta,i}(\lambda,\phi,\theta)\alpha_{\lambda,\theta}(\lambda,\phi,\theta) I_{BB}(\lambda,T_i)\cos\theta\sin\theta d\theta d\phi d\lambda. \tag{2.3}$$

where $\alpha_{\lambda,\theta}$ is the spectral directional absorptivity of the real surface. To help visualize the radiation angles (ϕ, θ), a depiction of the radiation leaving a real surface, with ϕ and θ identified, is shown in Figure 2.1.

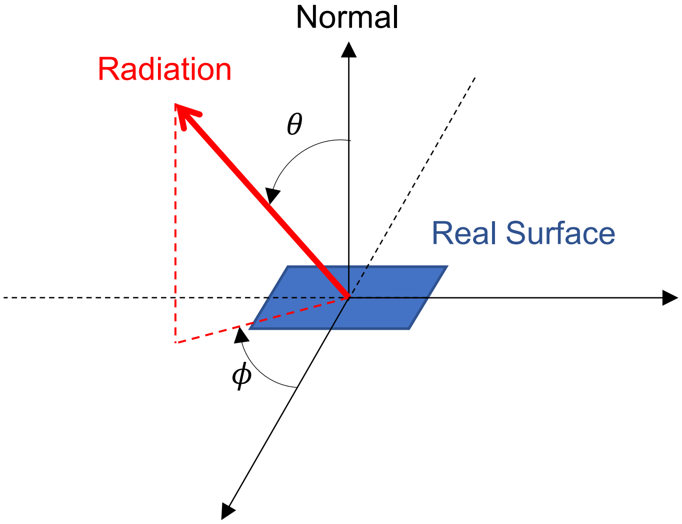


Figure 2.1. Depiction of the directionality (θ, ϕ) of radiation. Note that this depiction is also valid for irradiation on the surface.

2.1.2 Emission by Real Surfaces

Kirchhoff's law states that the spectral directional emissivity ($\varepsilon_{\lambda,\theta}$) and absorptivity ($\alpha_{\lambda,\theta}$) of a real surface are always equal (i.e., $\varepsilon_{\lambda,\theta} = \alpha_{\lambda,\theta}$) [46]. Since irradiation on a semitransparent medium (e.g., a SiN membrane) is either reflected, transmitted, or absorbed, its spectral directional absorptivity ($\alpha_{\lambda,\theta}$)—and thus emissivity ($\varepsilon_{\lambda,\theta}$)—must respect radiation balance [46]:

$$\alpha_{\lambda,\theta} = \varepsilon_{\lambda,\theta} = 1 - \tau_{\lambda,\theta} - \rho_{\lambda,\theta}, \qquad (2.4)$$

where $\rho_{\lambda,\theta}$, and $\tau_{\lambda,\theta}$ are the spectral directional reflectivity and transmissivity of the real surface, respectively. The hemispherical spectral emissivity (i.e., integrated over all angles) of the real surface can be calculated as [46]:

$$\varepsilon_\lambda(\lambda) = 2\int_0^{\frac{\pi}{2}} \varepsilon_{\lambda,\theta}(\lambda,\theta)\, cos\,\theta\, sin\,\theta d\theta. \tag{2.5}$$

In turn, the total hemispherical emissivity (i.e., integrated over all angles and all wavelengths) of the real surface at temperature T can be calculated as [46]:

$$\varepsilon = \frac{\int_0^\infty \varepsilon_\lambda(\lambda) q_{\lambda,BB}(\lambda,T)}{\int_0^\infty q_{\lambda,BB}(\lambda,T)}, \tag{2.6}$$

where,

$$q_{\lambda,BB}(\lambda,T) = \pi I_{BB}(\lambda,T). \tag{2.7}$$

2.2 RADIATIVE COOLING MATERIALS

The radiative cooling effectiveness of an object is dependant on its spectral radiative properties. As described in section 1, high emissivity in the 8 µm – 13 µm wavelength range and weak absorptivity elsewhere is necessary for effective radiative cooling. Some materials naturally possess mediocre spectral properties for radiative cooling; however, the advent of material science has allowed the fine-tuning of the spectral properties of objects (e.g., photonic science), which in turn allowed the pursuit of the ideal radiative cooler. This section elaborates on the various materials used for radiative cooling in the literature, including natural and bulk materials, and photonic devices.

2.2.1 Natural & Bulk Materials

Several examples of radiative cooling processes occur in nature. Plant canopies experience radiative cooling at night, which allows their temperature to diminish lower than ambient [47]. This phenomenon results in the accumulation of dew water on the plant canopies during summer nights even though the ambient temperature is above the dewpoint, and in frost during fall mornings even though the ambient temperature is above the freezing point. In addition, some types of ants have evolved to harness radiative cooling for thermoregulation purposes [48]. For instance, the Saharan silver ant body surface possesses high reflectivity in the solar and near-infrared spectrum, and high emissivity in the atmospheric transparency window. This allows the ant to maintain a cool body temperature as low as 22 °C below the ambient desert surface, under full daylight conditions.

In 1975, Catalanotti et al. published a pioneering study on the radiative cooling of bulk materials [16]. This study included a thought experiment to clearly understand the concept of radiative cooling. First, imagine a blackbody radiator floating in outer space. This radiator emits heat to deep space and receives negligeable heat back from space. Due to the vastness of outer space, the blackbody cools down and eventually reaches the temperature of outer space ($T \approx 3$ K). Hence, this blackbody radiator is experiencing radiative cooling. If the same radiator is instead placed on the surface of the Earth, this cooling phenomenon is immensely affected by the atmosphere and the sun. However, significant cooling is still possible via the atmospheric transparency window (i.e., 8 µm – 13 µm wavelength range). To enhance radiative cooling on Earth, the concept of an ideal selective radiator—possessing a blackbody emissivity in the 8 µm to 13 µm range and null emissivity elsewhere—was suggested (see Figure 2.2a). In an effort to create such an ideal radiator, a thin film of TEDLAR (a polyvinyl-fluoride plastic) was deposited on a sheet of aluminium, resulting in a spectral absorptivity shown in Figure 2.2b. With an experimental setup that reduces convective and conductive heat gains (i.e., an insulated box with a transparent cover), the radiative cooling of this bulk selective emitter was demonstrated, resulting in a temperature drop of 15 °C below ambient under diffused solar radiation conditions.

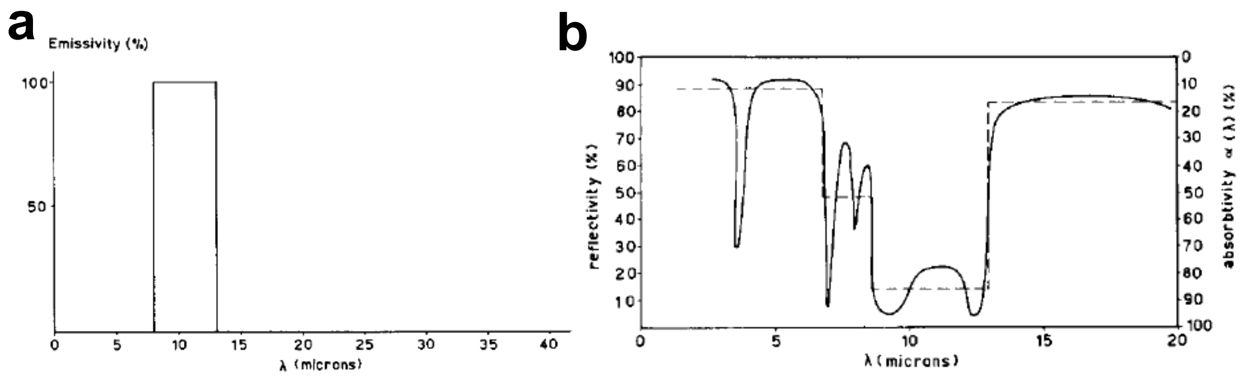


Figure 2.2. a) Emissivity of a conceptual ideal selective radiator, and b) Emissivity of TEDLAR on a sheet of aluminium, by Catalanotti et al. [16].

In the early 1980s, the radiative cooling of silicon-based bulk materials was demonstrated in work by Granqvist et al. [17], [18]. The spectral emissivity of silicon dioxide was shown to possess intrinsic radiative cooling properties (see Figure 2.3a). The radiative cooling of a silicon dioxide film on a sheet of aluminum was demonstrated in a thermally insulated box, resulting in a temperature drop of 14 °C at night [18]. In addition, the radiative cooling potential of silicon nitride was predicted to be significantly higher than that silicon dioxide because of its stronger emissivity in the 8 μm – 13 μm wavelength range (see Figure 2.3b). Thus, the cooling potential of silicon nitride (i.e., the material of choice in this study) was established.

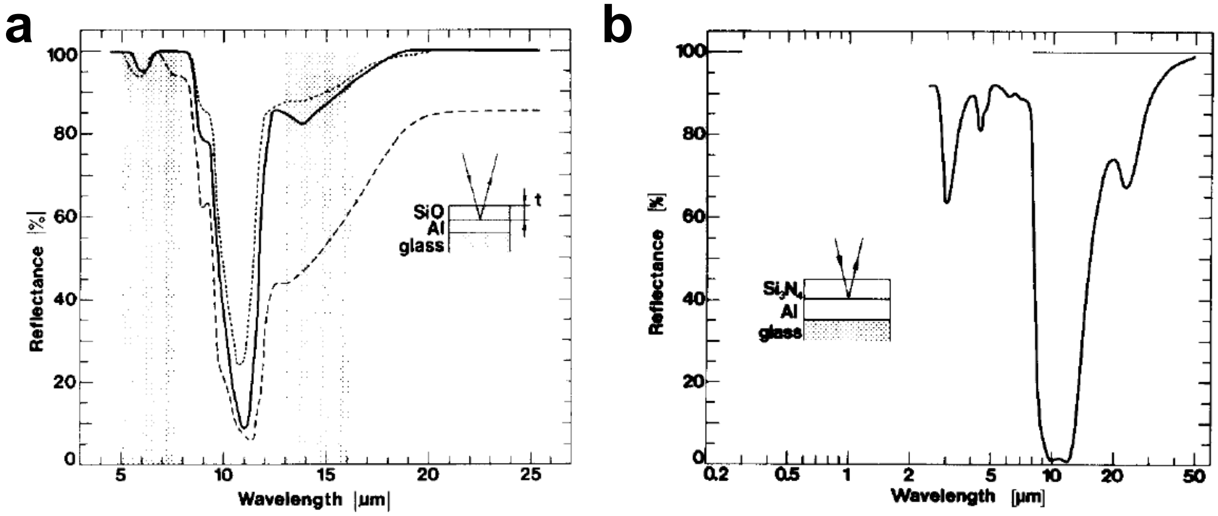


Figure 2.3. a) Measured spectral reflectivity for a 100 μm thick SiO_2 film on an aluminium film, and b) measured spectral reflectivity for Si_3N_4 on an aluminium film, by Granqvist et al. [18].

Other bulk materials (e.g., silicon oxynitride [19], paint [22], plastics [16], [20], [21]) have been studied for radiative cooling purposes. However, the cooling of such bulk materials was mostly limited to nighttime radiative cooling, or in some cases daytime cooling with diffuse solar irradiation conditions. Daytime radiative cooling under direct sunlight was only recently achieved thanks to the advancement of material science, especially in the field of photonics. The impact of photonic devices on the field of radiative cooling is elaborated in section 2.2.2.

2.2.2 *Photonic Devices*

Photonic devices take advantage of the spectral radiative properties of several materials in order to create objects with new radiative properties. The manipulation of these materials, in the form of multilayering or surface patterns, allows the fine tuning of the radiative properties of the photonic device. This is especially important for daytime radiative cooling because it allows the creation of a device with high emissivity in the 8 µm – 13 µm range and weak absorptivity elsewhere, especially in the solar spectrum (i.e., roughly 0.28 µm – 2.5 µm; see Figure 3.4).

In 2013, Rephaeli et al. proposed a theoretical photonic structure that behaves like a mirror in the solar spectrum and possesses strong emissivity in the atmospheric transparency window (see Figure 2.4a) [23]. Although this device shows impressive cooling potential in theory, it is very complex to fabricate. In 2014, Raman et al. designed a simpler, more practical, photonic structure for feasible large-scale fabrication (see Figure 2.4b) [24]. With this device, the first daytime radiative cooling experiment under direct sunlight was demonstrated with a temperature drop of 4.9 K below ambient. This trailblazing work inspired an emergence of studies for daytime radiative cooling with photonic structures, often in the from of multilayers [25]–[29], micro surface patterns [8], [30], [31], or metamaterials [32]–[36]. A notable experiment was conducted by Chen et al. in

2016 with the multilayer photonic structure illustrated in Figure 2.4c [26]. An average temperature of 37 K below ambient was maintained for a 24-hour experiment at high vacuum (P ≈ 10^{-5} hPa), with a record 42 K below ambient in peak daylight.

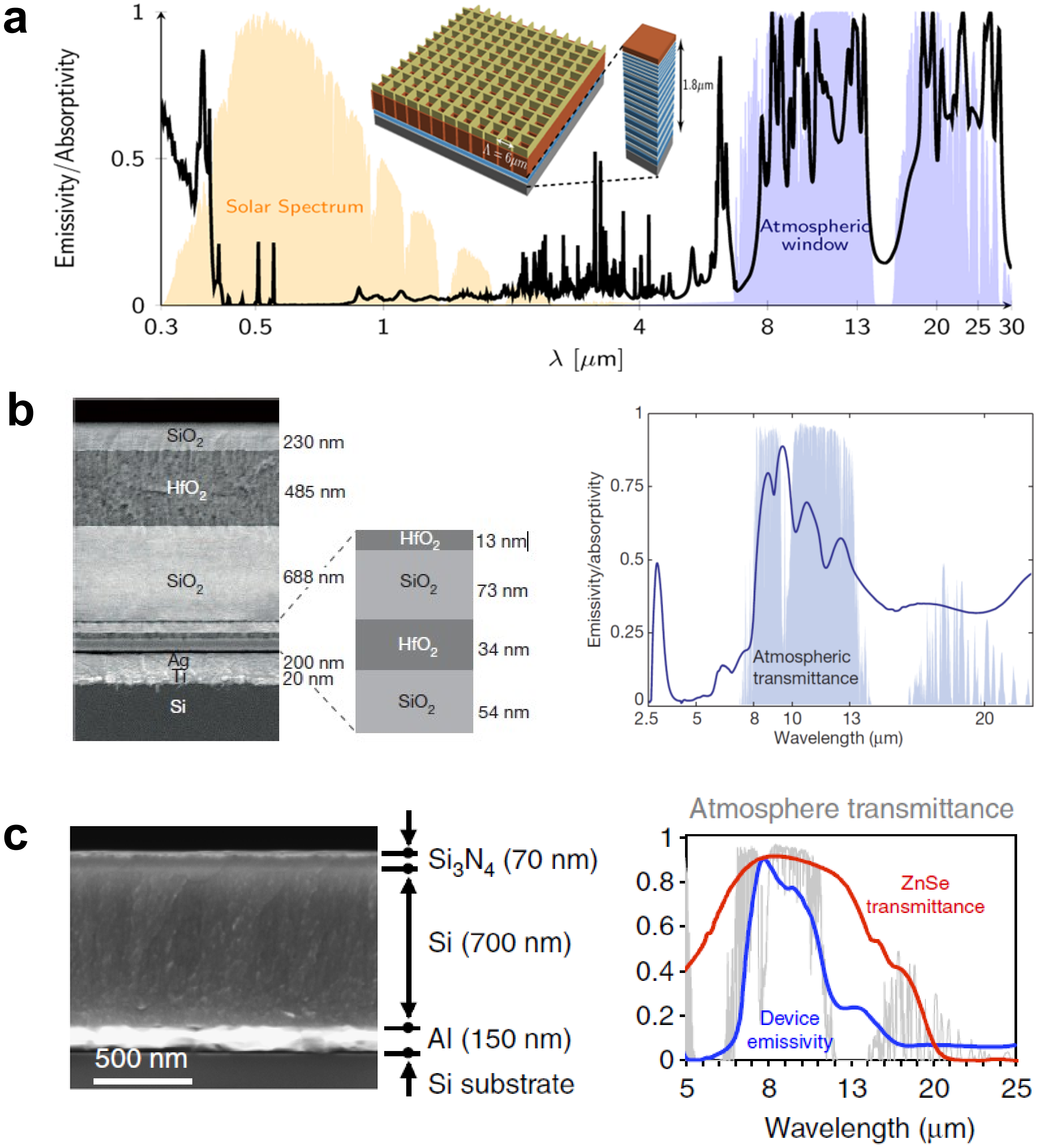


Figure 2.4. a) Theoretical photonic structure for daytime radiative cooling by Rephaeli et al. [23]. b) Photonic structure used in the first direct sunlight radiative cooling experiment by Raman et al. [24]. c) Multilayer photonic structure holding the current record for maximal temperature drop via radiative cooling, by Chen et al. [26].

The previous studies demonstrate cooling of the entire substrate on which the device is deposited, and thus no local thermal gradient between the cooling structure and the chip is created. This thesis investigates localized cooling of freestanding membranes, which creates a thermal gradient on-chip between the membrane and the supporting silicon frame.

2.3 RADIATIVE COOLING APPLICATIONS

Radiative cooling can be used as a renewable resource for a variety of applications. Notably, it can be used as a passive heat extraction process (e.g., for buildings [2]–[7], for photovoltaics [8]–[11]) or it can be used for energy harvesting applications by taking advantage of the thermal gradient created between the radiative cooler and the ambient environment [12]–[15]. This section elaborates on the investigated applications of radiative cooling.

2.3.1 Heat Extraction Applications

Radiative cooling can be an efficient and renewable method to extract excess heat from systems (e.g., buildings, powerplants). The thermal management of buildings accounts for a significant portion of global energy consumption and it will keep increasing due to global warming and the growing need for heat-dissipating electronics. To reduce the energy consumption of buildings, radiative cooling can be used in conjunction with conventional building cooling systems while requiring no additional energy input [2]–[7]. An interesting example is presented by Wang et al. where a photonic radiative cooling panel is integrated with a conventional radiant cooling system for a medium-sized office building (see Figure 2.5) [6]. Theoretical simulations show that the photonic radiative cooling panel—which can notably operate during the day, under direct sunlight—can save between 45% and 68% of the cooling electricity demand compared to a conventional stand-alone variable-air-volume cooling system.

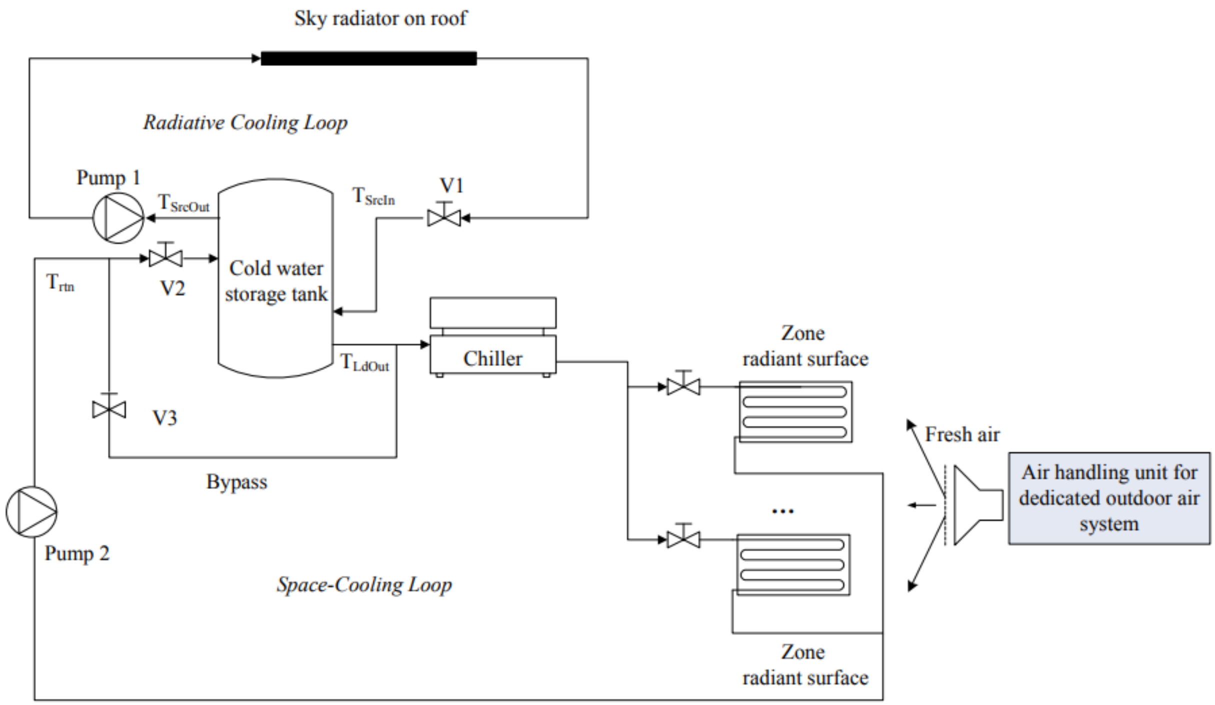


Figure 2.5. Photonic radiative cooling system for office buildings, by Wang et al. [6].

In 2022, Li et al. proposed radiative cooling as a solution for preventing the melting of ice in various settings (e.g., food preservation, ice sports) [49]. Interestingly, the radiative cooler in this case is made from cellulose acetate (CA)—a biodegradable material derived from natural cellulose in plants—which is eco-friendly and widely available. Experiments were conducted in a refrigerator simulating the atmospheric conditions of high-latitude regions (i.e., 0 ℃ to 8 ℃), under direct sunlight exposure (see Figure 2.6a). Results show that ice protected by the biodegradable radiative cooling film maintains a temperature approximately 7 ℃ lower than ambient. This allows the preservation of ice under direct sunlight, compared to bare ice which completely degrades within five days (see Figure 2.6b). This application could potentially reduce the significant energy demand, and thus greenhouse gas emissions, related to outdoor food preservation. Additionally, it could be used to slow the melting of glaciers, which is a critical element to various ecosystems.

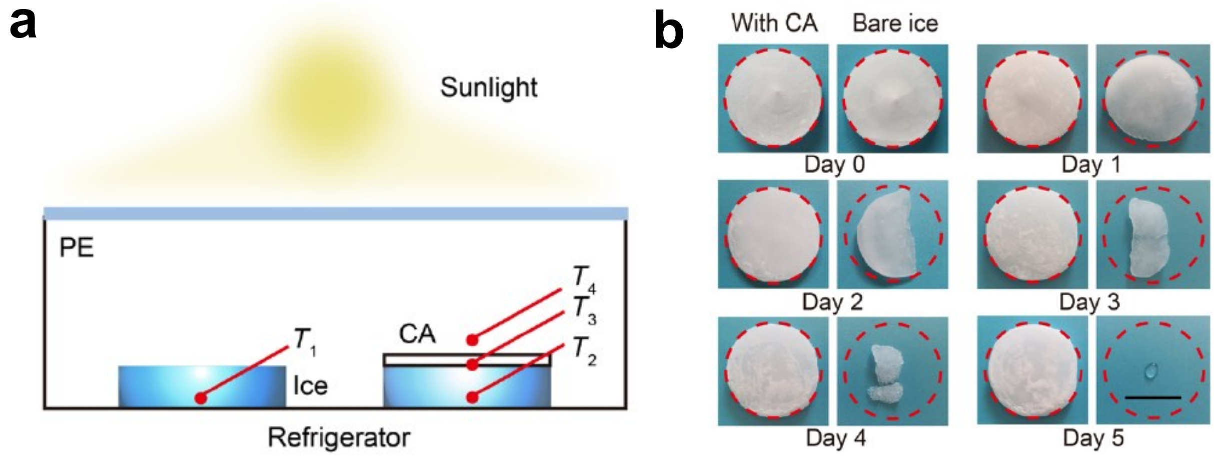


Figure 2.6. Radiative cooling as a solution for preventing the melting of ice [49]. a) Schematic of experiment testing the performance of the cellulose acetate (CA) film used for the protection of ice. b) Comparison of the evolution of ice mass between bare ice and ice with CA protection for a 5-day experiment.

Radiative cooling can be used as a passive method to extract heat from photovoltaics [8]–[11]. The efficiency of a p-n junction solar cell is inversely proportional to the temperature of the cell (i.e., its self-heating reduces the amount of useful energy generated) [50]. Advancements in the field of photonics makes radiative cooling an interesting solution to cool down solar cells under direct sunlight. Zhu et al. [8] conceptualized a theoretical micro photonic structure which could operate 17.6 K below a conventional bare solar cell (at 800 W·m^{-2} solar heating power; see Figure 2.7a). Such a temperature difference results in a 7.9% efficiency increase (e.g., for a solar efficiency of 20%, this corresponds to a 1.6% absolute efficiency boost). In 2020, Wang et al. experimentally demonstrated the cooling of a concentrated photovoltaic (CPV) solar cell via radiative cooling [10]. Compared to a conventional CPV solar cell, the CPV solar cell integrated with a radiative cooler maintained a 5 K to 36 K cooler temperature, corresponding to an 8% to 27% increase of open-circuit voltage. This cooler operating temperature also corresponds to an estimated lifetime extension of 4 to 15 times that of a conventional CPV solar cell. The experimental setup is depicted in Figure 2.7b.

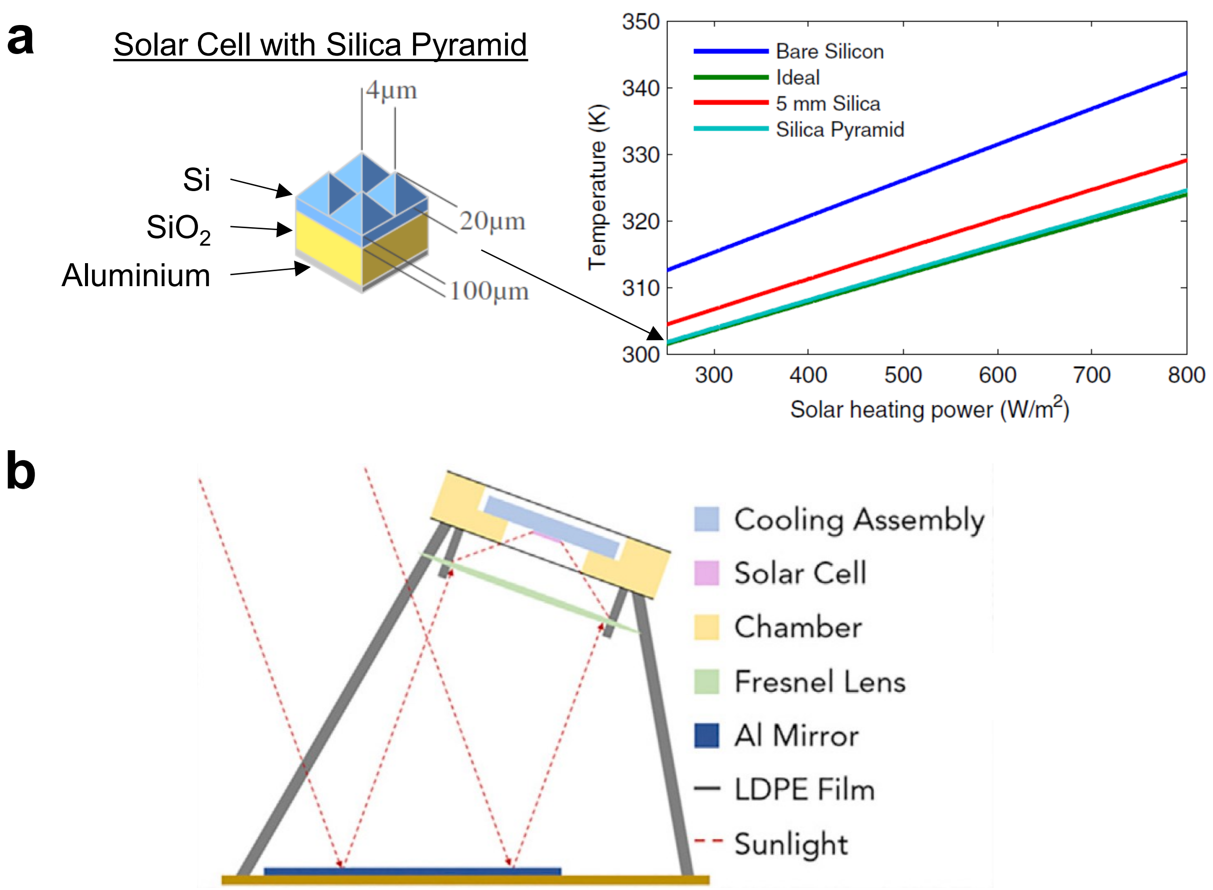


Figure 2.7. a) Theoretical photonic structure for radiative cooling of solar cells by Zhu et al. [8]. The left side shows the design of the solar cell with silica pyramids for radiative cooling. The right side shows a graph of the theoretical temperature of this photonic structure as a function of solar heating power compared to two other radiative cooling designs (ideal, and 5 mm Silica) and bare silicon. b) CPV solar cell integrated with a radiative cooling assembly, by Wang et al. [10].

2.3.2 *Energy Harvesting Applications*

Radiative cooling can be used to exploit the temperature difference between the warm surface of the Earth and the cold outer space ($\Delta T \approx 300$ K) for harnessing renewable energy. In 2014, Byrnes et al. [12] proposed two concepts for emissive energy harvesters (EEH): (i) thermal EEH (i.e., a heat engine thermoelectric generator running between the cold radiative cooler and a warm ambient object), and (ii) optoelectronic EEH (i.e., a rectifying antennae that is radiatively cooled while the rest of its supporting panel remains at ambient temperature). See Figure 2.8 for a depiction of both EEH designs.

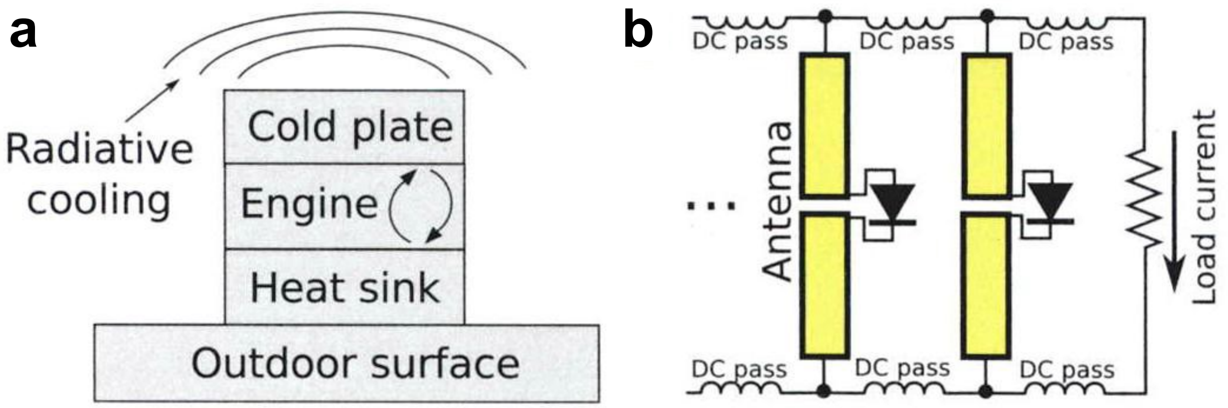


Figure 2.8. a) Thermal EEH, and b) optoelectronic EEH, proposed by Byrnes et al. [12].

Raman et al. experimentally demonstrated a thermal EEH in 2019 using a thermoelectric generator and a blackbody radiative cooler (see Figure 2.9) [13]. The blackbody maintained a temperature of 4 K to 5 K below ambient at night which, due to conductive heating, resulted in a 2 K thermal gradient between the cold side and the hot side of the thermoelectric generator. This resulted in a power generation of 25 mW·m^{-2}, which could directly power a light emitting diode. Similar work—i.e., generating electricity via a thermal EEH with a radiatively cooled macroscopic object—was demonstrated by Ishii et al. [14]. However, microscale electricity generation integrated on-chip via radiative cooling (i.e., the long-term objective of this work) has not been reported.

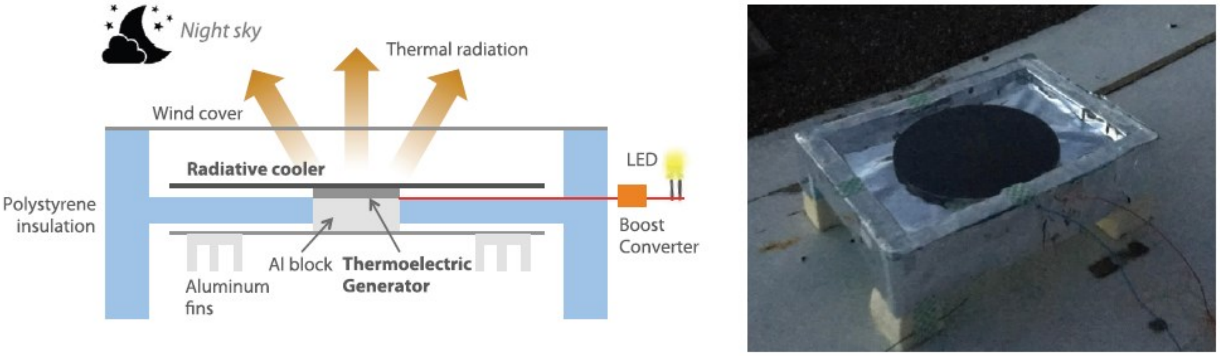


Figure 2.9. Thermal EEH demonstrated by Raman et al. in 2019 [13].

3 HEAT TRANSFER MODEL

A depiction of the radiative cooling experiment conducted in this work—a freestanding SiN membrane mounted inside a vacuum chamber and facing the sky through a zinc selenide (ZnSe) viewport—is presented in Figure 3.1. The main heat transfer rates are depicted in Figure 3.1, where q_m is the rate at which radiation is leaving the membrane, q_{atm} is the rate at which atmospheric irradiation is incident on the membrane, $q_{ch,top}$ and $q_{ch,back}$ are the rates at which chamber irradiation from the topside and backside of the membrane, respectively, are incident on the membrane, q_{cond} is the rate of conduction from the silicon frame to the membrane, and q_{sun} is the rate at which solar radiation is incident on the membrane. Convection is negligeable because the membrane is mounted inside a vacuum chamber which is kept at high vacuum in the 10^{-6} hPa range (see Appendix A for further justification).

The net cooling power (in W·m^{-2}) of the membrane is given by the energy balance of the heat transfer rates as follows:

$$q_{net} = q_m(T_m) - q_{atm}(T_{atm}) - q_{ch,top}(T_{ch}, \theta_v) - q_{ch,back}(T_{ch}) - q_{cond}(T_m, T_{ch}) - q_{sun}(\theta_{sun}), \tag{3.1}$$

where T_m, T_{atm}, and T_{ch} are the respective temperatures of the membrane, atmosphere, and vacuum chamber, θ_v is the viewport angle representing the membrane field of view (see Figure 3.1), and θ_{sun} is the angle at which the sun is facing the membrane (see Figure 3.1). For given values of T_{atm}, T_{ch}, θ_v, and θ_{sun}, the steady state (i.e., q_{net} = 0) energy balance can be solved to determine the membrane temperature (T_m)—and thus temperature drop ($\Delta T_m = T_{ch} - T_m$)—upon radiative cooling. For this purpose, the expressions for all heat transfer rates are detailed in section 3.2. Section 3.1 elaborates on the methods used to determine the membrane emissivity.

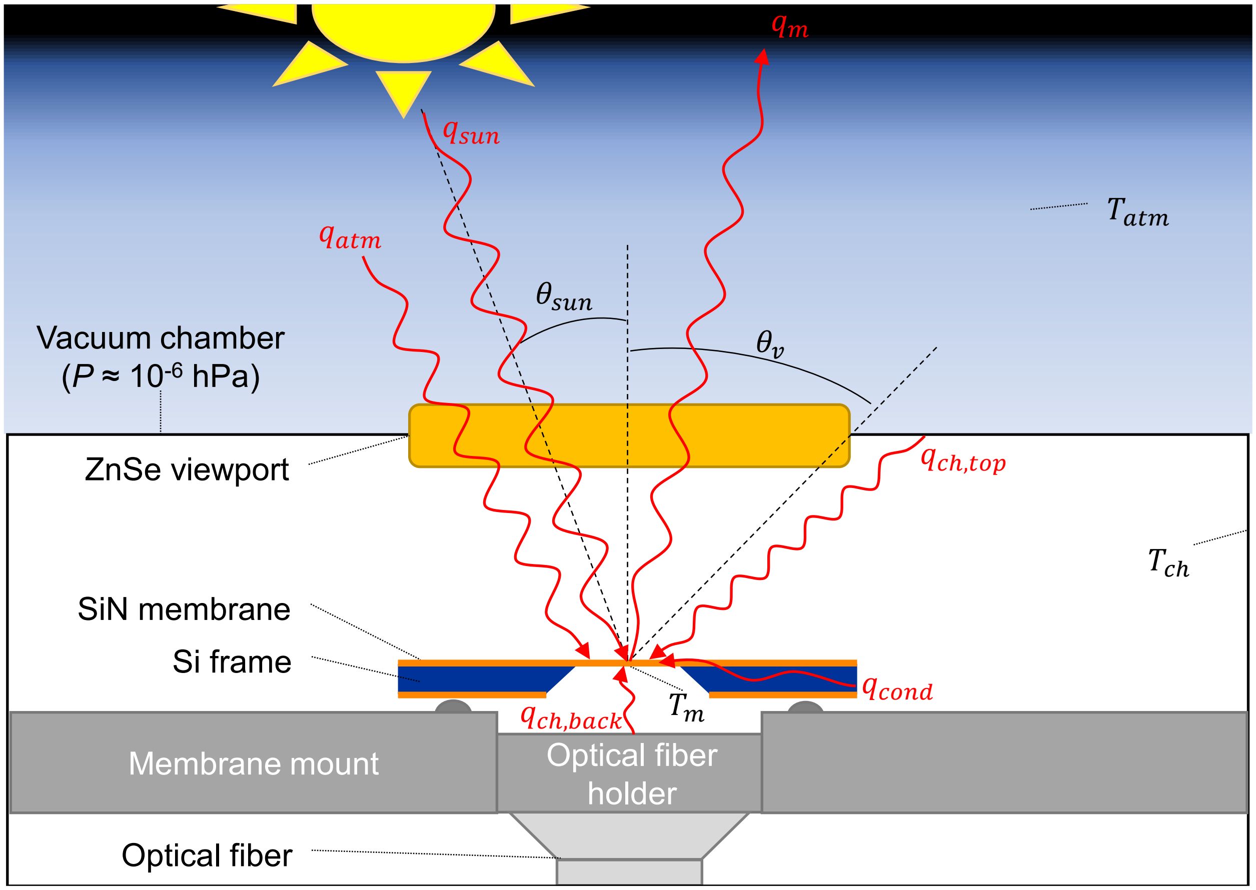


Figure 3.1. Radiative cooling experiment depiction: a freestanding SiN membrane mounted inside a vacuum chamber and facing the sky through a zinc selenide (ZnSe) viewport.

3.1 MEMBRANE EMISSIVITY

3.1.1 Thin Film vs. Multilayer Emissivity Models

The membrane spectral directional emissivity ($\varepsilon_{\lambda,\theta,m}$) is a central quantity for evaluating the net radiative cooling rate in eq. (3.1), and its value can be affected by the membrane surroundings. Specifically, the backside surrounding of the membrane (i.e., the optical fiber holder and membrane mount; see Figure 3.1) is an unpolished, partially reflective, metal surface that can act both as (i) a mirror that enhances coupling of the membrane with the sky and improves cooling or, (ii) a parasitic source of radiated heat. Precisely quantifying these effects is not immediately obvious, so the model accounts for the two extreme cases. In a best-case scenario, the backside is a perfect mirror that increases the effective emissivity of the membrane. In this case, the membrane is treated as a multilayer of 90 nm SiN, 1.4 mm vacuum (i.e., the space between the membrane and the optical fiber holder), and an aluminum mirror substrate. This case is referred to as the multilayer model (see Figure 3.2). In a worst-case scenario, the backside is a large isothermal surrounding emitting blackbody radiation, which means the membrane is treated as a standalone 90 nm SiN thin film. This case is referred to as the thin film model (see Figure 3.2). In both cases, the membrane spectral directional emissivity ($\varepsilon_{\lambda,\theta,m}$) is calculated using eq. (2.4), for which the membrane spectral directional reflectivity ($\rho_{\lambda,\theta,m}$) and transmissivity ($\tau_{\lambda,\theta,m}$) are calculated using the characteristic matrix formalism (see section 3.1.2) [51]. The resulting spectral hemispherical emissivity of both models is shown in Figure 3.2.

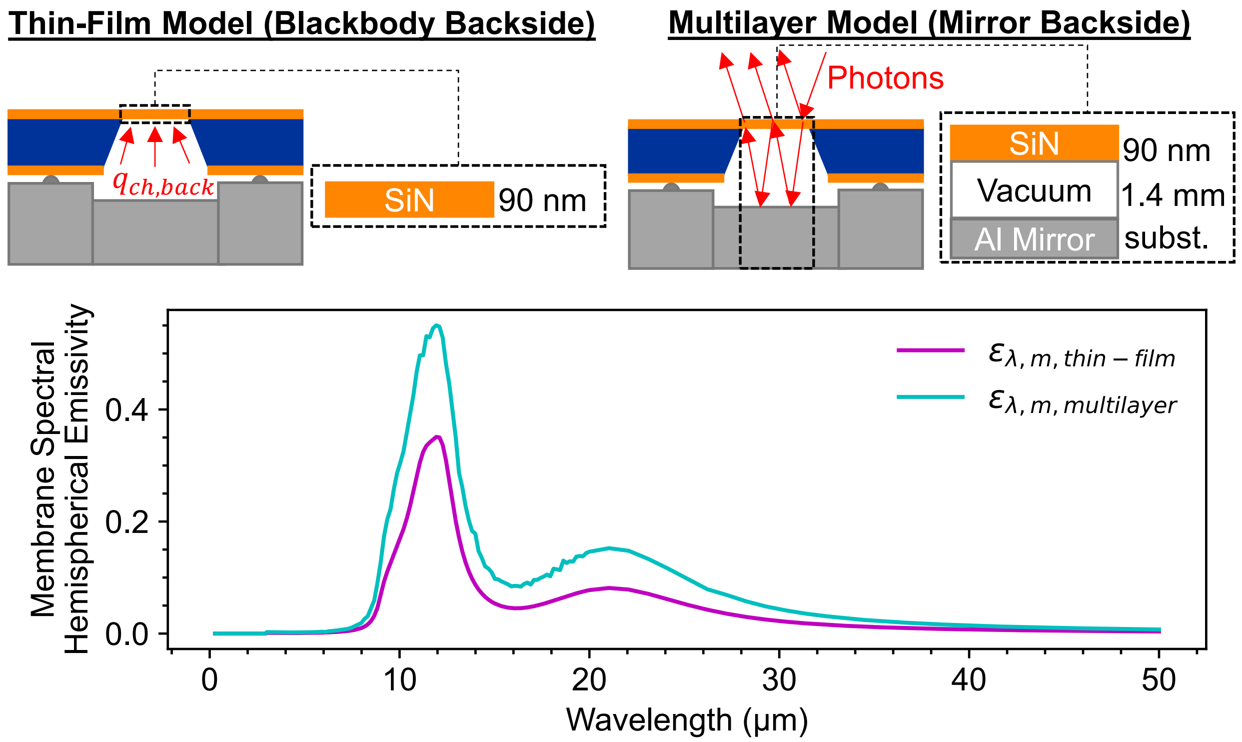


Figure 3.2. Depiction of both membrane emissivity models along with the respective hemispherical emissivity spectra.

3.1.2 Characteristic Matrix Formalism

The characteristic matrix formalism is used to calculate the spectral directional reflectivity ($\rho_{\lambda,\theta,m}$) and transmissivity ($\tau_{\lambda,\theta,m}$) of the membrane (multilayer or thin film) as follows [51]:

$$\rho_{\lambda,\theta,m} = \left(\frac{\eta_{inc}B - C}{\eta_{inc}B + C}\right)\left(\frac{\eta_{inc}B - C}{\eta_{inc}B + C}\right)^*, \tag{3.2}$$

$$\tau_{\lambda,\theta,m} = \frac{4\eta_{inc}Re(\eta_{out})}{(\eta_{inc}B + C)(\eta_{inc}B + C)^*}. \tag{3.3}$$

where η_{inc} and η_{out} are the admittance of the incident and output medium, respectively, and B and C represent the normalized total tangential magnetic and electric field at the front interface of the thin film assembly, respectively. The characteristic matrix (B, C) of the thin film assembly of z layers is a function of the refractive index (N_r), thickness (d_r), and propagation angle (θ_r) of each layer, along with the wavelength (λ) and the output medium admittance (η_{out}) as follows:

$$\begin{bmatrix} B \\ C \end{bmatrix} = \left\{\prod_{r=1}^{z}\begin{bmatrix} cos\delta_r & (isin\delta_r)/\eta_r \\ i\eta_r sin\delta_r & cos\delta_r \end{bmatrix}\right\}\begin{bmatrix} 1 \\ \eta_{out} \end{bmatrix}, \tag{3.4}$$

where the phase thickness (δ_r) and admittance (η_r) of each thin film layer is given by:

$$\delta_r = \frac{2\pi N_r d_r cos\theta_r}{\lambda}, \tag{3.5}$$

$$\eta_{r,TE} = N_r cos\theta_r \text{ for s-polarisation}, \tag{3.6}$$

$$\eta_{r,TM} = N_r / cos\theta_r \text{ for p-polarisation}. \tag{3.7}$$

Given that the wave vector in the plane of incidence is constant throughout the assembly of thin films, the cosine of the propagation angle may be substituted with the following equation (derived from Snell's law and trigonometric functions):

$$cos\theta_r = \sqrt{1 - \left(\frac{N_{inc} sin\theta_{inc}}{N_r}\right)^2}, \tag{3.8}$$

where θ_{inc} and N_{inc} are the incident angle and refractive index respectively. Using eqs. (3.2) and (3.3), the s-polarized and p-polarized transmissivity and reflectivity is determined, and the average of the real parts of both polarizations is taken to get the unpolarized reflectivity and transmissivity of the assembly of thin films. Note that the spectral refractive index of SiN is inferred from the Maxwell-Helmholtz-Drude model [52].

3.2 HEAT TRANSFER RATES

3.2.1 *Membrane Radiation*

The rate at which radiation is leaving the membrane (in W·m^{-2}) is calculated using the emissive power relation detailed in section 2.1.1, and is given by:

$$q_m(T_m) = A_{eff} \int_0^{\infty} \int_0^{2\pi} \int_0^{\frac{\pi}{2}} \varepsilon_{\lambda,\theta,m}(\lambda,\theta) I_{BB}(\lambda, T_m) \cos\theta \sin\theta d\theta d\phi d\lambda, \tag{3.9}$$

where λ, θ, and ϕ are the wavelength, azimuth angle and zenith angle of the radiation, respectively, $\varepsilon_{\lambda,\theta,m}$ is the spectral directional emissivity of the membrane, and I_{BB} is the spectral radiative blackbody intensity. The correction factor A_{eff} accounts for the fact that the thin film model (A_{eff} = 2) allows membrane radiation from both sides of the membranes, while the multilayer model (A_{eff} = 1) has a mirror backside (see Figure 3.2) that restricts radiation to the topside only.

3.2.2 *Atmospheric Irradiation*

Atmospheric irradiation is incident on the membrane only in the topside viewport range (i.e., $0 < \theta < \theta_v$), and the rate at which it is incident on the membrane is given by:

$$q_{atm}(T_{atm}) = \int_0^{\infty}\int_0^{2\pi}\int_0^{\theta_v} \tau_{\lambda,v}(\lambda)\varepsilon_{\lambda,\theta,atm}(\lambda,\theta)\alpha_{\lambda,\theta,m}(\lambda,\theta)I_{BB}(\lambda,T_{atm})\cos\theta\sin\theta d\theta d\phi d\lambda, \tag{3.10}$$

where $\tau_{\lambda,v}$ is the spectral transmissivity of the anti-reflection coated ZnSe viewport (Thorlabs WG71050-E3 [53]; see Figure 3.3), $\alpha_{\lambda,\theta,m}$ is the spectral directional absorptivity of the SiN membrane, and $\varepsilon_{\lambda,\theta,atm}$ is the spectral directional emissivity of the atmosphere. The ZnSe viewport is chosen for its selective transparency in the 8 μm – 13 μm wavelength range which minimizes parasitic heating from the atmosphere and the sun while not affecting the radiative cooling of the membrane. The spectral directional emissivity of the atmosphere has been modeled extensively in prior work, and Zhao et al. [31] provide a detailed summary of all developed models. A widely accepted [8], [24], [54]–[57] correlation for atmospheric emissivity is given by:

$$\varepsilon_{\lambda,\theta,atm}(\lambda,\theta) = 1 - \tau_{\lambda,atm}(\lambda)^{1/cos\theta}, \tag{3.11}$$

where $\tau_{\lambda,atm}$ is the atmospheric spectral transmissivity at normal incidence.

The atmospheric transparency—which is maximum in the 8 μm – 13 μm range—is the critical component that makes radiative cooling possible by allowing a significant portion of radiation to escape the Earth towards outer space. There are many factors that affect the atmospheric transmissivity; the most important ones are the atmospheric water content [58], the cloudiness [59], and the location on Earth [60]. The Moderate Resolution Atmospheric Transmission (MODTRAN) model analyzes optical measurements through the atmosphere and is widely used to extract the atmospheric spectral transmissivity for various climate conditions [43], [44]. In this work, experiments are conducted in the region of Ottawa, Canada, which is subject to variable atmospheric conditions. To account for this variability in the heat transfer model, $\tau_{\lambda,atm}$ is approximated for mid-latitude summer climate conditions as a weaker estimate, and for sub-arctic winter climate conditions as a stronger estimate (see Figure 3.3 for comparison).

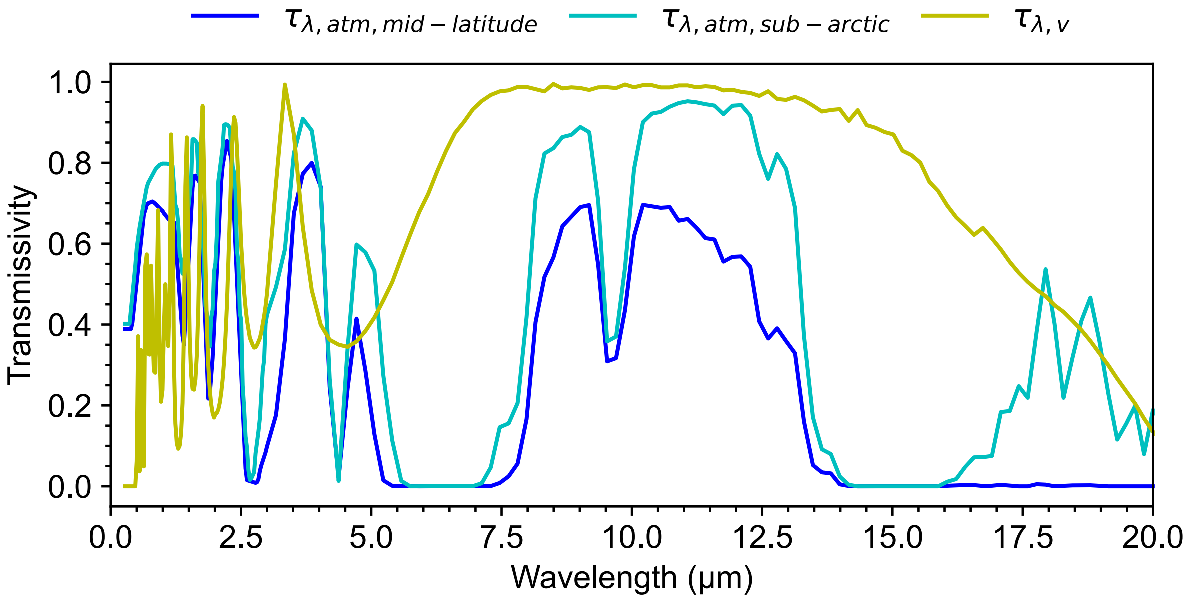


Figure 3.3. Atmospheric transmissivity ($\tau_{\lambda,atm}$) at normal incidence for mid-latitude summer and sub-arctic winter climate conditions computed by the MODTRAN model [43], [44], along with zinc selenide (ZnSe) viewport spectral transmissivity ($\tau_{\lambda,v}$) at normal incidence (Thorlabs WG71050-E3 [53]). $\tau_{\lambda,atm}$ and $\tau_{\lambda,v}$ are assumed null at wavelengths above 20 µm.

3.2.3 Chamber Irradiation

Due to the presence of the viewport, the rate at which vacuum chamber irradiation is incident on the membrane is divided into two components: (i) backside ($q_{ch,back}$), and (i) topside ($q_{ch,top}$) chamber irradiation.

Chamber irradiation on the backside of the membrane is calculated differently in the thin film model versus the multilayer model. In the thin film model, the backside is treated as a large isothermal surrounding. Therefore, the chamber irradiation incident on the backside of the membrane can be approximated by emission from a blackbody ($\varepsilon = 1$) at the chamber temperature (T_{ch}) as:

$$q_{ch,back}(T_{ch}) = \int_0^\infty \int_0^{2\pi} \int_0^{\frac{\pi}{2}} \alpha_{\lambda,\theta,m}(\lambda,\theta) I_{BB}(\lambda,T_{ch}) \cos\theta \sin\theta d\theta d\phi d\lambda \,. \tag{3.12}$$

The multilayer model considers $q_{ch,back} = 0$ since the optical fiber holder and membrane mount are treated as perfect mirrors that radiate no heat to the membrane.

The chamber irradiation incident on the topside of the membrane is given by:

$$q_{ch,top}(T_{ch},\theta_v) = \int_0^\infty \int_0^{2\pi} \int_0^{\frac{\pi}{2}} \varepsilon_{\lambda,\theta,ch}(\lambda,\theta_v) \alpha_{\lambda,\theta,m}(\lambda,\theta) I_{BB}(\lambda,T_{ch}) \cos\theta \sin\theta d\theta d\phi d\lambda \,, \tag{3.13}$$

where $\varepsilon_{\lambda,\theta,ch}$ is the spectral directional emissivity of the surrounding topside chamber. The topside chamber may be treated as a large isothermal enclosure that surrounds the membrane, except in the viewport range (i.e., $0 < \theta < \theta_v$) where radiation may be transmitted out of the chamber. The radiation that is not transmitted through the viewport—i.e., radiation that is reflected by the viewport or chamber walls—is assumed to have no chance of leaving the chamber. Hence, the spectral emissivity of the chamber is calculated as:

$$\varepsilon_{\lambda,\theta,ch}(\lambda,\theta_v) = \begin{cases} 1 - \tau_{\lambda,v}(\lambda), & 0 < \theta < \theta_v \\ 1, & \theta_v < \theta < \dfrac{\pi}{2}. \end{cases} \tag{3.14}$$

See Figure 3.3 for the transmissivity spectrum of the ZnSe viewport ($\tau_{\lambda,v}$).

3.2.4 *Solar Irradiation*

Solar radiation reaching the membrane can be categorized into two components: (i) direct solar radiation, and (ii) diffuse solar radiation (i.e., atmospherically-scattered radiation) [46]. The amount of each solar radiation component reaching the membrane is greatly affected by the atmospheric conditions, the geographical location, and the time of day and year. The AM1.5 spectrum (see Figure 3.4) is a commonly accepted international standard for the global (i.e., sum of direct and diffuse) solar intensity [61]. It is widely used for characterizing the performance of solar panels, and for characterizing the global spectral solar irradiance in the context of radiative cooling [24], [31], [56], [60]. The net solar irradiance of the AM1.5 spectrum is 1000 $W \cdot m^{-2}$, which is representative of the peak conditions in the Ottawa, Canada, region (see Appendix B). Therefore, global solar radiation incident on the membrane can be calculated as:

$$q_{sun}(\theta_{sun}) = \int_0^\infty \tau_{\lambda,v}(\lambda)\alpha_{\lambda,\theta,m}(\lambda,\theta_{sun})I_{AM1.5}(\lambda)d\lambda, \tag{3.15}$$

where θ_{sun} is the fixed angle at which the sun is facing the membrane, and depends on latitude, time of day, and day number of the year.

Assuming the membrane can be treated as a horizontal surface on the surface of the Earth, the angle at which the sun is facing the membrane can be calculated as [62]:

$$\theta_{sun} = \cos^{-1}(\cos\Phi\cos\delta\cos\omega + \sin\Phi\sin\delta), \tag{3.16}$$

where $\Phi = 45^{\circ}$ is the latitude of Ottawa, ω is the hour angle, and δ is the declination angle. The hour angle (ω) is the angular displacement of the sun east or west of the local meridian due to the rotation of the Earth, with 15° per hour (with afternoon being positive, and morning being negative). The declination angle is given by the equation of Cooper [62]:

$$\delta = 23.45 \sin\left(360 \frac{284 + n}{365}\right) \tag{3.17}$$

where n is the day number of the year. See Appendix C for a θ_{sun} calculation example.

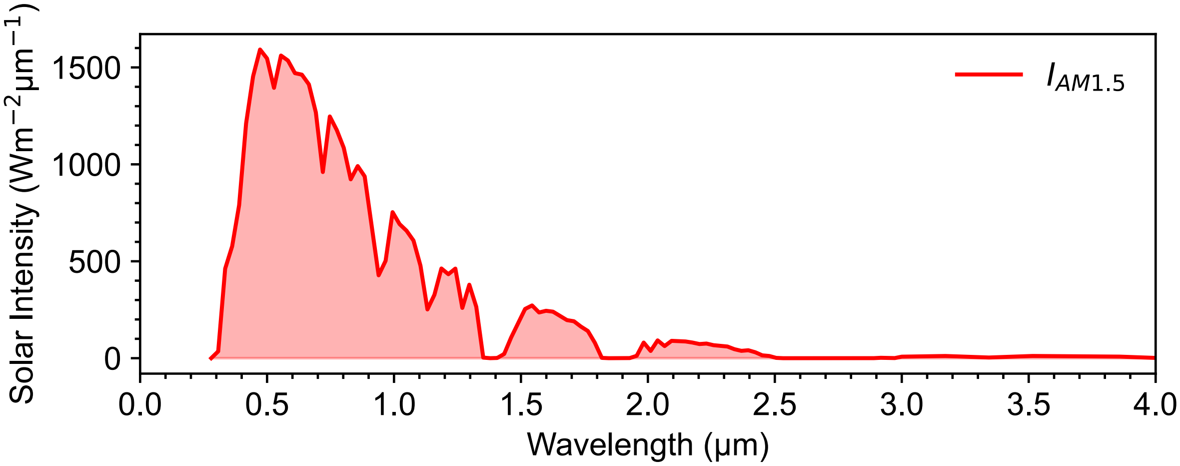


Figure 3.4. AM1.5 solar intensity spectrum, representing the distribution of solar power (in watts per square meter per nanometer of bandwidth) as a function of wavelength [61].

3.2.5 Frame Conduction

Conductive heat transfer from the silicon frame to the SiN membrane depends on the membrane geometry and was modeled recently in two different studies [37], [39]. Considering that only conductive and radiative heat transfer occurs on the membrane in this work (no convection at high vacuum), the rate of conduction from the silicon frame to the SiN membrane can be calculated from the fraction of heat transfer occurring via conduction versus radiation (x_{cond}) as [37]:

$$q_{cond}(T_{ch}, T_m) = x_{cond} \cdot \frac{2\varepsilon_m \sigma (T_{ch}^4 - T_m^4)}{1 - x_{cond}}, \tag{3.18}$$

where ε_m is the total hemispherical emissivity of the membrane, and $\sigma = 5.67 \times 10^{-8}$ W·m^{-2}·K^{-1} is the Stefan-Boltzmann constant. Since T_m is unknown, and since ε_m was shown to vary negligibly between 200 K to 300 K in [37], the assumption $T_m \approx T_{ch}$ is taken when evaluating ε_m (see section 2.1.2 for ε_m calculation details). In turn, x_{cond} is dependent on a membrane geometry factor (βr_{eff}) and is defined as [37]:

$$x_{cond} = \frac{2}{\beta r_{eff}} \frac{I_1(\beta r_{eff})}{I_0(\beta r_{eff})}, \tag{3.19}$$

where I_N is the Nth-order modified Bessel function of the first kind. βr_{eff} is defined as [37]:

$$\beta r_{eff} = \left(\sqrt{\frac{8\sigma \varepsilon_m T_{ch}^3}{kd}} \right) \cdot \left(1.252 \frac{L}{2} \right), \tag{3.20}$$

where k is the thermal conductivity of the membrane ($k \approx 2.7$ W·m^{-1}·K^{-1} for SiN [38]), $d \approx 90$ nm is the thickness of the membrane, and $L \approx 6$ mm is the side length of the square membrane. Here, the silicon frame is assumed to be at ambient chamber temperature (i.e., $T_{Si} \approx T_{ch}$). In this analysis, the effect of thermal contact resistance at the SiN-Si interface is assumed to be negligeable compared to the conductivity of the SiN membrane.

3.3 NUMERICAL SOLVER

For measured values of T_{atm}, T_{ch}, θ_v, and θ_{sun} during a radiative cooling experiment, the theoretical membrane temperature (T_m) is solved numerically by iterating values of T_m until the energy balance of eq. (3.1) reaches steady state (i.e., q_{net} = 0). Consequently, the theoretical membrane temperature drop can be determined with $\Delta T_m = T_{ch} - T_m$ (assuming $T_m = T_{ch}$ initially). The numerical solver is coded in Python (see Appendix J) and it takes approximately 1.5 seconds to complete a single temperature increment. In this case, temperature increments of 0.1 K are specified, which means the solver requires approximately 135 seconds when steady state is reached with a 9 K membrane temperature drop (i.e., 90 temperature increments of 0.1 K).

4 EXPERIMENTAL METHOD

This work assesses the temperature drop of a freestanding SiN membrane upon radiative cooling by tracking its mechanical resonance frequency shift. Figure 4.1 and Figure 4.2 display a schematic and picture of the experimental setup, respectively. There are four main systems to the experimental setup: (i) the optical fiber interferometer, (ii) the vacuum chamber, (iii) the temperature data acquisition (DAQ), and (iv) the power distribution. Each system is described in sections 4.1 to 4.4 below. A detailed procedure of the experiment can be found in section 4.5.

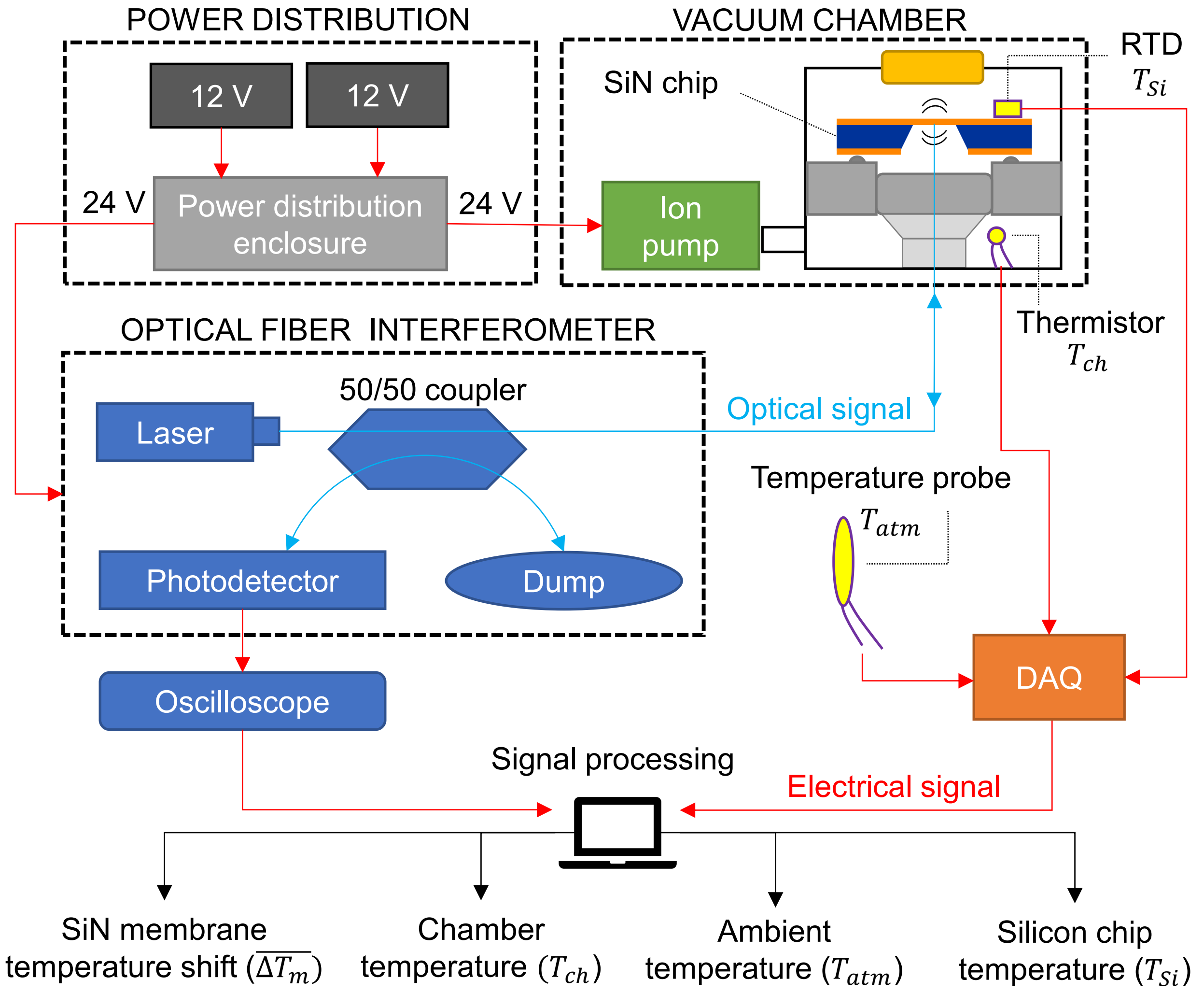


Figure 4.1. Experimental setup schematic.

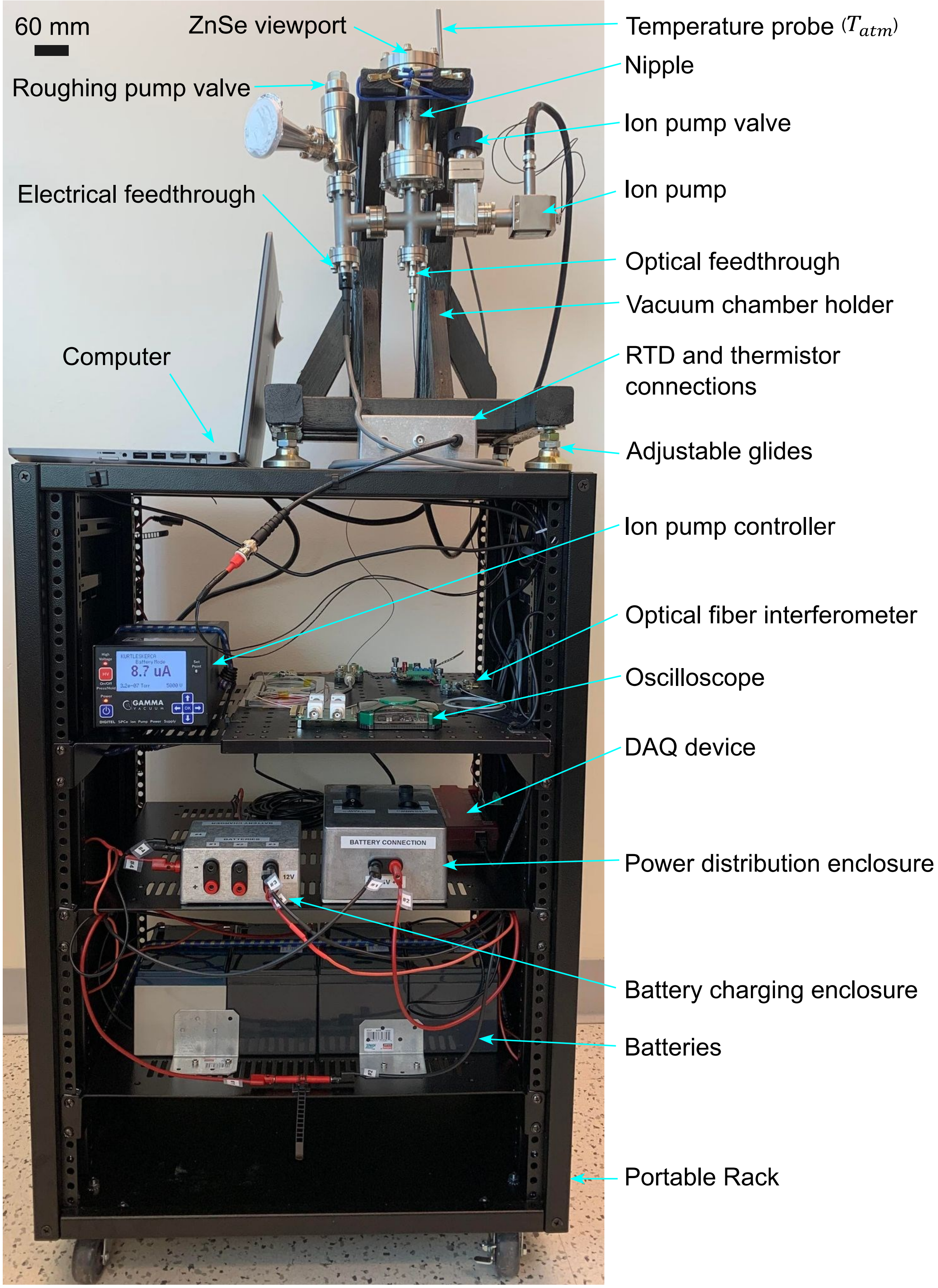


Figure 4.2. Experimental Setup Picture.

4.1 OPTICAL FIBER INTERFEROMETER

Laser interferometry is used for high-sensitivity displacement measurements. In this work, a cleaved optical fiber tip interferometer [63] is used to measure the surface displacement of a vibrating freestanding SiN membrane, from which the resonance frequency of the membrane (f_m) is inferred (see Figure 4.3).

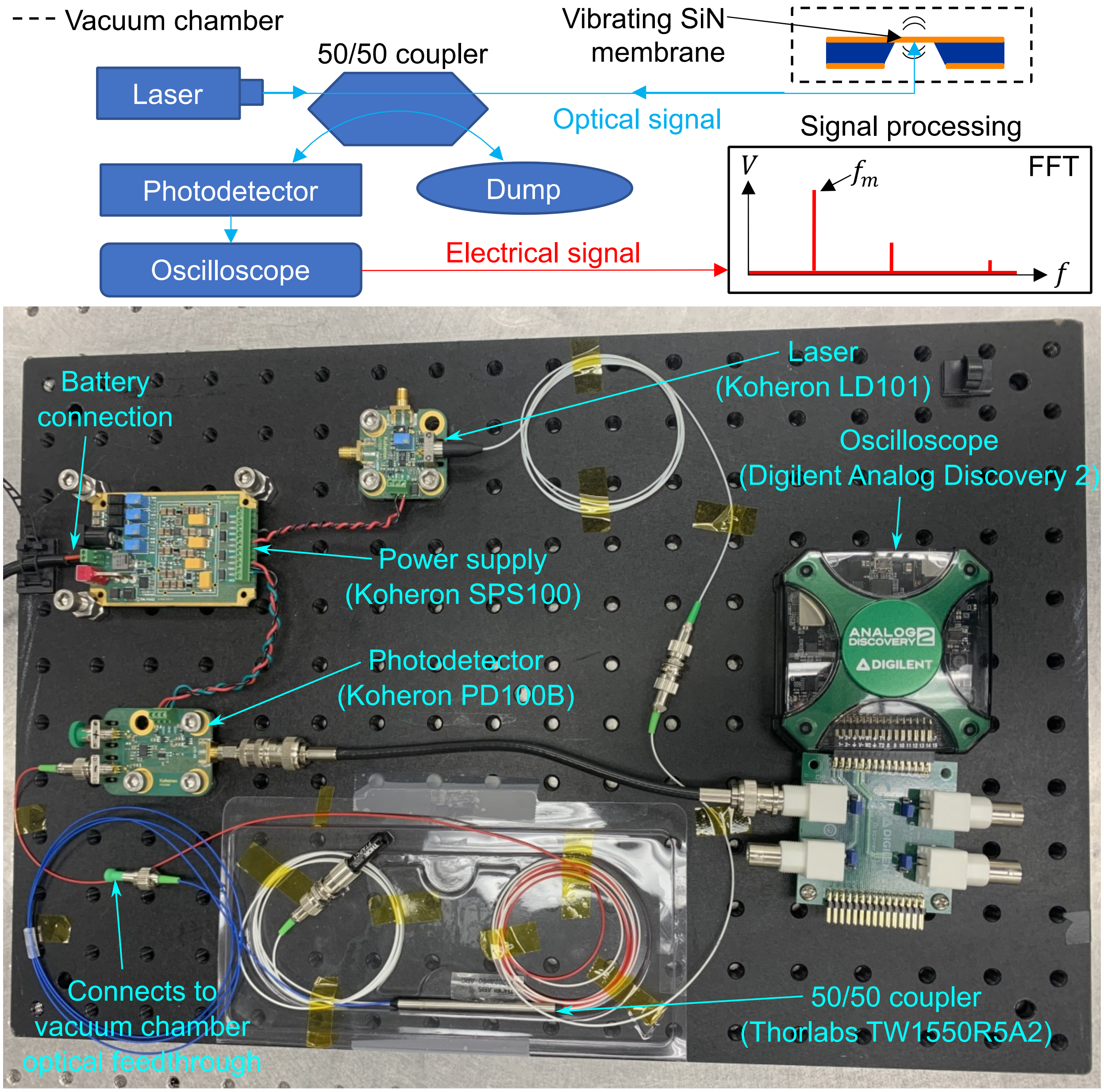


Figure 4.3. Optical fiber interferometer components.

The optical fiber interferometer consists of a monochromatic (1550 nm) laser (Koheron LD101), a 50:50 single-mode optical fiber coupler (Thorlabs TW1550R5A2), a photodetector (Koheron PD100B), and an oscilloscope (Digilent Analog Discovery 2). The 50:50 coupler means 50% of the laser light is dumped at each passing through the coupler. In this case, 50% of the laser light is dumped before reaching the membrane, and 50% of the reflected light is dumped before reaching the photodetector. The photodetector then converts the corresponding photon energy to an electrical signal, which can be analyzed to extract the membrane resonance frequency. Section 4.1.1 elaborates on the working principle of the optical fiber interferometer and how the membrane resonance frequency is obtained from the photodetector output voltage, and section 4.1.2 elaborates the relation between the membrane resonance frequency shift and temperature shift.

4.1.1 Working Principle of the Optical Fiber Interferometer

The working principle of the optical fiber interferometer is based on the optical interference occurring between the laser light reflected by the membrane and the laser light reflected by the optical fiber glass-vacuum interface (roughly 4% reflection coefficient). This wave interference is depicted in Figure 4.4a. The photodetector captures the reflected light, and its output voltage signal (V) is correlated to the distance between the fiber tip and the membrane (d) and the wavelength of the laser light (λ) by:

$$V = V_{mid} + A\cos\left(\frac{4\pi d}{\lambda}\right), \tag{4.1}$$

where $V_{mid} = (V_{max} + V_{min})/2$, $A = (V_{max} - V_{min})/2$, and V_{max} and V_{min} are the voltages corresponding to the maximum constructive and destructive interference, respectively (see Figure 4.4b). When the membrane is vibrating, the distance between the optical fiber tip and the membrane (d) changes periodically in correspondence to the vibration frequency. Accordingly, a variation in d results in a variation in photodetector output voltage (ΔV). The distance d corresponding to the most sensitive signal occurs when the two interfering waves are in quadrature, meaning $d = \lambda/8, 3\lambda/8, 5\lambda/8$, and so on. This is referred to as a quasi "linear range" and it results in a photodetector output signal $V \approx V_{mid}$ (see Figure 4.4b). In this range, the derivative of V with respect to d results in the following relation:

$$\Delta d = \frac{\lambda \Delta V}{2\pi(V_{max} - V_{min})}, \tag{4.2}$$

which establishes that Δd is proportional to ΔV. Precisely quantifying V_{max} and V_{min} is only important for quantifying the membrane vibration amplitude, which is not critical in this work. On the other hand, the resonance frequency of the vibrating SiN membrane (f_m) can be determined by computing the Fast Fourier Transform (FFT) of the photodetector output voltage signal—to obtain its frequency spectrum—and finding the frequency at which the FFT amplitude is largest (i.e., the peak). The frequency shift of the peak (Δf_m) upon radiative cooling is used to calculate the corresponding temperature shift (see section 4.1.2 for more details).

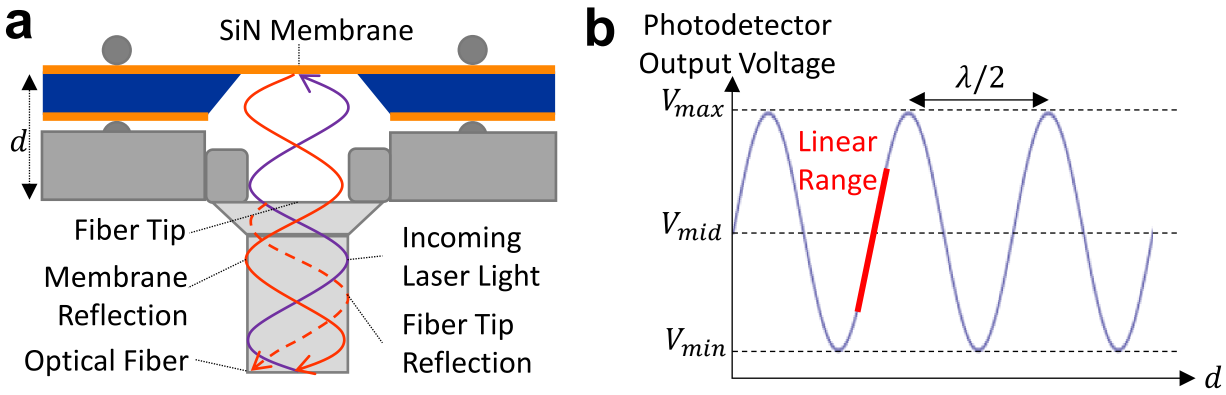
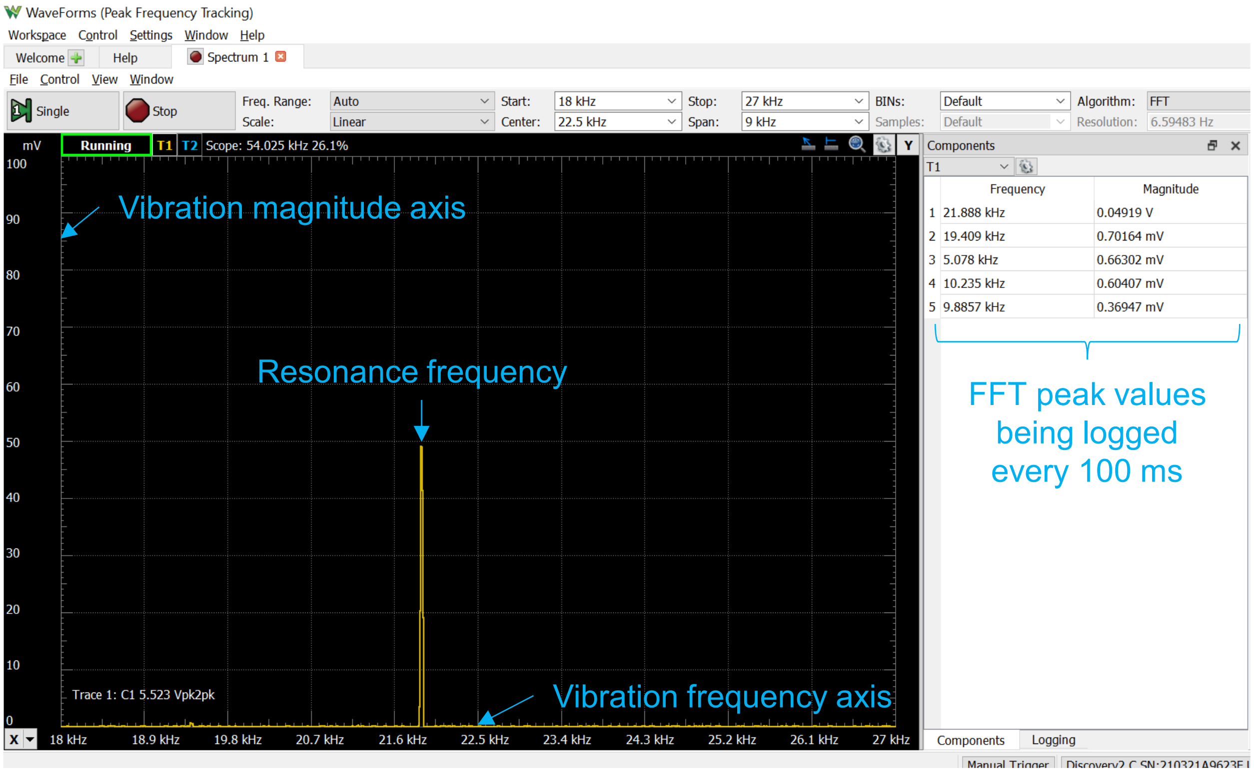


Figure 4.4. a) Optical interference pattern between the laser light reflected by the membrane and the laser light reflected by the optical fiber tip, and b) correlation between the photodetector output voltage and the distance between the optical fiber tip and the membrane.

The manufacturer of the oscilloscope used in this work (Digilent) provides a free software—Waveforms—which includes a spectrum analyzer function. This function is used to visualize the FFT of the oscilloscope signal and extract the frequency peak (i.e., membrane resonance frequency) in real time for all experiments. An overview of the Waveforms software is shown in Figure 4.5 and a detailed procedure of the resonance frequency logging steps is detailed in Appendix D.



Freq. Range: Auto
Scale: Linear
Start: 18 kHz
Center: 22.5 kHz
Stop: 27 kHz
Span: 9 kHz
BINs: Default
Samples: Default
Algorithm: FFT
Resolution: 6.59483 Hz
Scope: 54.025 kHz 26.1%
Trace 1: C1 5.523 Vpk2pk

Components T1

	Frequency	Magnitude
1	21.888 kHz	0.04919 V
2	19.409 kHz	0.70164 mV
3	5.078 kHz	0.66302 mV
4	10.235 kHz	0.60407 mV
5	9.8857 kHz	0.36947 mV

Figure 4.5. Digilent Waveforms spectrum analyzer overview.

4.1.2 Relation between the Membrane Temperature and Resonance Frequency Shift

The mechanical resonance frequency of the SiN membrane (f_m) is linked to its tensile stress (ς) and vibrational modeshape (n, j) by [64]:

$$f_{m_{(n,j)}} = \sqrt{\frac{\varsigma(n^2 + j^2)}{4\rho L^2}}, \tag{4.3}$$

where ρ is the density, L is the side length, and (n, j) represents the number of antinodes in the vibrational mode shape of the square membrane. The temperature drop of the SiN membrane during radiative cooling results in an increase in membrane tensile stress, which leads to a proportional rise in resonance frequency (Δf_m) [65]–[67]. For a membrane at uniform temperature, the membrane average temperature shift can be approximated from the resonance frequency shift (Δf_m) by [66], [37]:

$$\overline{\Delta T_m} = -[1+e]\frac{2\varsigma_0(1-\nu)}{a_{CTE}Ef_0}\Delta f_m, \tag{4.4}$$

where a_{CTE} is the thermal expansion coefficient, E is Young's modulus, f_0 is the initial resonance frequency (i.e., before the frequency shift), and ν is the Poisson ratio of the membrane. The initial tensile stress of the membrane is derived from eq. (4.3) as $\varsigma_0 = 4f_0^2\rho L^2 \cdot (n^2+j^2)^{-1}$. According to Zhang et al. [37], the approximation that the temperature profile of the membrane is uniform leads to an underestimation error of the membrane temperature shift, which can be compensated with a correction factor ($e \leq 0.2$) obtained from finite element simulations described in [37]. The value of e is dependent on βr_{eff} and (n, j) and can be obtained from Figure 4.6 (taken from Fig. 4 in [37]). In this work, the resonance frequency of the vibrational mode shape (n = 1, j = 1) is tracked, which results in $\beta r_{eff} \approx 8.3$ (calculated with eq. (3.20)) and a correction factor $e \approx 0.12$.

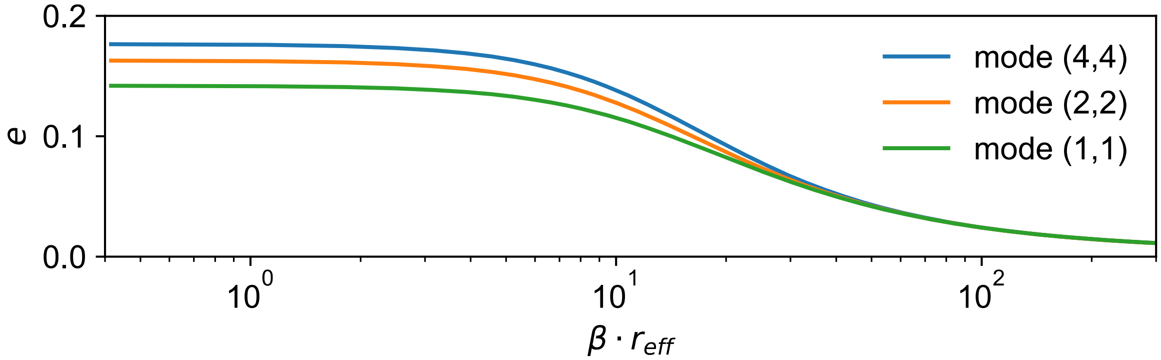


Figure 4.6. Underestimation error of the membrane temperature shift, by Zhang et al. [37].

The membrane dimensions were measured in [38] yielding thicknesses (d) of 90 ± 1.7 nm and side lengths (L) of 6 ± 0.015 mm. A description of the membrane fabrication process from low stress, low pressure chemical vapor deposition (LPCVD), SiN is available in [45]. The material properties of low stress LPCVD SiN are obtained from literature [68]–[73] as a_{CTE} = 2.2 ± 0.1 × 10^{-6} K^{-1}, E = 290 ± 90 GPa, ν = 0.25 ± 0.03, and ρ = 3000 ± 100 kg·m^{-3}.

The uncertainties of the material properties lead to an uncertainty in $\overline{\Delta T_m}$ in eq. (4.4), which is calculated using the propagation of uncertainty rule:

$$u_{\overline{\Delta T_m}} = \sqrt{\left(\frac{\partial \overline{\Delta T_m}}{\partial a_{CTE}} u_{a_{CTE}}\right)^2 + \left(\frac{\partial \overline{\Delta T_m}}{\partial L} u_L\right)^2 + \left(\frac{\partial \overline{\Delta T_m}}{\partial a_E} u_E\right)^2 + \left(\frac{\partial \overline{\Delta T_m}}{\partial \nu} u_\nu\right)^2 + \left(\frac{\partial \overline{\Delta T_m}}{\partial \rho} u_\rho\right)^2}, \tag{4.5}$$

where $\frac{\partial \overline{\Delta T_m}}{\partial x}$ represents the partial derivative of $\overline{\Delta T_m}$ with respect to the material property x, and u_x is the uncertainty of the material property x.

4.2 VACUUM CHAMBER

A vacuum environment is necessary in this work to remove convective heat transfer effects—i.e., to enhance radiative cooling—and prevent overdamping of the membrane vibrations. Figure 4.2 shows a picture of the vacuum chamber used in this work (see Appendix E for more pictures). Section 4.2.1 explains how the membrane is mounted inside the chamber, section 4.2.2 elaborates on the vacuum chamber holder design, and section 4.2.3 details the vacuum pumping method.

4.2.1 Membrane Mounting

The membrane is mounted inside the nipple of the vacuum chamber as closely as possible to the anti-reflection coated ZnSe viewport (Thorlabs WG71050-E3) to maximize the field of view of the membrane (see Figure 4.7a). The stainless-steel membrane mount is anchored to the chamber walls with spring plungers (see Figure 4.7b). Two different methods were used to fix the membrane on the mount: (i) magnetic spheres, and (ii) vacuum-compatible tape. In the first method, the membrane is placed between three sets of magnetic spheres to hold it in a fixed position on the mount (see Figure 4.7b). This method is optimal for higher quality factor measurements (i.e., for less dissipation, meaning longer vibrations); however, it is not strong enough to hold the membrane in place with a resistance temperature detector (RTD) on the Si frame (as shown in Figure 4.7a). Therefore, the second method is used for measurements with the RTD, where vacuum compatible tape is used to anchor the membrane on the mount (see Figure 4.7a). In all cases, the position of the membrane is chosen to ensure an optimal alignment with the optical fiber and the viewport.

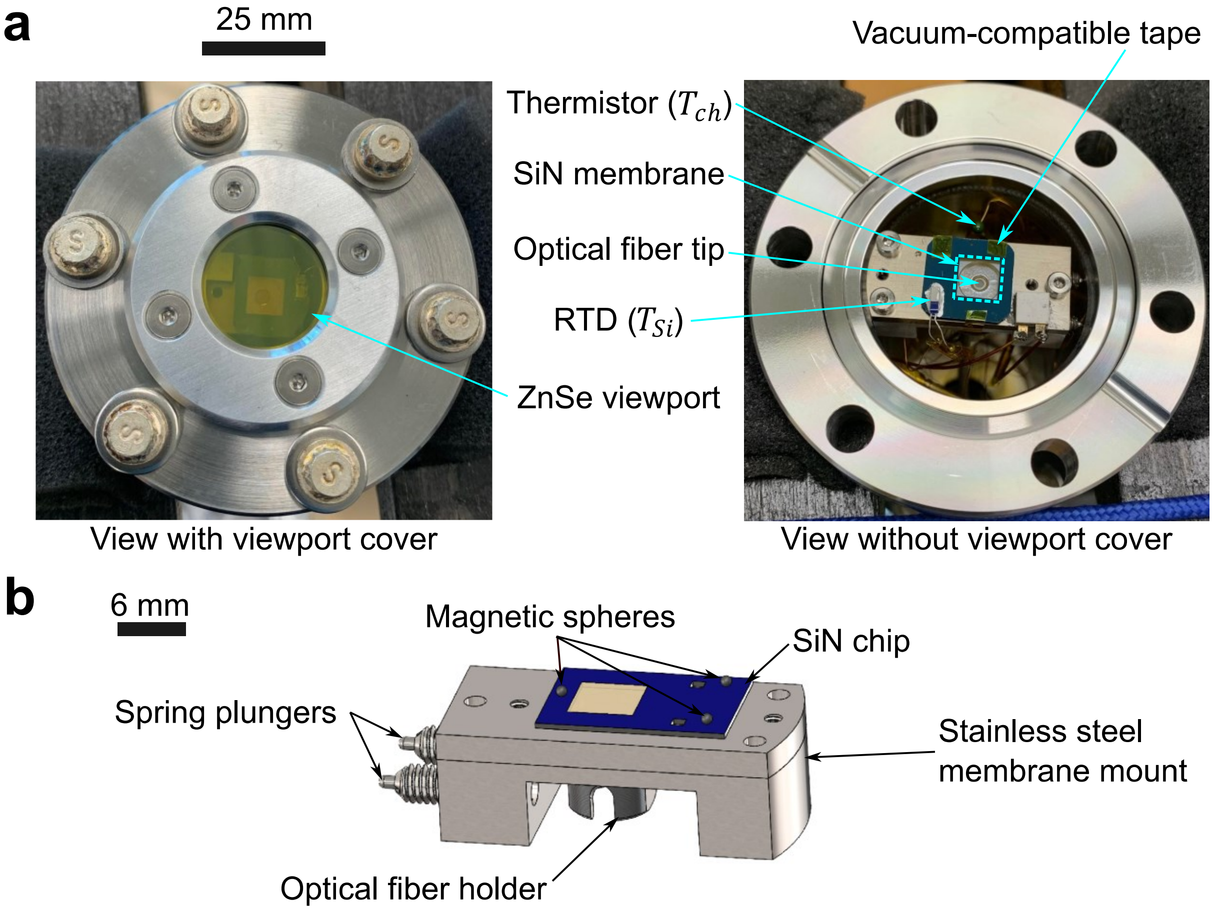


Figure 4.7. Membrane mounting a) picture, and b) CAD drawing.

4.2.2 Vacuum Chamber Holder

To maximize the radiative cooling power of the SiN membrane, its normal must be pointing the zenith. Therefore, a wooden frame is designed to hold the vacuum chamber such that this condition is respected. As shown in Figure 4.2, the wooden frame holds the vacuum chamber such that the membrane is facing the sky, and adjustable glides allow the levelling of the chamber to ensure the normal of the membrane is pointing the zenith.

4.2.3 Pumping Method

The vacuum chamber is kept at high vacuum ($P \approx 10^{-6}$ hPa) with an ion pump (Gamma Vacuum 3S Titan DI; see Figure 4.2). This type of pump is selected because it is lightweight, small, and can be powered by batteries, so it ensures the portability of the vacuum chamber. The ion pump controller (Gamma Vacuum SPC) is used to distribute power to, and control, the ion pump. Ion pumps require initial rough pumping to remove the bulk gas content inside the chamber (e.g., to 10^{-4} hPa for the Gamma Vacuum 3S Titan DI). In this work, a turbomolecular pump (Edwards T-Station 85) is used for initial pumping (see Figure 4.8). Once the vacuum chamber is pumped down to at least 10^{-4} hPa with the turbomolecular pump, the ion pump can reach and maintain a high vacuum pressure in the 10^{-6} to 10^{-7} hPa range inside the chamber.

Exposure to atmospheric contaminants (e.g., water vapor, dust, oils) can reduce the performance and lifetime of the ion pump. For this reason, the ion pump is kept at high vacuum at all times after the initial rough pump. In order to keep the ion pump at high vacuum when the chamber needs to be opened (i.e., when work needs to be performed inside the chamber), the following procedure is taken:

1. Connect the turbomolecular pump to the roughing pump valve with the vacuum tube (see Figure 4.8). Keep the roughing pump valve closed.
2. Pump down the vacuum tube to approximately the same pressure as the vacuum chamber (i.e., the pressure read by the ion pump controller).
3. Close the ion pump valve to isolate the ion pump from the chamber and open the roughing pump valve to transfer the pumping of the chamber to the turbomolecular pump.
    - Pumping down the vacuum tube is necessary to avoid a sudden large pressure shift in the chamber which could send a pressure wave that breaks the membrane.
4. Shut down the turbo pump and wait for the vacuum chamber pressure to rise back to ambient pressure. Note that, while this occurs, the ion pump is maintaining high vacuum in the small environment isolated from the chamber by the ion pump valve.
5. Perform the work inside the vacuum chamber with latex gloves to avoid contaminating the chamber with skin contaminants. Once the work is performed, clean the vacuum chamber with acetone and isopropanol. Close the chamber with new copper gaskets for a proper seal.
6. Once the chamber is closed and properly sealed, make sure that the vacuum tube connection is secure, open the roughing pump valve, and start the turbomolecular pump for pump down.
    - At this stage, baking the vacuum chamber can promote outgassing, and thus reduce high gas loads in the chamber. If the vacuum chamber requires baking, refer to Appendix F for additional steps.
7. Once the chamber pressure reaches at least 10^{-4} hPa, open the ion pump valve and close the roughing pump valve to transfer the pumping of the chamber back to the ion pump.

- In this step, it is good practice to continue pumping the chamber with the turbomolecular pump and keep the ion pump isolated for an extended period of time (usually 24 hours) to minimize the ion pump exposure to outgassing contaminants.

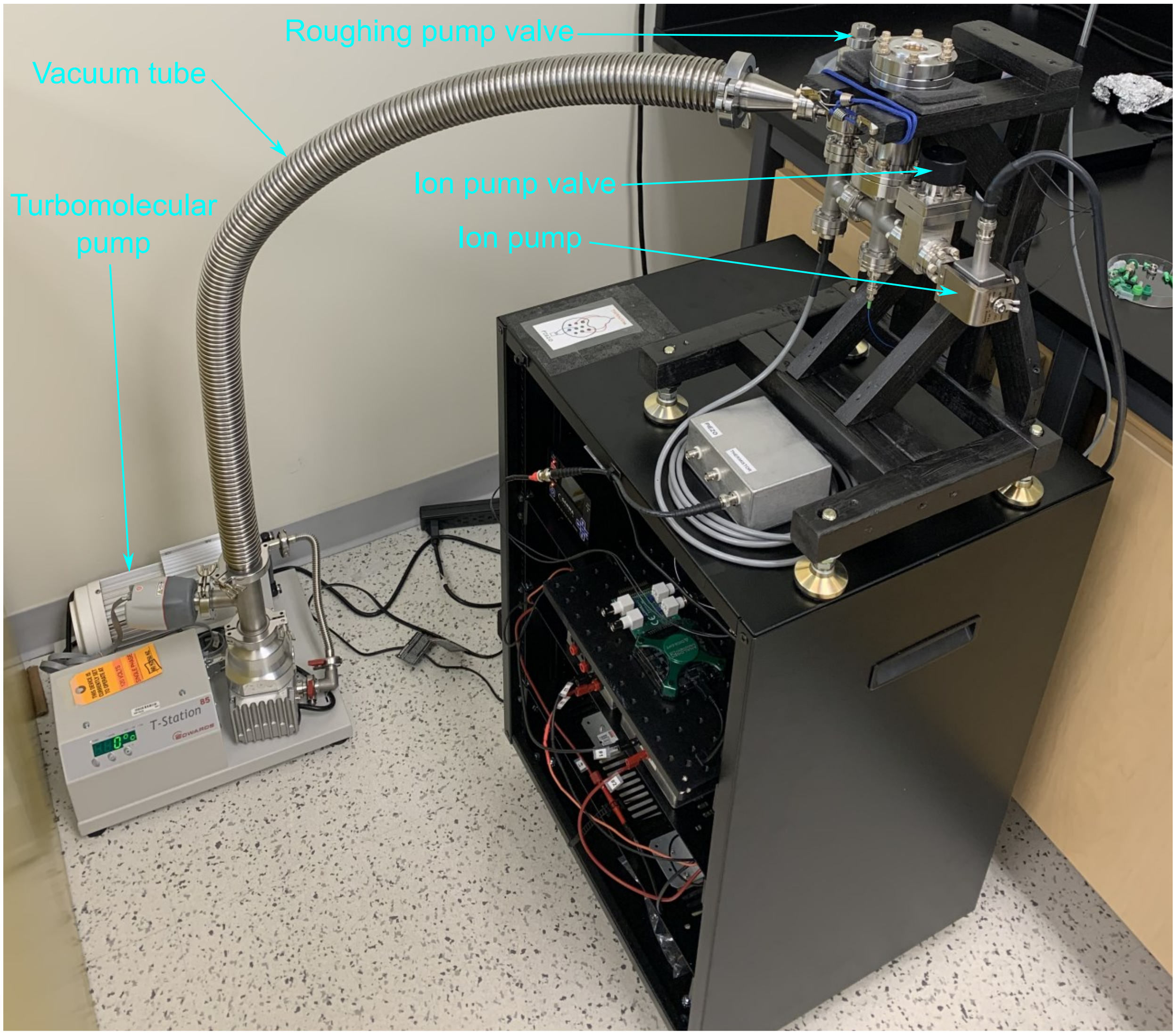


Figure 4.8. Turbomolecular pump connection to vacuum chamber for rough pumping.

4.3 TEMPERATURE DATA ACQUISITION

The temperature DAQ system uses a DAQ device (LabJack U6-Pro) to continuously measure the following temperatures:

1. The ambient atmospheric temperature (T_{atm}) with a temperature probe (LabJack EI-1034) located outside of the chamber (see Figure 4.2),
2. The temperature inside the vacuum chamber (T_{ch}) with a thermistor (Measurement Specialties 55036 10 kOhm precision glass NTC) located inside the nipple of the chamber (see Figure 4.7a),
3. The silicon frame temperature (T_{Si}) using an RTD (TE Connectivity 1 kOhm PTF) silver pasted to the Si frame (see Figure 4.7a).

The temperature probe is selected because of its sturdy, waterproof, design for outdoor measurements. The RTD is selected because of its rectangular surface area that facilitates its adhesion to the silicon frame. The thermistor is selected for its excellent tolerance (±0.1 °C between 0 °C and 70 °C). Figure 4.9 displays the connection of each temperature sensor to the analog inputs of the DAQ device. Since the DAQ device (LabJack U6-Pro) measures the output voltage of each temperature sensor, sections 4.3.1 to 4.3.3 detail the scaling equations to obtain the respective temperatures.

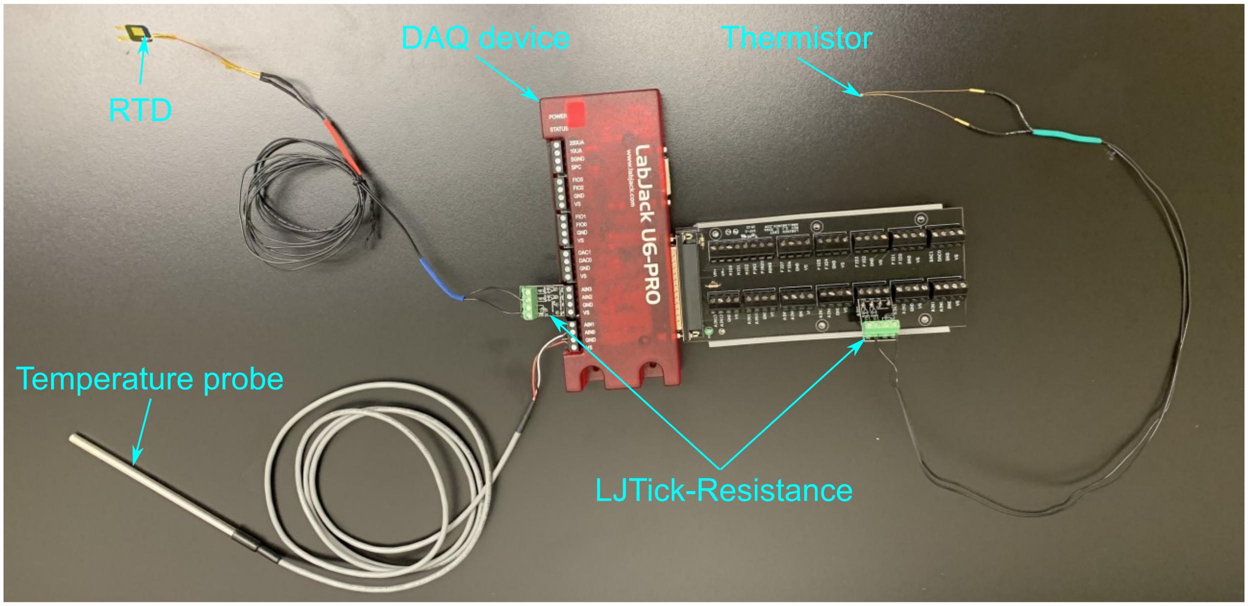


Figure 4.9. Temperature sensor connections to the analog inputs of the DAQ device.

4.3.1 Ambient Atmospheric Temperature

The output voltage of the temperature probe (LabJack EI-1034) is linearly proportional to its temperature, with a typical room temperature accuracy of ±0.2 K [74]. The scaling equation to convert the output voltage of the temperature probe ($V_{EI-1034}$) to its temperature is given by [74]:

$$T_{atm} = T_{EI-1034} = 55.56 \cdot V_{EI-1034} - 17.78. \tag{4.6}$$

4.3.2 Vacuum Chamber Temperature

The thermistor (Measurement Specialties 55036 10 kOhm precision glass NTC) changes resistance with temperature in a non-linear way, with a tolerance of ±0.1 °C between the temperatures of 0 °C to 70 °C. The relation between the resistance (Ω_{th}) and temperature (T_{th}) of the thermistor is given by the Steinhart-Hart β-parameter equation:

$$T_{ch} = T_{th} = \frac{\beta}{ln\left(\frac{\Omega_{th}}{\Omega_{th,298.15} \cdot e^{-\beta/298.15}}\right)} \tag{4.7}$$

where β = 3978 K is the temperature coefficient of the thermistor, and $\Omega_{th,298.15}$ = 10,000 Ohm is the thermistor resistance at room temperature (i.e., 25 °C or 298.15 K). To make use of eq. (4.7), the output voltage of the thermistor (V_{th}) read by the DAQ device must be converted to resistance (Ω_{th}). The LabJack LJTick-Resistance (see schematic in Figure 4.10a) [75] is used as an add-on accessory to the LabJack U6-Pro to provide access to a stable 2.5 V reference voltage (V_{ref}) and a fixed resistance (Ω_{fixed}) of 10,000 Ohm. By connecting the two leads of the thermistor to the reference voltage source and the input lead of the LJTick-Resistance (i.e., V_{ref} and VINA in Figure 4.10a, respectively), a voltage divider (see Figure 4.10b) is formed between the fixed resistance of the LJTick-Resistance and the unknown thermistor resistance. Therefore, the thermistor resistance can be calculated from the output voltage (V_{th}) by:

$$\Omega_{th} = \frac{10{,}000 \cdot (2.5 - V_{th})}{V_{th}}. \tag{4.8}$$

where Ω_{fixed} = 10,000 Ohm and V_{ref} = 2.5 V are the fixed reference resistance and voltage of the LJTick-Resistance, respectively [75].

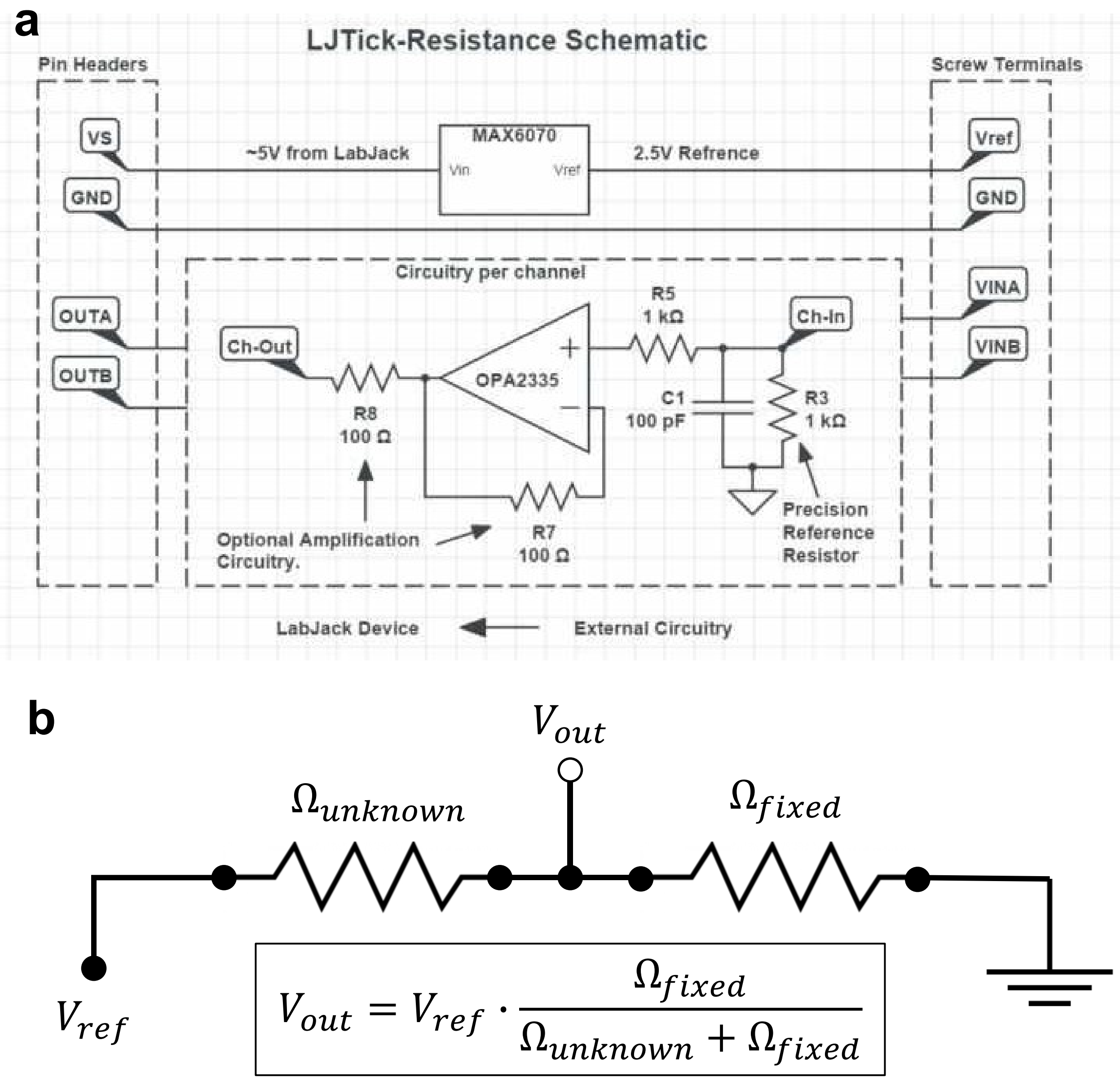


Figure 4.10. a) LJTick-Resistance schematic [75], and b) voltage divider schematic.

4.3.3 Silicon Frame Temperature

The RTD (TE Connectivity 1 kOhm PTF) changes resistance with temperature in a non-linear way, with a tolerance of ±0.43 °C at room temperature. The relation between the resistance (Ω_{RTD}) and temperature (T_{RTD}) of the RTD is given by the manufacturer as:

$$\Omega_{RTD} = \Omega_{RTD,273.15} \cdot (1 + a \cdot T_{RTD} + b \cdot T_{RTD}^2), \tag{4.9}$$

where $\Omega_{RTD,273.15}$ = 1000 Ohm is the nominal resistance of the RTD at 273.15 K (i.e., 0 °C), and a = 3.9083×10^{-3} and b = -5.665×10^{-7} are the coefficients of the RTD. Consequently, T_{RTD} can be obtained from Ω_{RTD} by applying the quadratic equation to eq. (4.9) as:

$$T_{Si} = T_{RTD} = \frac{-a + \sqrt{a^2 - 4 \cdot b \cdot \left(1 - {\Omega_{RTD}}/{\Omega_{RTD,273.15}}\right)}}{2 \cdot b}. \tag{4.10}$$

Like the thermistor, the RTD is connected to the LabJack U6-Pro via a LJTick-Resistance, but with a fixed resistance of 1,000 Ohm. Hence, a voltage divider (see Figure 4.10b) is formed between the RTD resistance and the fixed resistance of the LJTick-Resistance, and Ω_{RTD} can be calculated from the output voltage (V_{RTD}) by:

$$\Omega_{RTD} = \frac{1{,}000 \cdot (2.5 - V_{RTD})}{V_{RTD}}. \tag{4.11}$$

where Ω_{fixed} = 1,000 Ohm and V_{ref} = 2.5 V are the fixed reference resistance and voltage of the LJTick-Resistance, respectively [75].

4.4 POWER DISTRIBUTION SYSTEM

In order for the experimental setup to be portable, all electronic components are powered by batteries or by a laptop (via USB). The components requiring power, along with their power needs (i.e., input voltage and current) are enumerated in Table 4.1.

Table 4.1. Power requirements of electronic components.

Component	Manufacturer Product Name	Power Supply	Input Voltage	Input Current
Ion Pump Controller	Gamma Vacuum SPC	12 V 12 Ah Battery (x2)	24 V	15 – 30 mA[1]
Optical Fiber Interferometer Power Supply	Koheron SPS100	12 V 12 Ah Battery (x2)	24 V	2 A[2]
DAQ Device	LabJack U6	Laptop (USB)	-	-
Oscilloscope	Digilent Analog Discovery 2	Laptop (USB)	-	-

1. Dependant on the pressure inside the vacuum chamber.
2. Input current limit. In reality, the input current is in the mA range.

A power distribution enclosure was designed to distribute the required voltage from batteries to the ion pump controller and the optical fiber interferometer power supply (see Figure 4.11). There are three electronic components included in the circuit to ensure the safety of the equipment and users: (i) a diode, (ii) switches, and (iii) fuses. A diode only allows current to flow from the anode to the cathode, so it ensures the correct polarity of the battery connections to the enclosure. Switches are used to allow current flow to the electronic components only when needed. Fuses ensure the appropriate amount of current reaches the electronic components. If too much current is supplied by the batteries, the fuses will break the circuit to ensure the safety of the equipment.

The power source selected for the ion pump controller and the optical fiber interferometer are two 12 V 12 Ah batteries (BB Battery BP12-12) connected in series to fulfill the 24 V input voltage requirement. Since the maximum current draw is in the 2 A range, the 12 Ah batteries provide power for at least six hours (which is plenty of time to conduct radiative cooling experiments). In turn, the oscilloscope and DAQ device are both powered by a laptop via USB.

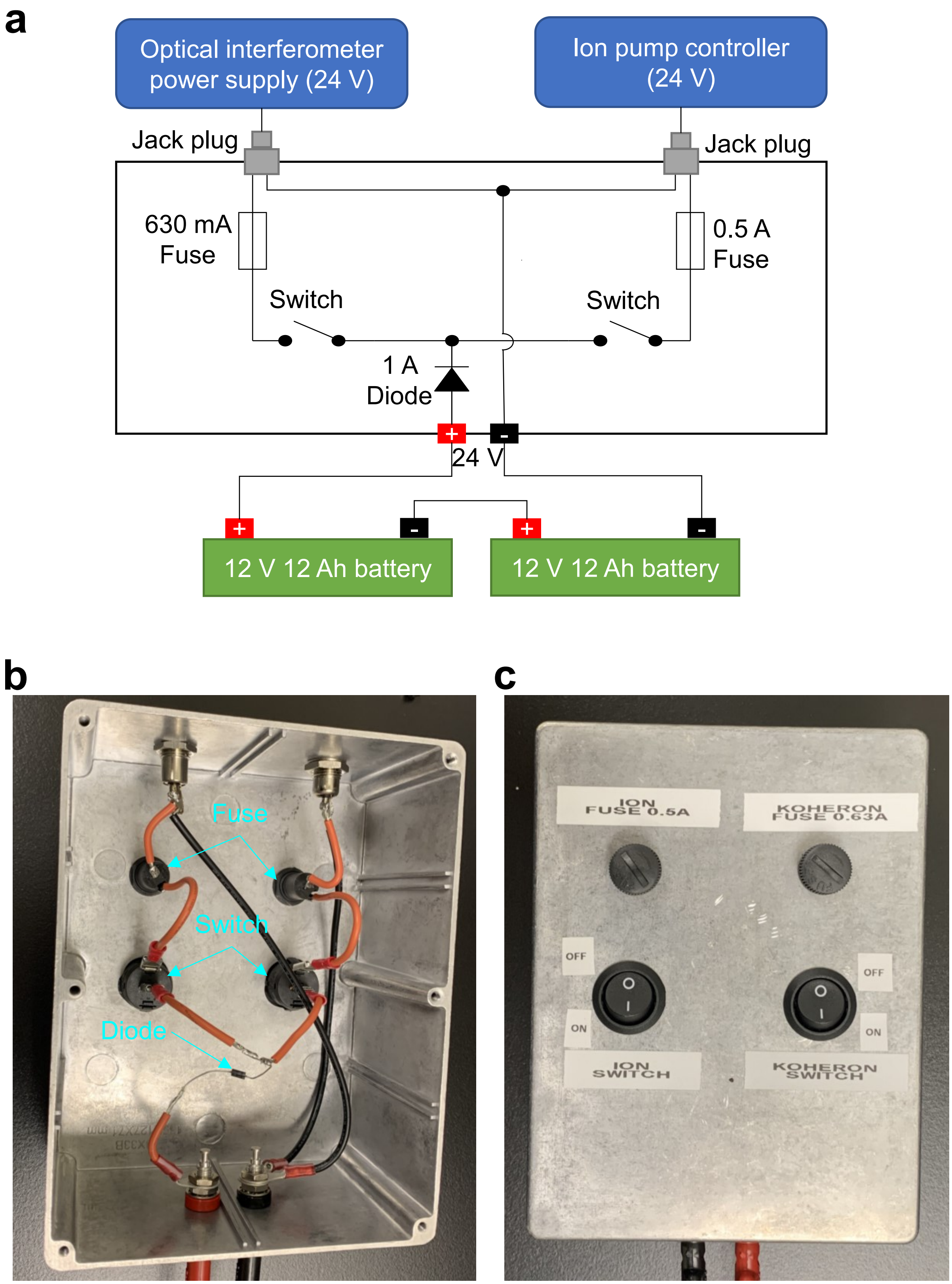


Figure 4.11. Power distribution enclosure a) schematic, b) interior, c) exterior.

4.5 EXPERIMENT PROCEDURE

The following procedure is applied for each radiative cooling experiment:

1. Bring the experimental setup (shown in Figure 4.2) outside with a clear view of the sky and begin continuously recording T_{ch}, T_{atm}, and T_{Si}. See Figure 5.1 for examples of experiment locations.
2. Wait for T_{ch}, T_{atm}, and T_{Si} to stabilize (typically 15 minutes, depending on outdoor temperature and sunlight exposure).
3. Cover the ZnSe viewport with aluminum foil to ensure that the membrane is coupled to the chamber only (i.e., no radiative cooling occurs).
4. Excite the membrane by light mechanical impact to the outer wall of the nipple (to make the membrane vibrate) and begin continuously tracking the membrane resonance frequency (f_m). See Appendix D for the procedure to log the resonance frequency with the Digilent Waveforms software.
    - Note that steps 3 and 4 typically only take a few seconds.
5. Uncover the ZnSe viewport to expose the membrane to the sky—for radiative cooling to occur—and measure the membrane resonance frequency shift (Δf_m).
6. Calculate the temperature drop of the membrane ($\overline{\Delta T_m}$) from the recorded resonance frequency shift using eq. (4.4).

The Python script used to process experimental data (T_{ch}, T_{atm}, T_{Si}, f_m) from the DAQ device (LabJack) and oscilloscope (Waveforms) data files is shown in Appendix K.

5 RESULTS & DISCUSSION

The temperature drop of a freestanding SiN membrane in vacuum upon radiative cooling is assessed for various experimental conditions. Experiments (20 in total) were conducted on 15 different days (during fall and summer months) in three different locations: (i) at the front entrance of the Advanced Research Complex at the University of Ottawa (uOttawa), (ii) in an open field in the rural region of Vankleek Hill, ON, Canada, and (iii) in an open field in the rural region of Fournier, ON, Canada. Figure 5.1 displays pictures of the experimental setup in each location. See Appendix G for the experimental setup transportation method to the rural locations. All experiment conditions—i.e., date, time, location, cloudiness, relative humidity, vacuum chamber temperature (T_{ch}), ambient atmospheric temperature (T_{atm}), viewport field of view angle (θ_v), and angle at which the sun is facing the membrane (θ_{sun})—and experimental results for the membrane average temperature shift ($\overline{\Delta T_m}$) are outlined in Table 5.1. The cloudiness level is determined visually by observing the sky directly above the setup (see Figure 5.2 for pictures of the sky during each experiment). The relative humidity is taken from Environment Canada's historical database for the weather and climate in Ottawa, ON (at the Ottawa International Airport station) [76]. Finally, T_{ch} and T_{atm} are determined as described in section 4.3, θ_{sun} is determined as described in section 3.2.4, and θ_v is determined as described in Appendix H.

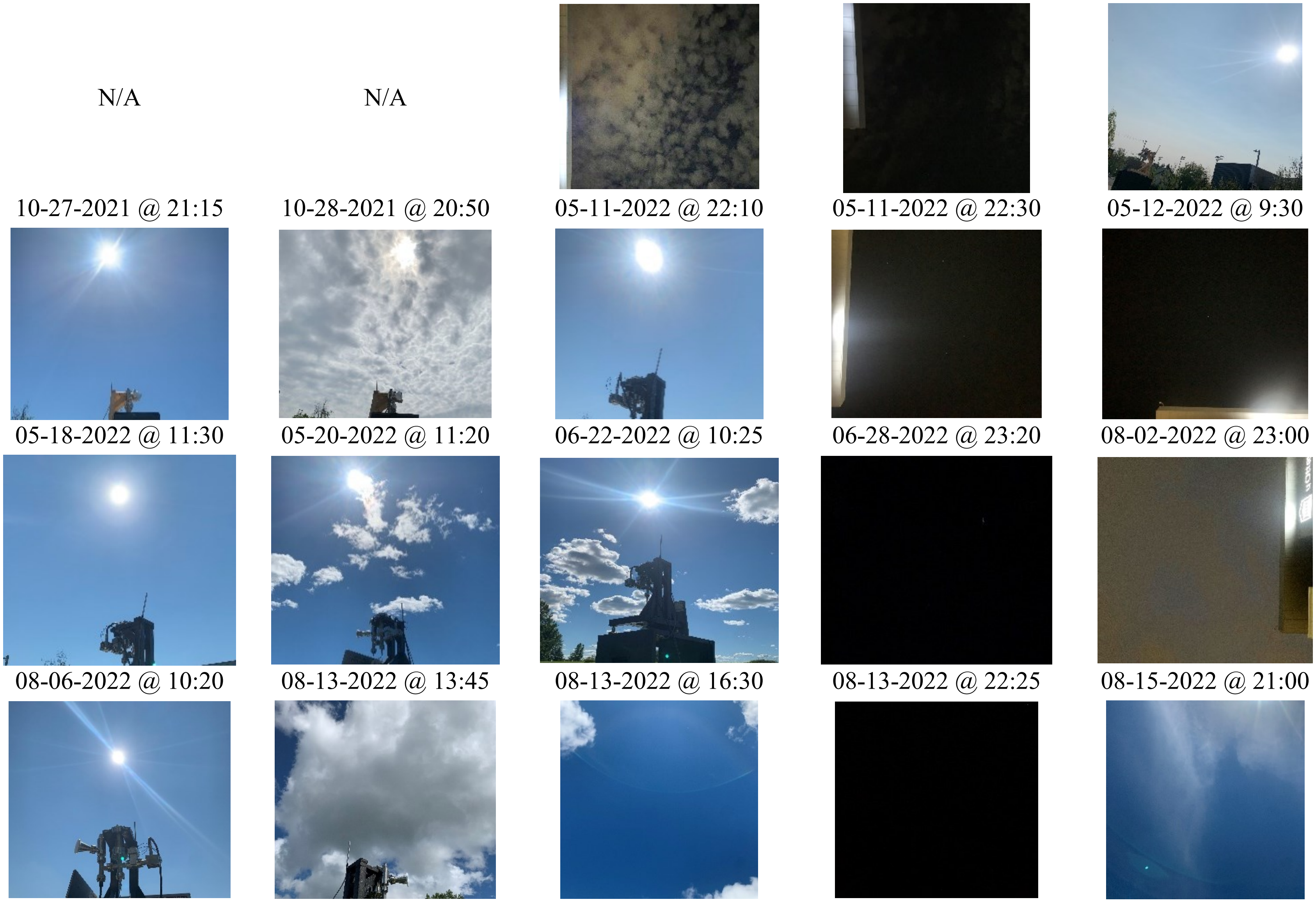
uOttawa location Rural locations

Figure 5.1. Pictures of the experiment locations. Left image: front of the Advanced Research Complex of the University of Ottawa, ON, Canada. Right top image: open field in the rural region of Fournier, ON, Canada. Right bottom image: open field in the rural region of Vankleek Hill, ON, Canada.

Table 5.1. Radiative cooling experiment conditions and experimental results.

Test	Date	Time	Location	Atmospheric Conditions		T_{atm} (K)	T_{ch} (K)[1]	θ_v (rad)	θ_{sun} (rad)	$\overline{\Delta T_m}$ (K)
				Cloudiness	Relative Humidity (%)					
1	10-27-2021	21:15	uOttawa	Cloudy	72	284	N/A	0.45	Night	-2.5 ± 0.8
2	10-28-2021	20:50	uOttawa	Clear	79	282	N/A	0.45	Night	-4.2 ± 1.3
3	05-11-2022	22:10	uOttawa	Cloudy	41	298	298	0.68	Night	-3.7 ± 1.2
4	05-11-2022	22:30	uOttawa	Clear	41	298	298	0.68	Night	-5.5 ± 1.7
5	05-12-2022	9:30	uOttawa	Clear	35	298	297	0.68	0.72	-4.7 ± 1.5
6	05-18-2022	11:30	uOttawa	Clear	41	294	290	0.68	0.46	-3.9 ± 1.2
7	05-20-2022	11:20	uOttawa	Cloudy	65	293	295	0.68	0.45	-1.8 ± 0.6
8	06-22-2022	10:25	uOttawa	Clear	75	300	N/A	0.68	0.54	-5.1 ± 1.6
9	06-28-2022	23:20	uOttawa	Clear	77	295	N/A	0.68	Night	-5.0 ± 1.6
10	08-02-2022	23:00	uOttawa	Clear	77	291	N/A	0.68	Night	-6.5 ± 2.1
11	08-06-2022	10:20	uOttawa	Clear	73	303	300	0.68	0.60	-4.6 ± 1.5
12	08-13-2022	13:45	Rural-FNR	Clear	43	300	300	0.68	0.66	-9.0 ± 2.8
13	08-13-2022	16:30	Rural-VKH	Clear	38	302	303	0.68	1.1	-9.3 ± 2.9
14	08-13-2022	22:25	Rural-VKH	Clear	63	289	292	0.68	Night	-7.1 ± 2.2
15	08-15-2022	21:00	uOttawa	Cloudy	63	298	298	0.68	Night	-2.5 ± 0.8
16	08-24-2022	11:20	uOttawa	Clear	74	301	N/A	0.68	0.62	-4.0 ± 1.3
17	08-27-2022	13:00	Rural-VKH	Cloudy	53	297	295	0.68	0.66	-4.3 ± 1.4
18	08-27-2022	13:20	Rural-VKH	Cloudy	53	297	298	0.68	0.66	-6.5 ± 2.1
19	08-29-2022	00:10	Rural-VKH	Clear	90	289	291	0.68	Night	-6.9 ± 2.2
20	08-29-2022	12:40	Rural-VKH	Clear	53	300	N/A	0.68	0.63	-7.6 ± 2.4

1. N/A means no thermistor measurements were taken.



N/A N/A

10-27-2021 @ 21:15 10-28-2021 @ 20:50 05-11-2022 @ 22:10 05-11-2022 @ 22:30 05-12-2022 @ 9:30

05-18-2022 @ 11:30 05-20-2022 @ 11:20 06-22-2022 @ 10:25 06-28-2022 @ 23:20 08-02-2022 @ 23:00

08-06-2022 @ 10:20 08-13-2022 @ 13:45 08-13-2022 @ 16:30 08-13-2022 @ 22:25 08-15-2022 @ 21:00

08-24-2022 @ 11:20 08-27-2022 @ 13:00 08-27-2022 @ 13:20 08-29-2022 @ 00:10 08-29-2022 @ 12:40

Figure 5.2. Pictures of the sky during each experiment. Some images have been adjusted for better contrast and brightness.

Figure 5.3a presents an experiment (performed on 08-27-2022 at 13:20, under direct sunlight) confirming that a temperature gradient on-chip between the SiN membrane and the silicon frame is created via radiative cooling. The reduction in temperature of the SiN membrane stabilizes approximately one second after uncovering the viewport, while the silicon frame remains approximately at constant temperature. The silicon frame is eventually heated by exposure to sunlight, but this occurs on a much larger time scale (see Figure 5.3b). It is possible that heating of the silicon frame can enhance the magnitude of the thermal gradient on-chip since thermal coupling of the membrane is radiation-dominated (i.e., it is weakly affected by heat conduction from the frame). However, this idea is not explored further, and this thesis limits experiments to short time scales during which the silicon frame temperature remains constant.

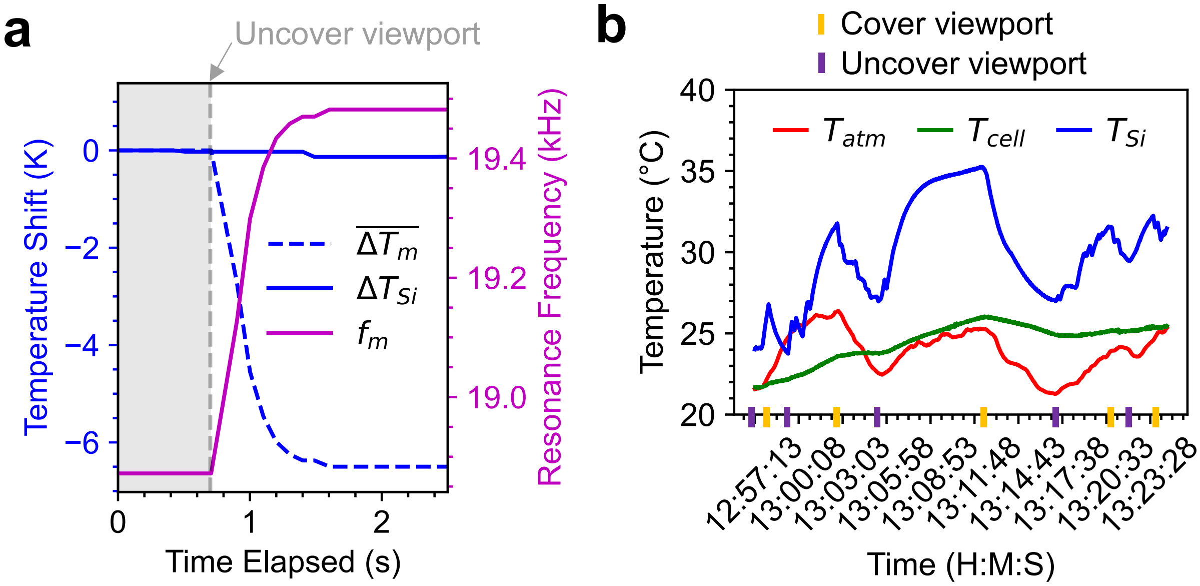


Figure 5.3. Results for an experiment conducted on 08-27-2022, under direct sunlight. a) Resonance frequency (f_m) and temperature shift (ΔT_m) of the SiN membrane upon radiative cooling while the silicon frame temperature shift (ΔT_{Si}) is minimal (at 13:20), and b) Evolution of ambient (T_{atm}), chamber (T_{ch}), and silicon frame (T_{Si}) temperature over a 25-minute period.

Successful radiative cooling of the membrane is achieved in all experiments (20 in total), and results show that atmospheric conditions (i.e., clouds, humidity) are the main factors responsible for the variability of the cooling, while solar irradiation is a weak factor. Figure 5.4 compiles all experimental results compared with the corresponding theoretical results (for the respective T_{ch}, T_{atm}, θ_v, and θ_{sun}). The heat transfer model bounds account for the two membrane emissivity models and for the variability of the atmospheric transparency. The weak cooling bound is calculated using the weaker thin film model emissivity (see Figure 3.2) and the weaker mid-latitude summer atmospheric transparency (see Figure 3.3), and the strong cooling bound is calculated using the stronger multilayer model emissivity (see Figure 3.2) and the stronger sub-arctic winter atmospheric transparency (see Figure 3.3). Notably, the model accounts only for clear sky and clear field of view conditions, which explains why some cloudy experiments show weaker cooling than expected by the model (e.g., 05-20-2022 and 08-15-2022). It is evident that atmospheric conditions (e.g., cloudiness, humidity) play a significant role in the magnitude of the membrane temperature drop. For example, two experiments were conducted 20 minutes apart on 05-11-2022 (at 22:10 and 22:30) while a cloud was passing over the setup (see pictures of the sky in Figure 5.2). This resulted in a 1.8 K (roughly 50%) increase in temperature drop from cloudy (3.7 ± 1.2 K drop) to clear sky (5.5 ± 1.7 K drop) conditions. The same effect occurred on 08-27-2022 between 13:00 and 13:20, where a 2.2 K (roughly 50%) increase in temperature drop was observed from cloudy (4.3 ± 1.4 K drop) to clear sky (6.5 ± 2.1 K drop) conditions. Cooling results are generally better with lower humidity levels, which makes sense because atmospheric transparency scales inversely with humidity. Moreover, it is evident that the urban landscape at the uOttawa location affects the cooling of the membrane because the rural experiments—with open field of view conditions—obtained significantly better results. This is most likely due to field of view obstruction as well as parasitic heating from the buildings and streetlights at the uOttawa location.

The maximal temperature drops were observed at the rural locations on 08-13-2022 with clear sky conditions. Three different experiments were conducted on this day and resulted in the record daytime cooling (9.3 ± 2.9 K at 16:30 with indirect sunlight, i.e., $\theta_{sun} > \theta_v$) and record nighttime cooling (7.1 ± 2.2 K at 22:25). A daytime experiment with direct sunlight was also conducted on this day (9.0 ± 2.8 K at 13:45) and confirms the weak influence of sunlight on the membrane. This makes sense because the absorptivity of the membrane in the solar spectrum is effectively null. However, the relative humidity was significantly higher during the nighttime experiment than during the daytime experiments on 08-13-2022. This means atmospheric transparency was most likely weaker, which would explain the weaker temperature drop during the nighttime experiment.

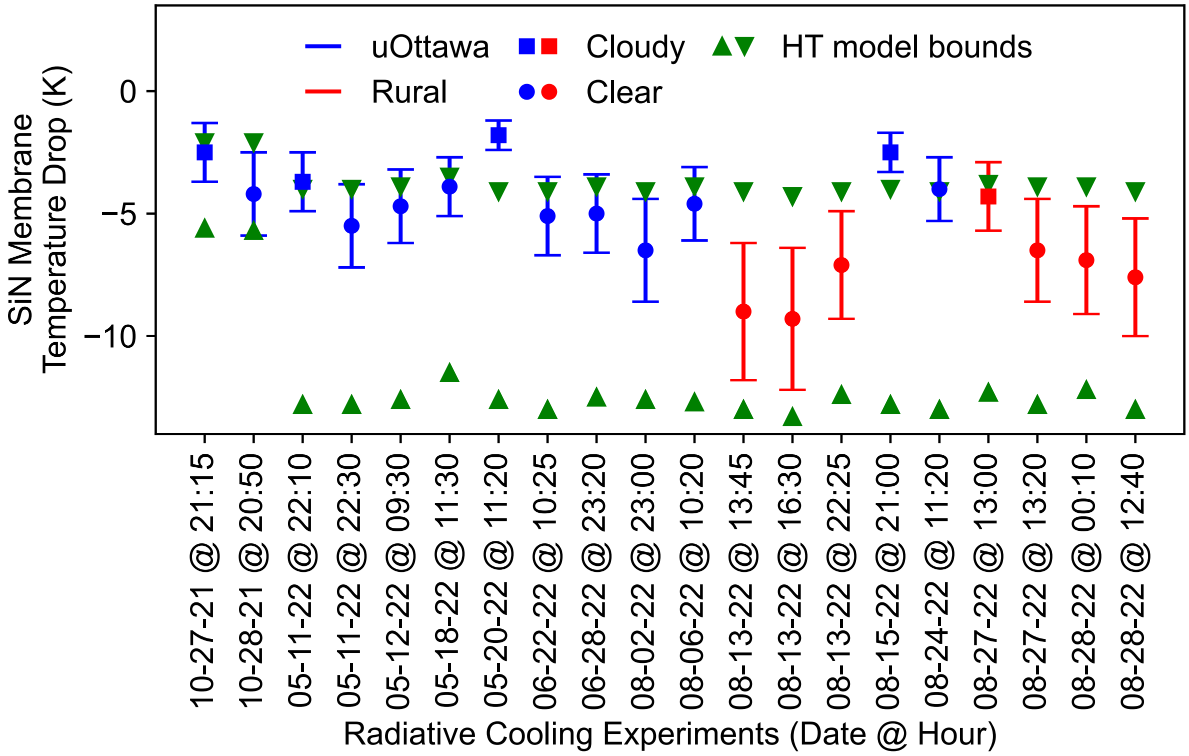


Figure 5.4. Membrane experimental temperature drop vs. theoretical temperature drop range for all experiments. Experimental uncertainties are calculated using the propagation of uncertainty rule (see section 4.1.2). The weak heat transfer cooling bound is calculated using the thin film model emissivity and the mid-latitude summer atmospheric transparency, and the strong heat transfer model cooling bound is calculated using the multilayer model emissivity and the sub-arctic winter atmospheric transparency.

6 FUTURE WORK & CONCLUDING REMARKS

6.1 POTENTIAL SETUP IMPROVEMENTS

The parasitic heating sources (q_{atm}, $q_{ch,top}$, $q_{ch,back}$, q_{cond}, q_{sun}) can be minimized, or even eliminated, by improving the experimental setup. Figure 6.1a shows the weight of each parasitic heating rate against q_m in the steady state energy balance considering the thin film model emissivity. Figure 6.1b shows the theoretical cooling potential of the membrane after successive improvements to the experimental setup, where the thin film model (with a blackbody backside) is used as the baseline.

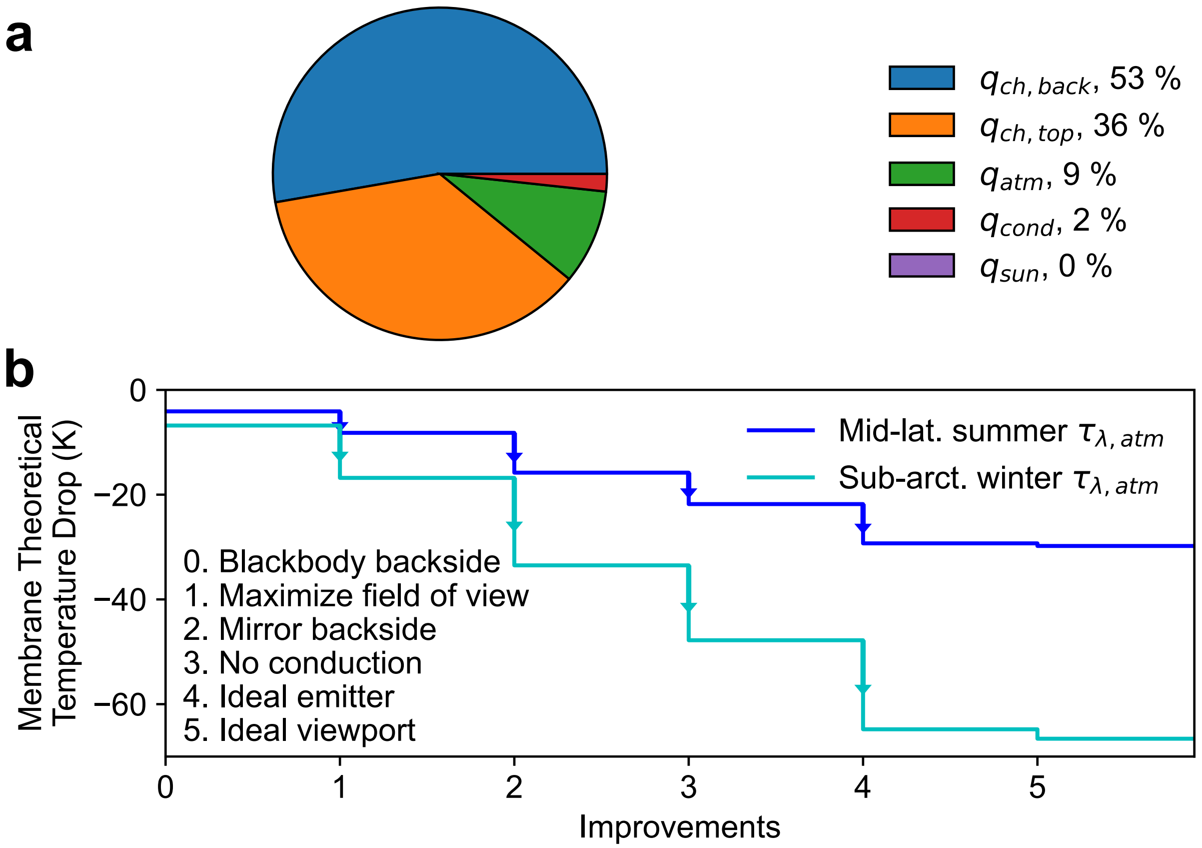


Figure 6.1. a) Theoretical weight of each parasitic heating rate against q_m in the steady state energy balance for the thin film model. b) Theoretical membrane temperature drops for successive improvements to the experimental setup (with constants $T_{ch} = T_{atm} = 300$ K and $\theta_{sun} = 0$).

The first improvement in Figure 6.1b is to maximize the field of view of the membrane (i.e., to θ_v = π/2). This would eliminate the chamber walls portion of $q_{ch,top}$, and it could be achieved by using a larger viewport. Larger ZnSe viewports are not currently available on the market, however a custom order for a larger viewport, as well as a custom viewport vacuum flange, may be possible. The second improvement is to maximize backside reflection by approaching a perfect mirror with the backside surrounding of the membrane. This would eliminate $q_{ch,back}$ and it could potentially be achieved through polishing of the optical fiber holder surface or adding a reflective layer (e.g., aluminum) under the SiN membrane. The third improvement is to eliminate conductive heating effects from the Si frame (q_{cond}). This could potentially be achieved by increasing the side length of the membrane (L), or by patterning the membrane in a trampoline shape [77], which would increase the fraction of heat transfer occurring via radiation versus conduction. The fourth, more hypothetical, improvement is to replace the SiN membrane with an ideal emitter—possessing an emissivity of unity in the 8 μm to 13 μm range and null elsewhere—to (i) maximize q_m in the atmospheric transparency window, and (ii) eliminate q_{sun} and q_{atm} outside of the 8 μm – 13 μm range. While achieving a perfect emitter is unlikely, emissivity engineering could potentially increase the emissivity peak of the SiN membrane in the 8 μm – 13 μm range (e.g., see section 6.2 for the theoretical potential of metasurfaces). The final improvement, also more hypothetical, is to replace the ZnSe viewport with an ideal viewport, which would be transparent in the 8 μm – 13 μm range to fully eliminate $q_{ch,top}$. The final improvement minimally improves cooling because the ZnSe viewport is almost ideal already. These modifications would leave q_{atm} (in the 8 μm – 13 μm range only) as the only remaining parasitic heating source. Figure 6.1b displays these improvements considering both mid-latitude summer and sub-arctic winter atmospheric transparency conditions. The maximal membrane temperature drop achieved with the current experimental setup is 9.3 K and, depending on atmospheric transparency conditions, it could potentially increase to 22 K – 48 K with the same membrane and the suggested setup improvements (i.e., improvements 1 to 3), and up to 30 K – 67 K with an ideal emitter and viewport.

6.2 METASURFACE DESIGN FOR RADIATIVE COOLING

Metasurfaces consist of thin materials deposited on the surface of photonic devices for the goal of manipulating spectral transmission and reflection properties. For this work, incorporating metasurface elements to the SiN membrane could potentially increase the emissivity peak of the membrane in the 8 μm – 13 μm range, thus increasing the radiative cooling power. A simple metasurface design—a titanium cross deposited on a SiN membrane—was investigated by Arthur Mardon (MITACS intern for summer 2022 with our group). Finite-Difference-Time-Domain (FDTD) simulations (with Ansys Lumerical) were conducted for various cross dimensions (e.g., thickness, height, arm width/length) to calculate the emissivity of the resulting metasurface. The parameters resulting in the best emissivity (i.e., highest peak in the 8 μm – 13 μm range) is shown in Figure 6.2a, and the corresponding emissivity spectrum is shown in Figure 6.2b (compared to the emissivity of a bare SiN thin film). Here, the spectral normal (not hemispherical) emissivity of the metasurface is shown because it is much less computationally expensive to obtain, compared to hemispherical emissivity, in Lumerical. However, the emissivity of a SiN membrane has a very weak angle dependency (as shown in Appendix I), which means the spectral normal emissivity is still a reliable indicator for the radiative cooling power of the metasurface.

The radiative cooling potential of the metasurface shown in Figure 6.2b can be calculated using the heat transfer model detailed in section 3. For $T_{atm} = T_{ch} = 300$ K, $\theta_v = 0.68$ rad, and mid-latitude summer atmospheric transparency, the theoretical temperature drop of the metasurface is 3.3 K, compared to 4.3 K for a bare SiN membrane (calculated with the thin film model). Even though the emissivity of the metasurface is higher in the 8 µm – 13 µm range, it still experiences less significant radiative cooling than bare SiN because of its additional parasitic heating outside of the 8 µm – 13 µm range. To enhance radiative cooling, the emissivity of the metasurface outside of the 8 µm – 13 µm range will need to be minimized, and this should be investigated as future work.

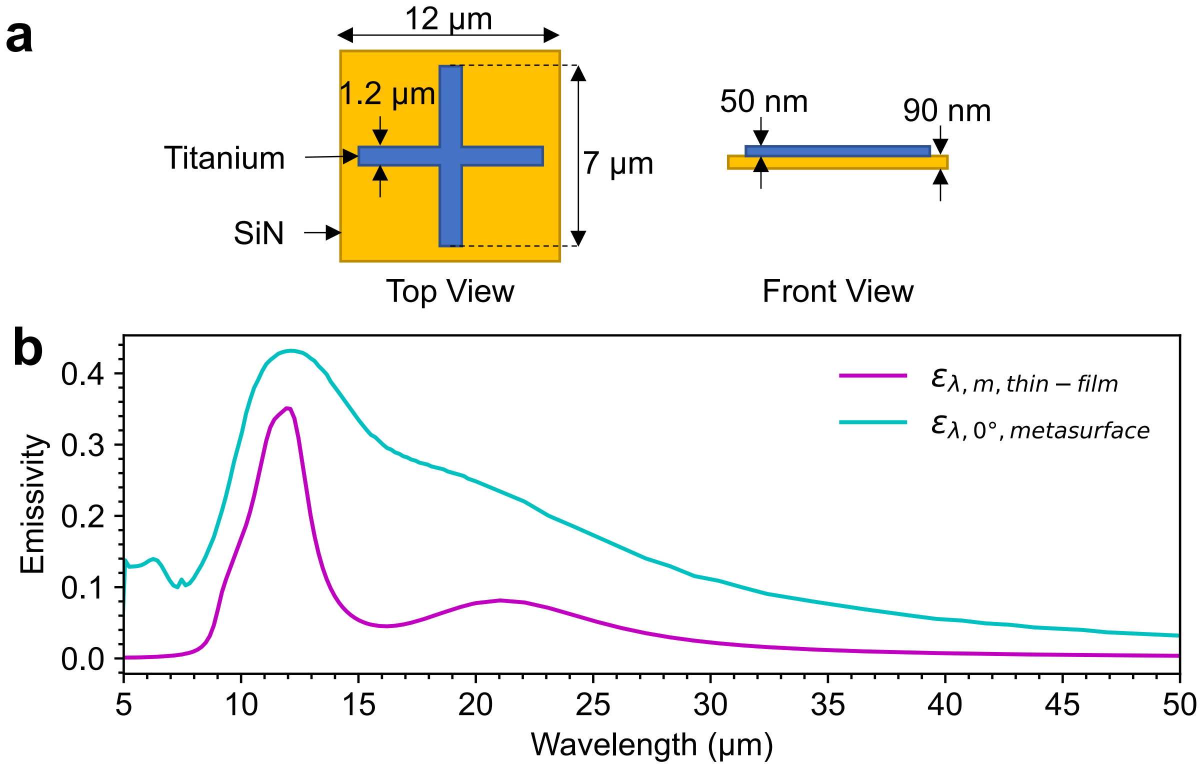


Figure 6.2. a) Depiction of a metasurface design consisting of a titanium cross deposited on a 90 nm thick SiN membrane. b) Spectral hemispherical emissivity (thin film model) of SiN thin film membrane compared to the spectral normal emissivity of the metasurface determined with FDTD simulations.

6.3 HEAT ENGINE ON-CHIP

Thermoelectric generators take advantage of the Seebeck effect to convert thermal gradients into electrical energy [78], [79]. In short, electrons in an n-type thermoelectric material move away from heat, generating a positive charge on the hot side. In a p-type thermoelectric material, holes experience the same effect, generating a negative charge on the hot side. This produces a voltage difference at the heated junction between n-type and p-type thermoelectric materials, which means electricity is generated (see Figure 6.3a).

Integrating a thermoelectric circuit on the SiN chip used in this work could potentially enable a micro heat engine by exploiting the on-chip thermal gradient created via radiative cooling. In this case, a planar thermoelectric circuit is necessary to harness the flow of heat between the radiatively cooled SiN membrane and the warmer silicon frame. A conceptual schematic of an integrated thermoelectric circuit on the SiN chip is shown in Figure 6.3b. It consists of metal contacts deposited on the SiN membrane and on the Si frame (which can be achieved by metal sputtering [65]), and thermoelectric wires deposited on the metal contacts to complete the thermoelectric circuit. Although this design is very simple, it allows for multiple thermoelectric generators to be connected in series to increase the overall voltage supply.

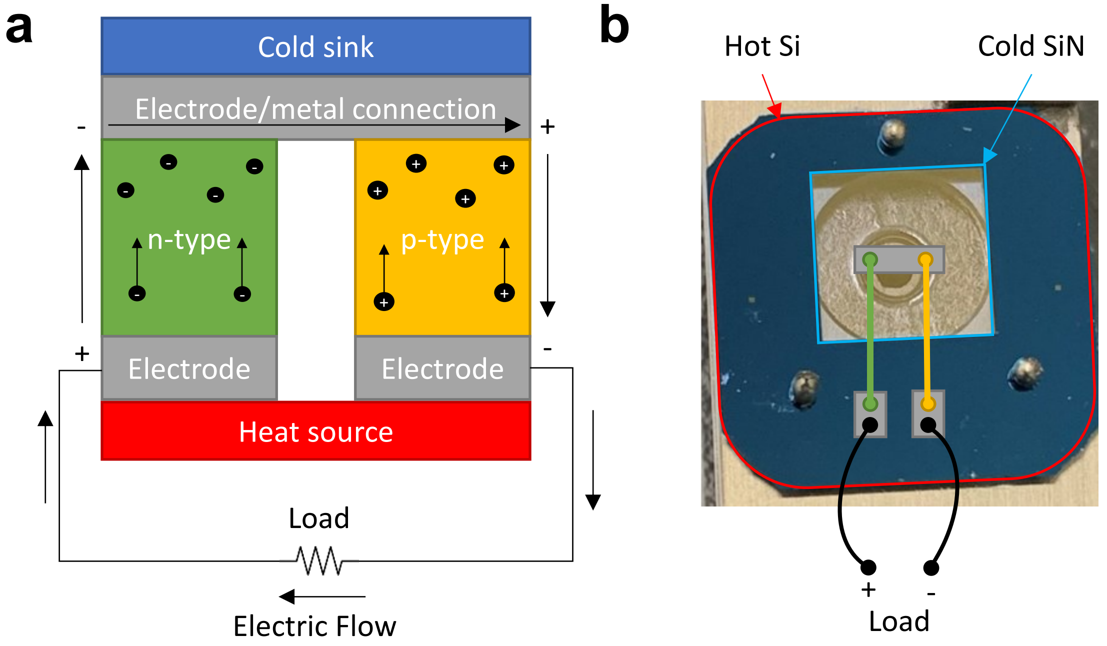


Figure 6.3. a) Seebeck effect schematic, and b) conceptual thermoelectric generator integrated on the SiN chip.

Planar on-chip thermoelectric circuits have been demonstrated [42], [80]. An interesting design integrated with a freestanding SiN membrane was demonstrated by Su et al. in 2018 [42] using silicon germanium (SiGe) thin films as the thermoelectric material (with p-type and n-type doping). Figure 6.4a outlines the fabrication process of the device, and Figure 6.4b shows the electric flow of the thermoelectric circuit on-chip. While this device is used as an on-chip microrefrigerator (i.e., voltage is supplied to the device to cool down the freestanding SiN), the same design can be used for the reverse application of harvesting electricity from a radiatively cooled SiN membrane.

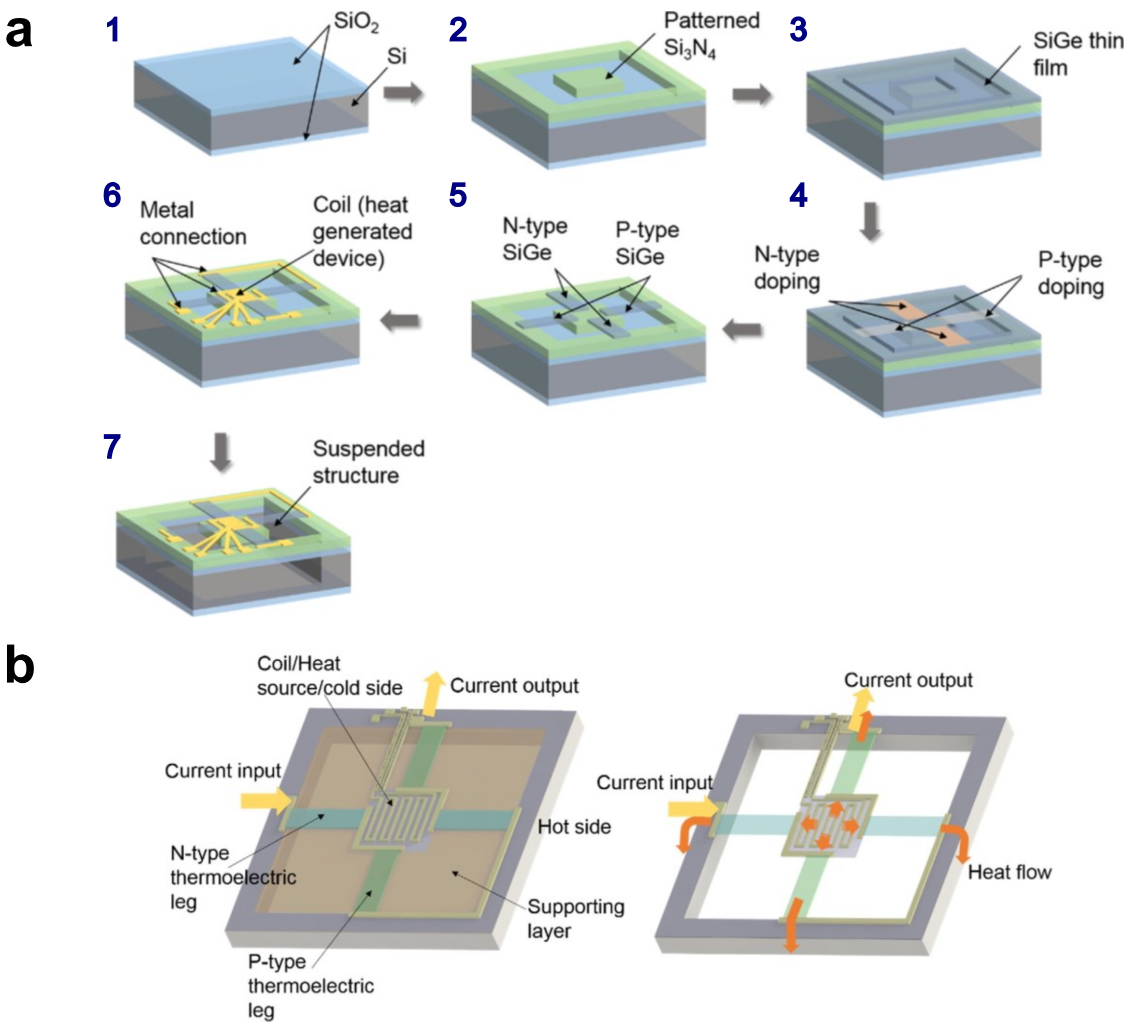


Figure 6.4. a) Fabrication process and b) energy flow schematic for conventional planar design (left) and freestanding planar design (right) for the thermoelectric microrefrigerator in [42].

The efficiency of a thermoelectric generator is dependant on the temperature-dependant figure of merit (ZT) of the thermoelectric material and is given by [81]:

$$\eta_{TEG} = \eta_{Carnot} \frac{\sqrt{1+ZT}-1}{\sqrt{1+ZT}+1-\eta_{Carnot}}, \tag{6.1}$$

where $\eta_{Carnot} = (T_{hot} - T_{cold}) \cdot T_{hot}^{-1}$ is the Carnot efficiency. The figure of merit of a thermoelectric material is defined as $ZT = S^2 T \kappa k^{-1}$, where S is the Seebeck coefficient (V·K^{-1}), κ is the electrical conductivity (S·m^{-1}), k is the thermal conductivity (W·m^{-1}·K^{-1}), and $T = 0.5 \cdot (T_{hot} + T_{cold})$ is the average value of the hot and cold temperatures (in K). Evidently, thermoelectric materials must possess a high Seebeck coefficient and electrical conductivity, while having weak thermal conductivity, to maximize ZT and thus the efficiency of the thermoelectric generator. The current state-of-the-art thermoelectric materials have ZT values of 1.38 (p-type BiSbTe [82]) and 1.04 (n-type CuBiTeSe [83]) at 300 K [81].

In this experiment, a theoretical cooling of 48 K is possible with improvements to the experimental setup, which corresponds to a Carnot efficiency of η_{Carnot} = 0.16 (at T_{hot} = T_{ch} = 300 K). Hence, with a thermal gradient on-chip of 48 K created via radiative cooling and with integrated state-of-the-art thermoelectric materials ($ZT \approx 1$), the on-chip heat-engine would theoretically have an efficiency of roughly η_{TEG} = 0.03. The upper limit power generation of the on-chip heat engine can not be greater than the emissive power of the membrane, which is in the 60 W·m^{-2}. Hence, the theoretical upper limit power generation of the heat engine (with efficiency η_{TEG} = 0.03) is roughly 1.8 W·m^{-2}, which corresponds to 65 µW for a 6 mm × 6 mm membrane. While this scale of power generation is significantly lower than typical photovoltaic solar panels (roughly 200 W·m^{-2} in good sunlight conditions), this heat engine would have the benefit of operating continuously throughout the day and night. In addition, the efficiency of the heat engine can be optimized by heating up the silicon frame with waste heat and solar irradiation to increase the thermal gradient on-chip.

There are a variety of microelectronics requiring small amounts of energy which could be fulfilled by a micro heat engine such as the one described in this section. For instance, integrated circuit timers require only 88 nW of power, and some analog clocks require as little as 390 nW [84]. In addition, self-powered small-scale electronics could be very useful in the aerospace industry, where space savings are crucial. The elevated altitude of aerospace applications would also enhance the effectiveness of radiative cooling since atmospheric transparency increases with altitude (i.e., atmospheric density decreases with altitude), and cloudiness may be completely avoided if a high enough altitude is reached.

6.4 RADIATIVE COOLING IN CAVITY OPTOMECHANICS

Freestanding SiN thin film membranes possess exceptionally low mechanical dissipation (i.e., high mechanical Quality factor), making them extremely popular in optomechanical sensing applications [37], [65], [67], [72], [85]–[94]. Interestingly, high-Quality factor SiN membranes are used in the research of cavity optomechanics, which studies the relation between electromagnetic radiation and mechanical motion at the micro/nano scale [95]. The field of cavity optomechanics is exploring the possibility of high-precision sensing (e.g., extremely small forces, masses, displacements) by damping and cooling of the mechanical oscillators through cavity cooling [86], [87], [96]. Radiative cooling could potentially be useful in these applications since it allows for coupling with outer space, instead of the ambient environment, as the thermal bath. This can be studied as future work.

6.5 CONCLUDING REMARKS

To conclude, the objective of this thesis was achieved by demonstrating a localized thermal gradient on-chip by radiative cooling of a silicon nitride nanomechanical resonator while its supporting silicon frame remains at constant temperature. This is, to the best of my knowledge, the first reported thermal gradient on-chip created via radiative cooling. Cooling of the SiN membrane was successfully demonstrated in various atmospheric conditions, and maximal temperature drops of 9.3 ± 2.9 K and 7.1 ± 2.2 K were demonstrated during the day and night, respectively. Improvements to the experimental setup could increase the temperature drop to, depending on atmospheric conditions, 22 K – 48 K for the same membrane, while emissivity engineering eventually yields a theoretical maximum cooling of 30 K – 67 K with an ideal emitter.

This work is a first step towards creating sustainable micro heat engines on-chip for the purpose of self-powering in small scale applications. However, there are limitations to such a device. First, its power generation potential is significantly lower than photovoltaic solar panels by a couple orders of magnitude. Second, atmospheric conditions play a significant role in the magnitude of the thermal gradient on-chip. This means the power generation potential of the heat engine can be substantially minimized, or even eliminated, in poor atmospheric conditions (e.g., heavy rain, high humidity, significant cloud cover), which also manifests in inconsistent power generation. Additionally, the experiments in this work are conducted at high vacuum using an energy consuming ion pump. To make this a completely passive process, other solutions that remove convective heating effects should be explored (e.g., vacuum sealed container, thermally insulated box). Nonetheless, such a micro heat engine has the advantage of working 24 hours per day in good atmospheric conditions, unlike solar panels which are limited to daytime only. Additionally, this work assesses the membrane temperature drop at short time scales only, but observations show that the silicon frame temperature can reach much higher temperatures than the ambient environment at longer time scales under direct sunlight. This means the on-chip heat engine could potentially be more efficient under direct sunlight since the membrane heat transfer is dominated by radiation exchange with the environment rather than conduction from the silicon frame. Longer time scale experiments should be investigated as future work to validate that a larger on-chip thermal gradient is indeed achieved over time under direct sunlight. Hence, this device could potentially allow continuous fulfillment of power needs in small scale applications when atmospheric conditions are not too harsh. An interesting idea would be to incorporate integrated on-chip photovoltaics [97] on such a device to increase the power generation during the day. With the advent of remote technological applications, such a sustainable micro heat engine could help alleviate the need for non-renewable sources of electricity and hazardous batteries.

REFERENCES

[1] A. Bouchard *et al.*, “Localized Thermal Gradients On-Chip by Radiative Cooling of Silicon Nitride Nanomechanical Resonators.” Sep. 29, 2022. doi: arXiv:2209.15004.

[2] X. Sun, Y. Sun, Z. Zhou, M. A. Alam, and P. Bermel, “Radiative sky cooling: fundamental physics, materials, structures, and applications,” *Nanophotonics*, vol. 6, no. 5, pp. 997–1015, Jul. 2017, doi: 10.1515/nanoph-2017-0020.

[3] M. N. Bahadori, “Passive Cooling Systems in Iranian Architecture,” *Scientific American*, vol. 238, no. 2, pp. 144–155, 1978.

[4] X. Lu, P. Xu, H. Wang, T. Yang, and J. Hou, “Cooling potential and applications prospects of passive radiative cooling in buildings: The current state-of-the-art,” *Renewable and Sustainable Energy Reviews*, vol. 65, pp. 1079–1097, Nov. 2016, doi: 10.1016/j.rser.2016.07.058.

[5] S. Zhang and J. Niu, “Cooling performance of nocturnal radiative cooling combined with microencapsulated phase change material (MPCM) slurry storage,” *Energy and Buildings*, vol. 54, pp. 122–130, Nov. 2012, doi: 10.1016/j.enbuild.2012.07.041.

[6] W. Wang, N. Fernandez, S. Katipamula, and K. Alvine, “Performance assessment of a photonic radiative cooling system for office buildings,” *Renewable Energy*, vol. 118, pp. 265–277, Apr. 2018, doi: 10.1016/j.renene.2017.10.062.

[7] F. Bu, D. Yan, G. Tan, H. Sun, and J. An, “Systematically incorporating spectrum-selective radiative cooling into building performance simulation: Numerical integration method and experimental validation,” *Applied Energy*, vol. 312, p. 118733, Apr. 2022, doi: 10.1016/j.apenergy.2022.118733.

[8] L. Zhu, A. Raman, K. X. Wang, M. A. Anoma, and S. Fan, “Radiative cooling of solar cells,” *Optica*, vol. 1, no. 1, p. 32, Jul. 2014, doi: 10.1364/OPTICA.1.000032.

[9] L. Zhu, A. P. Raman, and S. Fan, “Radiative cooling of solar absorbers using a visibly transparent photonic crystal thermal blackbody,” *Proc. Natl. Acad. Sci. U.S.A.*, vol. 112, no. 40, pp. 12282–12287, Oct. 2015, doi: 10.1073/pnas.1509453112.

[10] Z. Wang *et al.*, “Lightweight, Passive Radiative Cooling to Enhance Concentrating Photovoltaics,” *Joule*, vol. 4, no. 12, pp. 2702–2717, Dec. 2020, doi: 10.1016/j.joule.2020.10.004.

[11] B. Zhao *et al.*, “Radiative cooling of solar cells with micro-grating photonic cooler,” *Renewable Energy*, vol. 191, pp. 662–668, May 2022, doi: 10.1016/j.renene.2022.04.063.

[12] S. J. Byrnes, R. Blanchard, and F. Capasso, “Harvesting renewable energy from Earth’s mid-infrared emissions,” *Proceedings of the National Academy of Sciences*, vol. 111, no. 11, pp. 3927–3932, Mar. 2014, doi: 10.1073/pnas.1402036111.

[13] A. P. Raman, W. Li, and S. Fan, “Generating Light from Darkness,” *Joule*, vol. 3, no. 11, pp. 2679–2686, Nov. 2019, doi: 10.1016/j.joule.2019.08.009.

[14] S. Ishii, T. D. Dao, and T. Nagao, “Radiative cooling for continuous thermoelectric power generation in day and night,” *Appl. Phys. Lett.*, vol. 117, no. 1, p. 013901, Jul. 2020, doi: 10.1063/5.0010190.

[15] L. Yu, Z. Xi, S. Li, D. Pang, H. Yan, and M. Chen, “All-day continuous electrical power generator by solar heating and radiative cooling from the sky,” *Applied Energy*, vol. 322, p. 119403, Sep. 2022, doi: 10.1016/j.apenergy.2022.119403.

[16] S. Catalanotti, V. Cuomo, G. Piro, D. Ruggi, V. Silvestrini, and G. Troise, “The radiative cooling of selective surfaces,” *Solar Energy*, vol. 17, no. 2, pp. 83–89, May 1975, doi: 10.1016/0038-092X(75)90062-6.

[17] C. G. Granqvist and A. Hjortsberg, “Radiative cooling to low temperatures: General considerations and application to selectively emitting SiO films,” *Journal of Applied Physics*, vol. 52, no. 6, pp. 4205–4220, Jun. 1981, doi: 10.1063/1.329270.

[18] C. G. Granqvist, A. Hjortsberg, and T. S. Eriksson, “Radiative cooling to low temperatures with selectivity IR-emitting surfaces,” *Thin Solid Films*, vol. 90, no. 2, pp. 187–190, 1982.

[19] T. S. Eriksson, E. M. Lushiku, and C. G. Granqvist, “Materials for radiative cooling to low temperature,” *Solar Energy Materials*, vol. 11, no. 3, pp. 149–161, Nov. 1984, doi: 10.1016/0165-1633(84)90067-4.

[20] A. Addeo, L. Nicolais, G. Romeo, B. Bartoli, B. Coluzzi, and V. Silvestrini, “Light selective structures for large scale natural air conditioning,” *Solar Energy*, vol. 24, no. 1, pp. 93–98, 1980, doi: 10.1016/0038-092X(80)90024-9.

[21] P. Berdahl, M. Martin, and F. Sakkal, “Thermal performance of radiative cooling panels,” *International Journal of Heat and Mass Transfer*, vol. 26, no. 6, pp. 871–880, Jun. 1983, doi: 10.1016/S0017-9310(83)80111-2.

[22] A. W. Harrison and M. R. Walton, “Radiative cooling of TiO2 white paint,” *Solar Energy*, vol. 20, no. 2, pp. 185–188, 1978, doi: 10.1016/0038-092X(78)90195-0.

[23] E. Rephaeli, A. Raman, and S. Fan, “Ultrabroadband Photonic Structures To Achieve High-Performance Daytime Radiative Cooling,” *Nano Lett.*, vol. 13, no. 4, pp. 1457–1461, Apr. 2013, doi: 10.1021/nl4004283.

[24] A. P. Raman, M. A. Anoma, L. Zhu, E. Rephaeli, and S. Fan, “Passive radiative cooling below ambient air temperature under direct sunlight,” *Nature*, vol. 515, no. 7528, pp. 540–544, Nov. 2014, doi: 10.1038/nature13883.

[25] A. R. Gentle and G. B. Smith, “A Subambient Open Roof Surface under the Mid-Summer Sun,” *Advanced Science*, vol. 2, no. 9, p. 1500119, 2015, doi: 10.1002/advs.201500119.

[26] Z. Chen, L. Zhu, A. Raman, and S. Fan, “Radiative cooling to deep sub-freezing temperatures through a 24-h day–night cycle,” *Nat Commun*, vol. 7, no. 1, p. 13729, Dec. 2016, doi: 10.1038/ncomms13729.

[27] M. A. Kecebas, M. P. Menguc, A. Kosar, and K. Sendur, “Passive radiative cooling design with broadband optical thin-film filters,” *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 198, pp. 179–186, Sep. 2017, doi: 10.1016/j.jqsrt.2017.03.046.

[28] Y. Huang, M. Pu, Z. Zhao, X. Li, X. Ma, and X. Luo, “Broadband metamaterial as an ‘invisible’ radiative cooling coat,” *Optics Communications*, vol. 407, pp. 204–207, Jan. 2018, doi: 10.1016/j.optcom.2017.09.036.

[29] Y. Zhu, Y.-H. Ye, D. Wang, and Y. Cao, “Quasi-periodic selective multilayer emitter for sub-ambient daytime radiative cooling,” *AIP Advances*, vol. 11, no. 2, p. 025109, Feb. 2021, doi: 10.1063/5.0035138.

[30] Y. Lu *et al.*, “A Universal Route to Realize Radiative Cooling and Light Management in Photovoltaic Modules,” *Solar RRL*, vol. 1, no. 10, p. 1700084, 2017, doi: 10.1002/solr.201700084.

[31] B. Zhao, M. Hu, X. Ao, N. Chen, and G. Pei, “Radiative cooling: A review of fundamentals, materials, applications, and prospects,” *Applied Energy*, vol. 236, pp. 489–513, Feb. 2019, doi: 10.1016/j.apenergy.2018.12.018.

[32] M. M. Hossain, B. Jia, and M. Gu, "A Metamaterial Emitter for Highly Efficient Radiative Cooling," *Advanced Optical Materials*, vol. 3, no. 8, pp. 1047–1051, 2015, doi: 10.1002/adom.201500119.

[33] J.-W. Cho, T.-I. Lee, D.-S. Kim, K.-H. Park, Y.-S. Kim, and S.-K. Kim, "Visible to near-infrared thermal radiation from nanostructured tungsten anntennas," *J. Opt.*, vol. 20, no. 9, p. 09LT01, Aug. 2018, doi: 10.1088/2040-8986/aad708.

[34] K. Sun *et al.*, "Metasurface Optical Solar Reflectors Using AZO Transparent Conducting Oxides for Radiative Cooling of Spacecraft," *ACS Photonics*, vol. 5, no. 2, pp. 495–501, Feb. 2018, doi: 10.1021/acsphotonics.7b00991.

[35] B. Ko, D. Lee, T. Badloe, and J. Rho, "Metamaterial-Based Radiative Cooling: Towards Energy-Free All-Day Cooling," *Energies*, vol. 12, no. 1, p. 89, Dec. 2018, doi: 10.3390/en12010089.

[36] T. Wang, Y. Wu, L. Shi, X. Hu, M. Chen, and L. Wu, "A structural polymer for highly efficient all-day passive radiative cooling," *Nat Commun*, vol. 12, no. 1, p. 365, Dec. 2021, doi: 10.1038/s41467-020-20646-7.

[37] C. Zhang, M. Giroux, T. A. Nour, and R. St-Gelais, "Radiative Heat Transfer in Freestanding Silicon Nitride Membranes," *Phys. Rev. Applied*, vol. 14, no. 2, p. 024072, Aug. 2020, doi: 10.1103/PhysRevApplied.14.024072.

[38] N. Snell, C. Zhang, G. Mu, A. Bouchard, and R. St-Gelais, "Heat Transport in Silicon Nitride Drum Resonators and its Influence on Thermal Fluctuation-Induced Frequency Noise," *Phys. Rev. Applied*, vol. 17, no. 4, p. 044019, Apr. 2022, doi: 10.1103/PhysRevApplied.17.044019.

[39] M. Piller *et al.*, "Thermal radiation dominated heat transfer in nanomechanical silicon nitride drum resonators," *Appl. Phys. Lett.*, vol. 117, no. 3, p. 034101, Jul. 2020, doi: 10.1063/5.0015166.

[40] M. Giroux, C. Zhang, N. Snell, G. Mu, M. Stephan, and R. St-Gelais, "High resolution measurement of near-field radiative heat transfer enabled by nanomechanical resonators," *Appl. Phys. Lett.*, vol. 119, no. 17, p. 173104, Oct. 2021, doi: 10.1063/5.0068700.

[41] L. Vicarelli, A. Tredicucci, and A. Pitanti, "Micromechanical Bolometers for Subterahertz Detection at Room Temperature," *ACS Photonics*, vol. 9, no. 2, pp. 360–367, Feb. 2022, doi: 10.1021/acsphotonics.1c01273.

[42] Y. Su, J. Lu, D. Villaroman, D. Li, and B. Huang, "Free-standing planar thermoelectric microrefrigerators based on nano-grained SiGe thin films for on-chip refrigeration," *Nano Energy*, vol. 48, pp. 202–210, Jun. 2018, doi: 10.1016/j.nanoen.2018.03.054.

[43] A. Berk, P. Conforti, R. Kennett, T. Perkins, F. Hawes, and J. van den Bosch, "MODTRAN6: a major upgrade of the MODTRAN radiative transfer code," *Proceedings of SPIE*, vol. 9088, p. 90880H, Jun. 2014, doi: 10.1117/12.2050433.

[44] Berk, Alexander, Conforti, Patrick, and Hawes, Fred, "An accelerated line-by-line option for MODTRAN combining on-the-fly generation of line center absorption with 0.1 cm-1 bins and pre-computed line tails," *Proceedings of SPIE*, vol. 9471, p. 947217, May 2015, doi: 10.1117/12.2177444.

[45] G. Mu *et al.*, "Observation of Silicon Nitride Nanomechanical Resonator Actuation Using Capacitive Substrate Excitation," p. 12, Dec. 2021, doi: 10.48550/arXiv.2112.09303.

[46] T. L. Bergman and F. P. Incropera, Eds., *Fundamentals of heat and mass transfer*, 7th ed. Hoboken, NJ: Wiley, 2011.

[47] M. Okada, M. Okada, and H. Kusaka, “Dependence of Atmospheric Cooling by Vegetation on Canopy Surface Area During Radiative Cooling at Night: Physical Model Evaluation Using a Polyethylene Chamber,” *J. Agric. Meteorol.*, vol. 72, no. 1, pp. 20–28, 2016, doi: 10.2480/agrmet.D-15-00015.

[48] N. N. Shi, C.-C. Tsai, F. Camino, G. D. Bernard, N. Yu, and R. Wehner, “Keeping cool: Enhanced optical reflection and radiative heat dissipation in Saharan silver ants,” *Science*, vol. 349, no. 6245, pp. 298–301, Jul. 2015, doi: 10.1126/science.aab3564.

[49] J. Li *et al.*, “Protecting ice from melting under sunlight via radiative cooling,” *Sci. Adv.*, vol. 8, no. 6, p. eabj9756, Feb. 2022, doi: 10.1126/sciadv.abj9756.

[50] W. Shockley and H. J. Queisser, “Detailed Balance Limit of Efficiency of p-n Junction Solar Cells,” *Journal of Applied Physics*, vol. 32, no. 3, pp. 510–519, Mar. 1961.

[51] H. A. Macleod, *Thin-film optical filters*, Fifth edition. Boca Raton: CRC Press, Taylor & Francis Group, 2018.

[52] G. Cataldo, J. A. Beall, H.-M. Cho, B. McAndrew, M. D. Niemack, and E. J. Wollack, “Infrared dielectric properties of low-stress silicon nitride,” *Opt. Lett.*, vol. 37, no. 20, p. 4200, Oct. 2012, doi: 10.1364/OL.37.004200.

[53] “Zinc Selenide (ZnSe) Windows.” https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=3981 (accessed Apr. 22, 2022).

[54] H. Bao, C. Yan, B. Wang, X. Fang, C. Y. Zhao, and X. Ruan, “Double-layer nanoparticle-based coatings for efficient terrestrial radiative cooling,” *Solar Energy Materials and Solar Cells*, vol. 168, pp. 78–84, Aug. 2017, doi: 10.1016/j.solmat.2017.04.020.

[55] W. Li, Y. Shi, K. Chen, L. Zhu, and S. Fan, “A Comprehensive Photonic Approach for Solar Cell Cooling,” *ACS Photonics*, vol. 4, no. 4, pp. 774–782, Apr. 2017, doi: 10.1021/acsphotonics.7b00089.

[56] M. Chen, D. Pang, X. Chen, H. Yan, and Y. Yang, “Passive daytime radiative cooling: Fundamentals, material designs, and applications,” *EcoMat*, vol. 4, no. 1, Jan. 2022, doi: 10.1002/eom2.12153.

[57] S. Fan and W. Li, “Photonics and thermodynamics concepts in radiative cooling,” *Nat. Photon.*, vol. 16, no. 3, pp. 182–190, Mar. 2022, doi: 10.1038/s41566-021-00921-9.

[58] A. W. Harrison, “Effect of atmospheric humidity on radiation cooling,” *Solar Energy*, vol. 26, no. 3, pp. 243–247, 1981, doi: 10.1016/0038-092X(81)90209-7.

[59] B. A. Kimball, S. B. Idso, and J. K. Aase, “A model of thermal radiation from partly cloudy and overcast skies,” *Water Resources Research*, vol. 18, no. 4, pp. 931–936, 1982, doi: 10.1029/WR018i004p00931.

[60] Md. M. Hossain and M. Gu, “Radiative Cooling: Principles, Progress, and Potentials,” *Adv. Sci.*, vol. 3, no. 7, p. 1500360, Jul. 2016, doi: 10.1002/advs.201500360.

[61] “Reference Air Mass 1.5 Spectra.” https://www.nrel.gov/grid/solar-resource/spectra-am1.5.html (accessed Apr. 22, 2022).

[62] J. A. Duffie and W. A. Beckman, *Solar Engineering of Thermal Processes*, 4th ed. John Wiley & Sons, 2013.

[63] D. Rugar, H. J. Mamin, and P. Guethner, “Improved fiber-optic interferometer for atomic force microscopy,” *Appl. Phys. Lett.*, vol. 55, no. 25, pp. 2588–2590, Dec. 1989, doi: 10.1063/1.101987.

[64] P.-L. Yu, T. P. Purdy, and C. A. Regal, "Control of Material Damping in High-Q Membrane Microresonators," *Phys. Rev. Lett.*, vol. 108, no. 8, p. 083603, Feb. 2012, doi: 10.1103/PhysRevLett.108.083603.

[65] R. St-Gelais, S. Bernard, C. Reinhardt, and J. C. Sankey, "Swept-Frequency Drumhead Optomechanical Resonators," *ACS Photonics*, vol. 6, no. 2, pp. 525–530, Feb. 2019, doi: 10.1021/acsphotonics.8b01519.

[66] A. Blaikie, D. Miller, and B. J. Alemán, "A fast and sensitive room-temperature graphene nanomechanical bolometer," *Nat Commun*, vol. 10, no. 1, p. 4726, Dec. 2019, doi: 10.1038/s41467-019-12562-2.

[67] M. Piller, N. Luhmann, M.-H. Chien, and S. Schmid, "Nanoelectromechanical infrared detector," in *Optical Sensing, Imaging, and Photon Counting: From X-Rays to THz 2019*, San Diego, United States, Sep. 2019, p. 1. doi: 10.1117/12.2528416.

[68] Y. Toivola, J. Thurn, R. F. Cook, G. Cibuzar, and K. Roberts, "Influence of deposition conditions on mechanical properties of low-pressure chemical vapor deposited low-stress silicon nitride films," *Journal of Applied Physics*, vol. 94, no. 10, pp. 6915–6922, Nov. 2003, doi: 10.1063/1.1622776.

[69] S. D. Senturia, *Microsystem Design*. Kluwer Academic, 2002.

[70] K. E. Petersen, "Silicon as a mechanical material," *Proceedings of the IEEE*, vol. 70, no. 5, pp. 420–457, May 1982, doi: 10.1109/PROC.1982.12331.

[71] G. Carlotti *et al.*, "Measurement of the elastic and viscoelastic properties of dielectric films used in microelectronics," *Thin Solid Films*, vol. 414, no. 1, pp. 99–104, Jul. 2002, doi: 10.1016/S0040-6090(02)00430-3.

[72] S.-E. Zhu, M. Krishna Ghatkesar, C. Zhang, and G. C. A. M. Janssen, "Graphene based piezoresistive pressure sensor," *Appl. Phys. Lett.*, vol. 102, no. 16, p. 161904, Apr. 2013, doi: 10.1063/1.4802799.

[73] H. Ftouni *et al.*, "Thermal conductivity of silicon nitride membranes is not sensitive to stress," *Phys. Rev. B*, vol. 92, no. 12, p. 125439, Sep. 2015, doi: 10.1103/PhysRevB.92.125439.

[74] "EI-1034 Datasheet," *LabJack.* https://labjack.com/pages/support?doc=/datasheets/accessories/ei-1034-datasheet/ (accessed Oct. 13, 2022).

[75] "LJTick-Resistance Datasheet," *LabJack.* https://labjack.com/en-ca/pages/support?doc=/datasheets/accessories/ljtick-resistance-datasheet/#:~:text=The%20LJTick-Resistance%20(LJTR)%20makes (accessed Oct. 13, 2022).

[76] E. and C. C. Canada, "Historical Data - Climate - Environment and Climate Change Canada," Oct. 31, 2011. https://climate.weather.gc.ca/historical_data/search_historic_data_e.html (accessed Sep. 19, 2022).

[77] C. Reinhardt, T. Müller, A. Bourassa, and J. C. Sankey, "Ultralow-Noise SiN Trampoline Resonators for Sensing and Optomechanics," *Phys. Rev. X*, vol. 6, no. 2, p. 021001, Apr. 2016, doi: 10.1103/PhysRevX.6.021001.

[78] H. J. Goldsmid, "The Physics of Thermoelectric Energy Conversion - Chapter 1 - The Seebeck and Peltier Effect," *The Physics of Thermoelectric Energy Conversion*, pp. 1–1 to 1–3, Apr. 2017, doi: 10.1088/978-1-6817-4641-8ch1.

[79] H. J. Goldsmid, “The Physics of Thermoelectric Energy Conversion - Chapter 2 - The Thermoelectric Figure of Merit,” *The Physics of Thermoelectric Energy Conversion*, pp. 2–1 to 2–8, Apr. 2017, doi: 10.1088/978-1-6817-4641-8ch2.

[80] S. Pelegrini *et al.*, “Development and characterization of a microthermoelectric generator with plated copper/constantan thermocouples,” *Microsyst Technol*, vol. 20, no. 4–5, pp. 585–592, Apr. 2014, doi: 10.1007/s00542-013-1993-7.

[81] Z. Ouyang and D. Li, “Modelling of segmented high-performance thermoelectric generators with effects of thermal radiation, electrical and thermal contact resistances,” *Sci Rep*, vol. 6, no. 1, p. 24123, Jul. 2016, doi: 10.1038/srep24123.

[82] W. Xie *et al.*, “Identifying the Specific Nanostructures Responsible for the High Thermoelectric Performance of (Bi,Sb) $_2$ Te $_3$ Nanocomposites,” *Nano Lett.*, vol. 10, no. 9, pp. 3283–3289, Sep. 2010, doi: 10.1021/nl100804a.

[83] W.-S. Liu *et al.*, “Thermoelectric Property Studies on Cu-Doped n-type CuxBi2Te2.7Se0.3 Nanocomposites,” *Advanced Energy Materials*, vol. 1, no. 4, pp. 577–587, 2011, doi: 10.1002/aenm.201100149.

[84] H. Toshiyoshi, S. Ju, H. Honma, C.-H. Ji, and H. Fujita, “MEMS vibrational energy harvesters,” *Science and Technology of Advanced Materials*, vol. 20, no. 1, pp. 124–143, Dec. 2019, doi: 10.1080/14686996.2019.1569828.

[85] B. M. Zwickl *et al.*, “High quality mechanical and optical properties of commercial silicon nitride membranes,” *Appl. Phys. Lett.*, vol. 92, no. 10, p. 103125, Mar. 2008, doi: 10.1063/1.2884191.

[86] D. J. Wilson, C. A. Regal, S. B. Papp, and H. J. Kimble, “Cavity Optomechanics with Stoichiometric SiN Films,” *Phys. Rev. Lett.*, vol. 103, no. 20, p. 207204, Nov. 2009, doi: 10.1103/PhysRevLett.103.207204.

[87] C. A. Regal and K. W. Lehnert, “From cavity electromechanics to cavity optomechanics,” *J. Phys.: Conf. Ser.*, vol. 264, p. 012025, Jan. 2011, doi: 10.1088/1742-6596/264/1/012025.

[88] M. S. Hanay *et al.*, “Single-protein nanomechanical mass spectrometry in real time,” *Nature Nanotech*, vol. 7, no. 9, pp. 602–608, Sep. 2012, doi: 10.1038/nnano.2012.119.

[89] T. Larsen, S. Schmid, L. G. Villanueva, and A. Boisen, “Photothermal Analysis of Individual Nanoparticulate Samples Using Micromechanical Resonators,” *ACS Nano*, vol. 7, no. 7, pp. 6188–6193, Jul. 2013, doi: 10.1021/nn402057f.

[90] X. C. Zhang, E. B. Myers, J. E. Sader, and M. L. Roukes, “Nanomechanical Torsional Resonators for Frequency-Shift Infrared Thermal Sensing,” *Nano Lett.*, vol. 13, no. 4, pp. 1528–1534, Apr. 2013, doi: 10.1021/nl304687p.

[91] G. Brucoli, P. Bouchon, R. Haïdar, M. Besbes, H. Benisty, and J.-J. Greffet, “High efficiency quasi-monochromatic infrared emitter,” *Appl. Phys. Lett.*, vol. 104, no. 8, p. 081101, Feb. 2014, doi: 10.1063/1.4866342.

[92] S. Chakram, Y. S. Patil, L. Chang, and M. Vengalattore, “Dissipation in Ultrahigh Quality Factor SiN Membrane Resonators,” *Phys. Rev. Lett.*, vol. 112, no. 12, p. 127201, Mar. 2014, doi: 10.1103/PhysRevLett.112.127201.

[93] Y. Zhang, Y. Watanabe, S. Hosono, N. Nagai, and K. Hirakawa, “Room temperature, very sensitive thermometer using a doubly clamped microelectromechanical beam resonator for bolometer applications,” *Appl. Phys. Lett.*, vol. 108, no. 16, p. 163503, Apr. 2016, doi: 10.1063/1.4947444.

[94] L. Laurent, J.-J. Yon, J.-S. Moulet, M. Roukes, and L. Duraffourg, “12 − μ m -Pitch Electromechanical Resonator for Thermal Sensing,” *Phys. Rev. Applied*, vol. 9, no. 2, p. 024016, Feb. 2018, doi: 10.1103/PhysRevApplied.9.024016.

[95] M. Aspelmeyer, T. J. Kippenberg, and F. Marquardt, “Cavity optomechanics,” *Rev. Mod. Phys.*, vol. 86, no. 4, pp. 1391–1452, Dec. 2014, doi: 10.1103/RevModPhys.86.1391.

[96] C. H. Metzger and K. Karrai, “Cavity cooling of a microlever,” *Nature*, vol. 432, no. 7020, pp. 1002–1005, Dec. 2004, doi: 10.1038/nature03118.

[97] C. Steffan, P. Greiner, B. Deutschmann, C. Kollegger, and G. Holweg, “Energy harvesting with on-chip solar cells and integrated DC/DC converter,” in *2015 45th European Solid State Device Research Conference (ESSDERC)*, Sep. 2015, pp. 142–145. doi: 10.1109/ESSDERC.2015.7324733.

[98] Z. M. Zhang, *Nano/Microscale Heat Transfer*. Cham: Springer International Publishing, 2020. doi: 10.1007/978-3-030-45039-7.

[99] G. S. Springer, “Heat Transfer in Rarefied Gases,” in *Advances in Heat Transfer*, vol. 7, Elsevier, 1971, pp. 163–218. doi: 10.1016/S0065-2717(08)70018-2.

[100] “PVWatts Calculator.” https://pvwatts.nrel.gov/ (accessed Apr. 22, 2022).

APPENDICES

A. CONVECTIVE EFFECTS AT HIGH VACUUM

The convection heat transfer between the membrane and ambient air is dependant on the gas dynamics inside the chamber, which can be characterized by the Knudsen number:

$$K_n = \frac{\Lambda}{l}, \tag{A1}$$

where Λ is the mean free path of the air molecules and l is the characteristic length (i.e., the gaseous gap distance over which thermal transport occurs) [98]. Here, there are two characteristic lengths: (i) the backside gap (i.e., the distance between the membrane and the membrane mount, $l \approx 1.4$ mm), and (ii) the topside gap (i.e., the distance between the membrane and the viewport, $l \approx 10$ mm). From the ideal gas law, the mean free path of air can be determined by:

$$\Lambda = \frac{k_B T}{\sqrt{2}\pi P D_m^2}, \tag{A2}$$

where P and T are the gas pressure and temperature, respectively, $k_B = 1.381 \times 10^{-23}$ J·K^{-1} is the Boltzmann constant, and $D_m \approx 0.3$ nm is the molecular diameter of air. For a chamber at $P \approx 10^{-6}$ hPa and $T \approx 300$ K, the mean free path is approximately 104 m, which corresponds to Knudsen numbers of 74,000 (backside) and 10,400 (topside). Hence, the air flow in the chamber may be defined by the "free molecule flow" regime because the mean free path of the air molecules is much larger than the characteristic lengths in the chamber (i.e., $K_n \gg 10$). The air conduction net heat flux between two plates (i.e., the membrane and the viewport or mount) in this regime is given by [98]:

$$q_{air,FM} = \frac{(T_m - T_c) \cdot [\mathrm{A}_T(\gamma + 1)c_v P]}{(2 - \mathrm{A}_T)\sqrt{8\pi R_{air} T_{m,FM}}}, \tag{A3}$$

where T_m and T_c are the temperatures of the membrane and chamber, respectively, A_T is the thermal accommodation coefficient, R_{air} = 287 J·kg^{-1}·K^{-1} is the gas constant of air, γ = 1.4 is the specific heat ratio for diatomic gases, c_v = 716.6 J·kg^{-1}·K^{-1} is the constant volume specific heat for air, and $T_{m,FM}$ is the effective mean temperature for a free molecule flow regime. The thermal accommodation coefficient (A_T) is dependant on various parameters (e.g., surface conditions, temperature) and has been measured experimentally for various conditions [99]. For a wide range of engineered surfaces, the thermal accommodation coefficient is 0.8 < A_T < 1. Here we take A_T = 0.9 (i.e., coefficient for air flow with glass). For a free molecule regime, $T_{m,FM}$ is given by:

$$T_{m,FM} = \frac{4T_m' T_c{}'}{\left(\sqrt{T_m'} + \sqrt{T_c{}'}\right)^2}, \tag{A4}$$

where,

$$T_m' = \frac{T_m + (1 - \mathrm{A}_T)T_c}{2 - \mathrm{A}_T}, \tag{A5}$$

$$T_c' = \frac{T_c + (1 - \mathrm{A}_T)T_m}{2 - \mathrm{A}_T}. \tag{A6}$$

By taking P = 10^{-6} hPa, T_c = 300 K, and T_m = 290 K (i.e., assuming a 10 K membrane temperature drop), the air conduction net heat flux between the ZnSe viewport and the membrane is calculated as $q_{air,FM}$ = 1.61 × 10^{-4} W·m^{-2}. Considering the membrane radiative cooling power is in the 40 W·m^{-2} range per side (calculated with eq. (3.9) using the thin film model emissivity), the air conduction only offsets around 0.0004% of the cooling power. Thus, the convective effects on the membrane in the chamber at high vacuum are negligeable.

B. OTTAWA SOLAR IRRADIANCE

The PVWatts calculator by the National Renewable Energy Laboratory [100] provides solar irradiation estimates for solar panels based on typical weather data from a multi-year historical period. Table A1 outlines the parameters used to determine the solar irradiance on the SiN membrane in this experiment.

Table A1. PVWatts calculator parameters

Specification	Input
Requested Location	Ottawa, ON, Canada
Latitude	45.41º North
Longitude	75.7º West
Surface Tilt	0º (horizontal)

The calculator outputs an estimate of the hourly direct and diffuse solar irradiation (in $W{\cdot}m^{-2}$) components separately. By taking the sum of these two components, the global solar irradiance (in $W{\cdot}m^{-2}$) can be determined. For the parameters of Table A1, the monthly maximum global solar irradiation is outlined in Table A2. As shown in Table A2, the peak conditions for Ottawa are in the 1000 $W{\cdot}m^{-2}$ range.

Table A2. Monthly maximum global solar irradiation on a horizontal surface in Ottawa

Month	Maximum Global Solar Irradiance (W·m^{-2})
January	1002
February	1096
March	1119
April	1104
May	1087
June	1056
July	1032
August	1026
September	1043
October	1014
November	979
December	949

C. EXAMPLE CALCULATION: ANGLE AT WHICH THE SUN IS FACING THE MEMBRANE

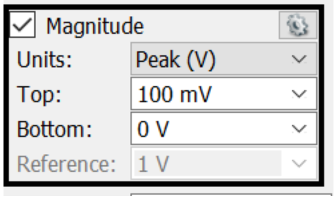
For an experiment conducted on 2022-05-11, in Ottawa, at 9:30, θ_{sun} is determined with the following steps:

1. Ottawa has a latitude of 45° N, so the latitude $\Phi = 45^{\text{o}}$.
2. 2022-05-11 is the 131st day of the year, so the day number $n = 131$.
3. From eq. (3.17) with $n = 131$, the declination angle $\delta = 17.8^{o}$.
4. 9:30 is 2.5 hours before noon, so the hour angle $\omega = -2.5\ hours \cdot \left(\frac{15^{o}}{hour}\right) = -37.5^{o}$
5. From eq. (3.16), the angle at which the sun is facing the membrane $\theta_{sun} = 41.4^{o}$ or 0.72 rad.

D. STEPS FOR LOGGING RESONANCE FREQUENCY WITH WAVEFORMS

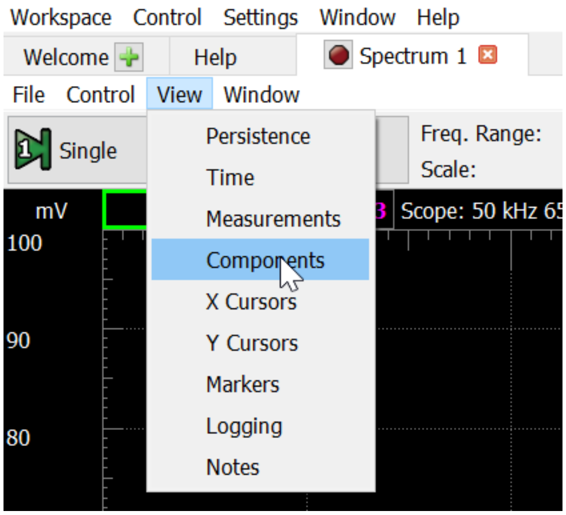
Once the experimental setup (i.e., the optical fiber interferometer and vacuum chamber) is built as described in section 4, the following steps are used to log the resonance frequency of the SiN membrane using Digilent Waveforms:

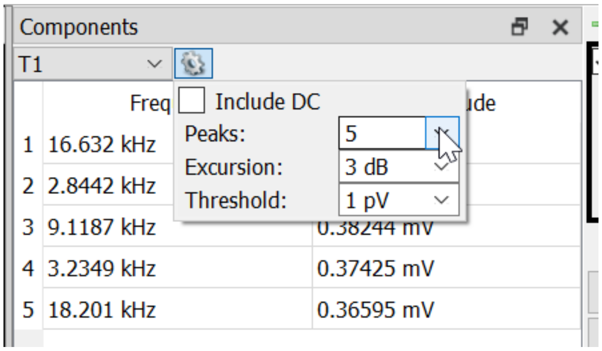
1. Open the “Spectrum Analyzer” function to obtain the FFT spectrum of the oscilloscope (Digilent Analog Discovery 2) signal (as shown in Figure 4.5).
2. Set the signal amplitude units to “Peak (V).”



☑ Magnitude
Units: Peak (V)
Top: 100 mV
Bottom: 0 V
Reference: 1 V

3. Open the “Components” tab (View → Components) and set the number of peaks to 5 (more or less peaks may be selected depending on how much noise data is needed).

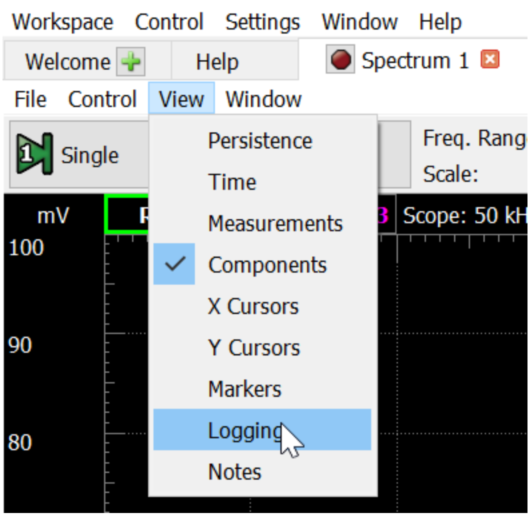


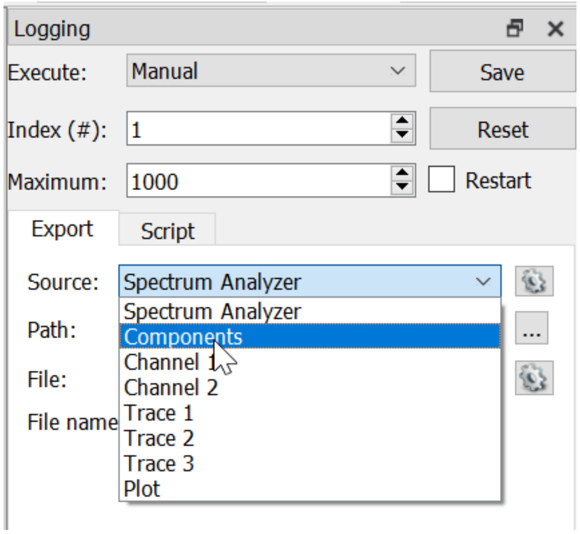


☐ Include DC
Peaks: 5
Excursion: 3 dB
Threshold: 1 pV

	Freq	ude
1	16.632 kHz	
2	2.8442 kHz	
3	9.1187 kHz	0.38244 mV
4	3.2349 kHz	0.37425 mV
5	18.201 kHz	0.36595 mV

4. Open the “Logging” tab (View → Logging) and select “Components” as the logging source.




Logging

Execute: Manual

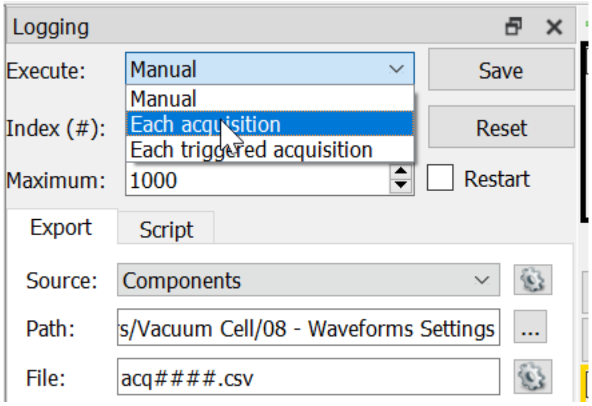
Index (#): 1

Maximum: 1000

☐ Restart

Source: Spectrum Analyzer

5. Identify the folder path and the file type (I typically use .txt) and execute the logging for "Each acquisition." The program will start logging the FFT frequency peaks in a .txt file approximately every 100 ms.


Logging

Execute: Manual

Maximum: 1000

☐ Restart

Source: Components

Path: s/Vacuum Cell/08 - Waveforms Settings

File: acq####.csv

6. To stop logging, change the execute variable to "Manual."

E. VACUUM CHAMBER PICTURES

Additional pictures of the vacuum chamber are shown in Figure A1. Refer to Figure 4.2 for component identification.

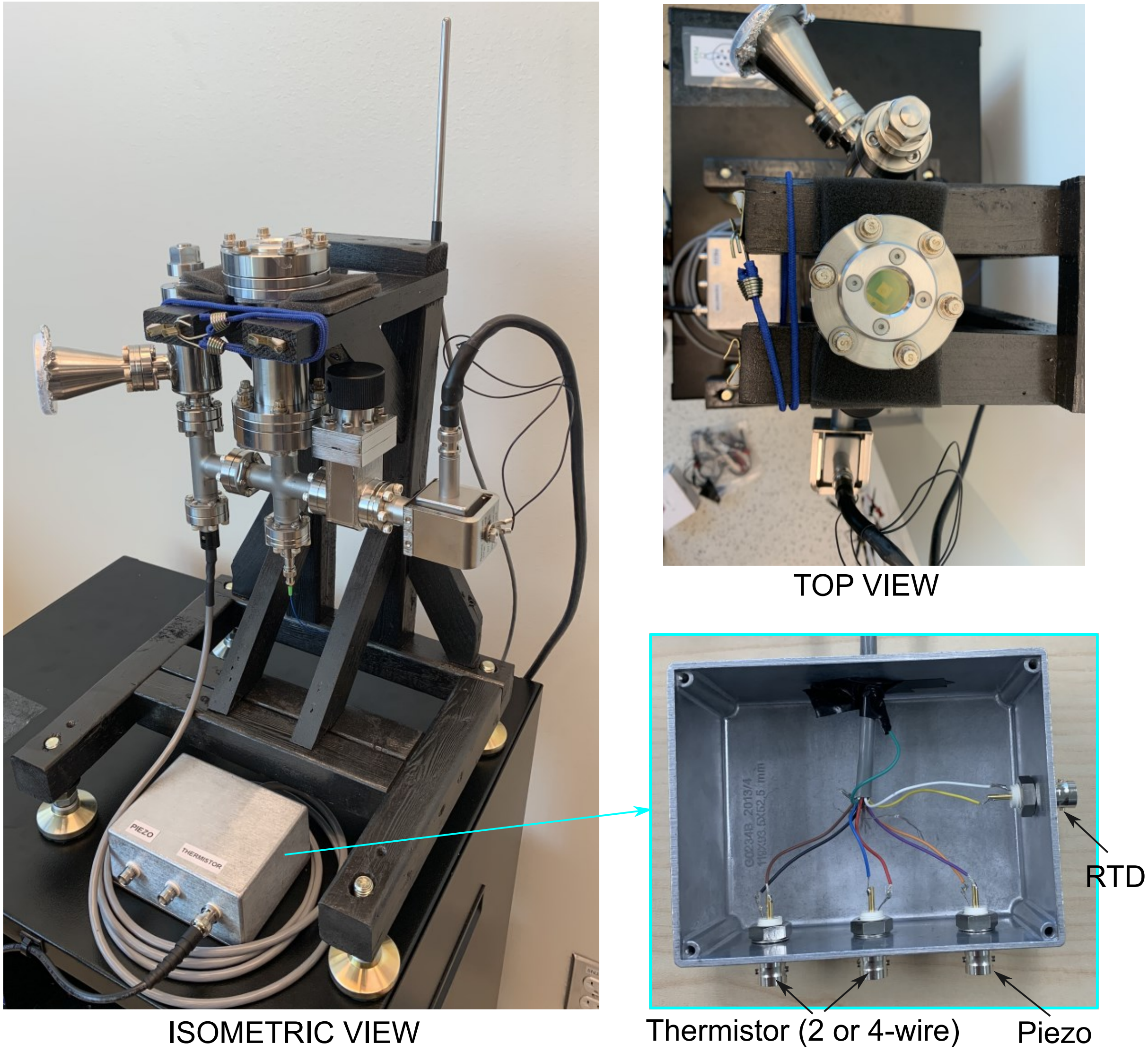


Figure A1. Vacuum chamber isometric and top view pictures, including the wiring of the electrical connection enclosure.

F. VACUUM CHAMBER BAKING

Vacuum chamber baking is used as an extra step in the pump down procedure to promote outgassing, and thus reduce high gas loads in the chamber once baking is complete. To bake the vacuum chamber in this experiment, the following procedure is used:

1. Make sure the vacuum chamber is secured in a location that allows for high temperature settings (i.e., no wooden objects nearby).
    - In this experiment, the chamber was fixed on an elevated breadboard, with minimal contact area with other objects (see Figure A2).
2. Stick an adhesive thermocouple (Omega SA2C-T) to the vacuum chamber (see Figure A2a).
3. Wrap the vacuum chamber with a fiberglass heater rope (Omega FGR-030). Make sure the rope is tight and covers the entire surface area of the chamber, except for the ion pump valve (see Figure A2a). Make sure the rope does not overlap or touch itself.
4. Connect the thermocouple and the heater rope to the temperature controller (Automation Direct SOLO 4848).
5. Turn on the turbomolecular pump and bake the chamber for 24 hours.
    - Limit the bake to 150 °C to avoid damaging the viton o-ring inside the viewport cover. If no viton o-ring is used (i.e., a standard viewport with a copper gasket is used), the bake may go up to 250 °C (limit of the ion pump).
    - To reduce heat loss, cover the chamber with aluminium foil (see Figure A2b).
6. After 24 hours, turn off the bake and wait for the chamber to cool down (this may take over one hour). This allows the final outgassing to be pumped by the turbomolecular pump instead of the ion pump.

7. Once the chamber is at ambient temperature and its pressure has stabilized, disconnect the heater rope. Close the roughing pump valve and open the ion pump valve.
8. The chamber is now kept at high vacuum by the ion pump and it is ready for experiments.

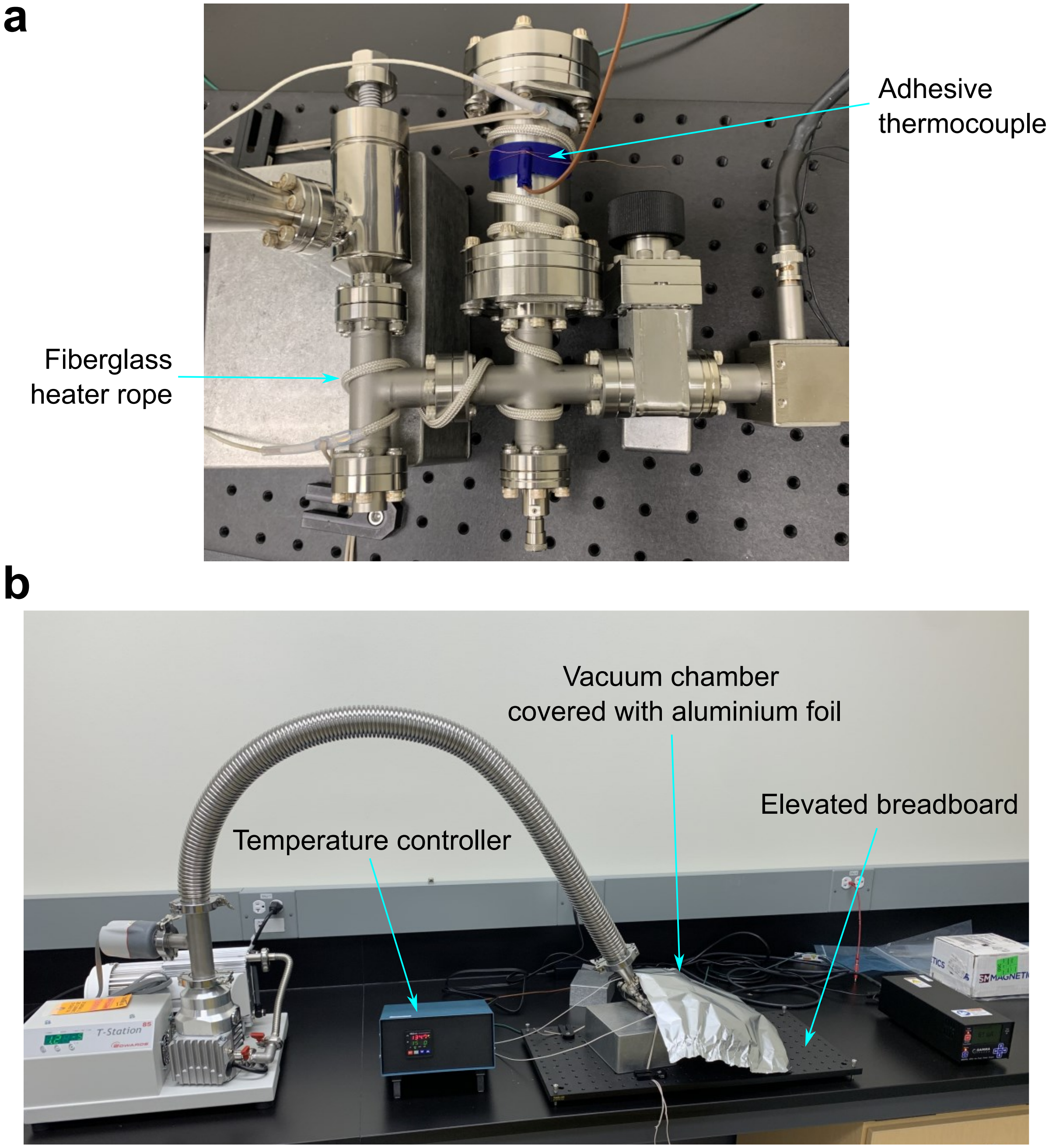


Figure A2. Vacuum chamber baking. a) heater rope wrapping technique, and b) baking setup.

In this work the chamber was baked on 02-09-2022. However, upon opening of the chamber on 06-29-2022, a black circle was observed on the middle of the membrane (see Figure A3). This may have been caused by an accumulation of outgassing from the membrane mount to be deposited on the backside of the membrane during the bake. No major changes to the resonance frequency of the membrane were observed, but the membrane was replaced nonetheless. No baking was performed subsequently.

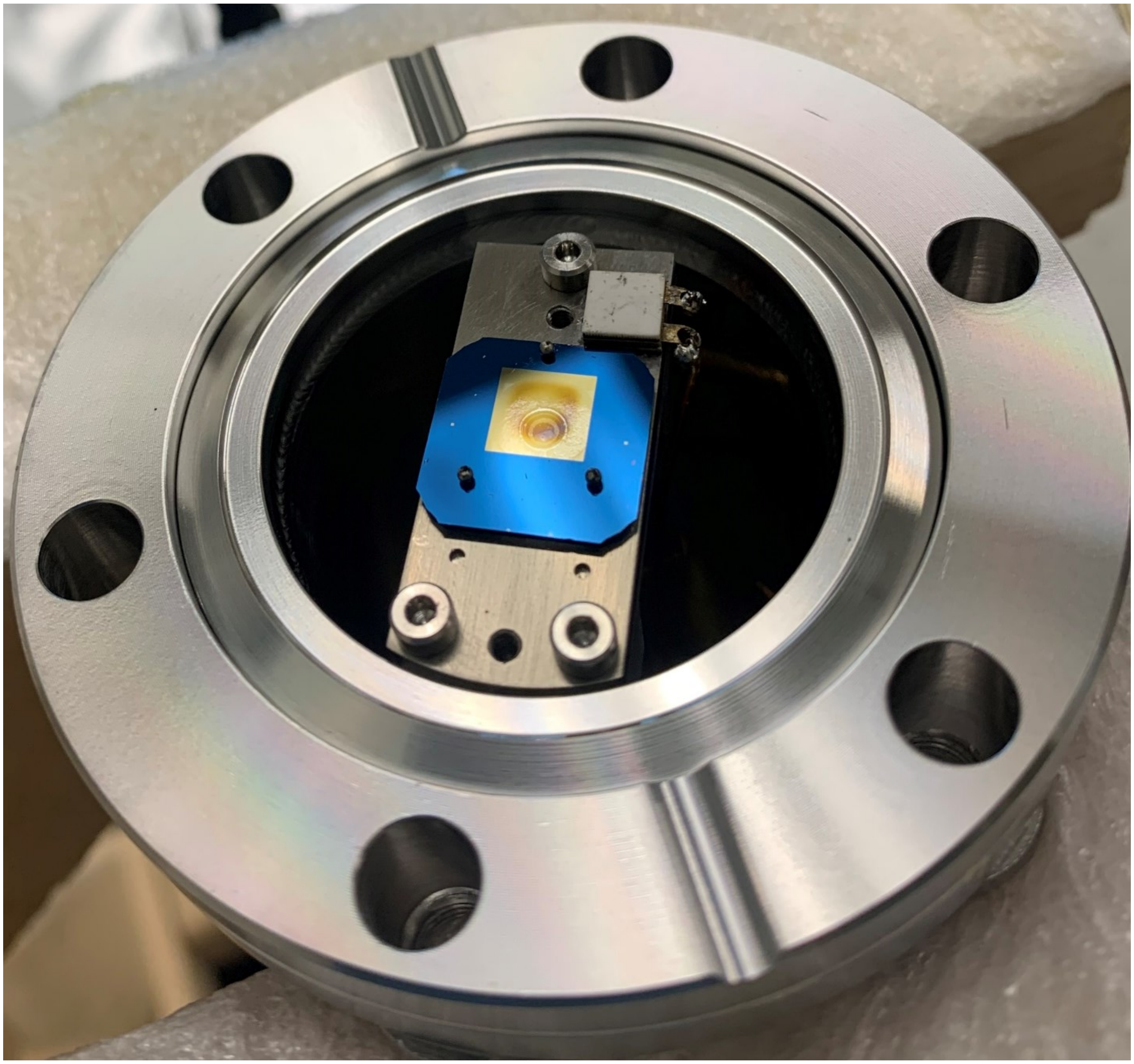


Figure A3. Membrane mount outgassing deposit on backside of membrane.

G. EXPERIMENTAL SETUP TRANSPORTATION METHOD

The experimental setup was transported to the rural locations (approximately 100 km away from the University of Ottawa) by car. See Figure A4 for mounting method in the car (Hyundai Kona 2019 with back seats flattened).

Figure A4. Experimental setup transportation method to rural locations.

H. VIEWPORT ANGLE MEASUREMENTS

The viewport angle (θ_v) is calculated using simple trigonometry with CAD measurements of the mounting setup. From experiment 3 and onwards (refer to Table 5.1), the membrane mount location inside the nipple of the vacuum chamber is depicted in Figure A5. Hence, the viewport angle is calculated as $\theta_v = \tan^{-1}\left({}^{10.16}/_{12.59}\right) = 0.68$.

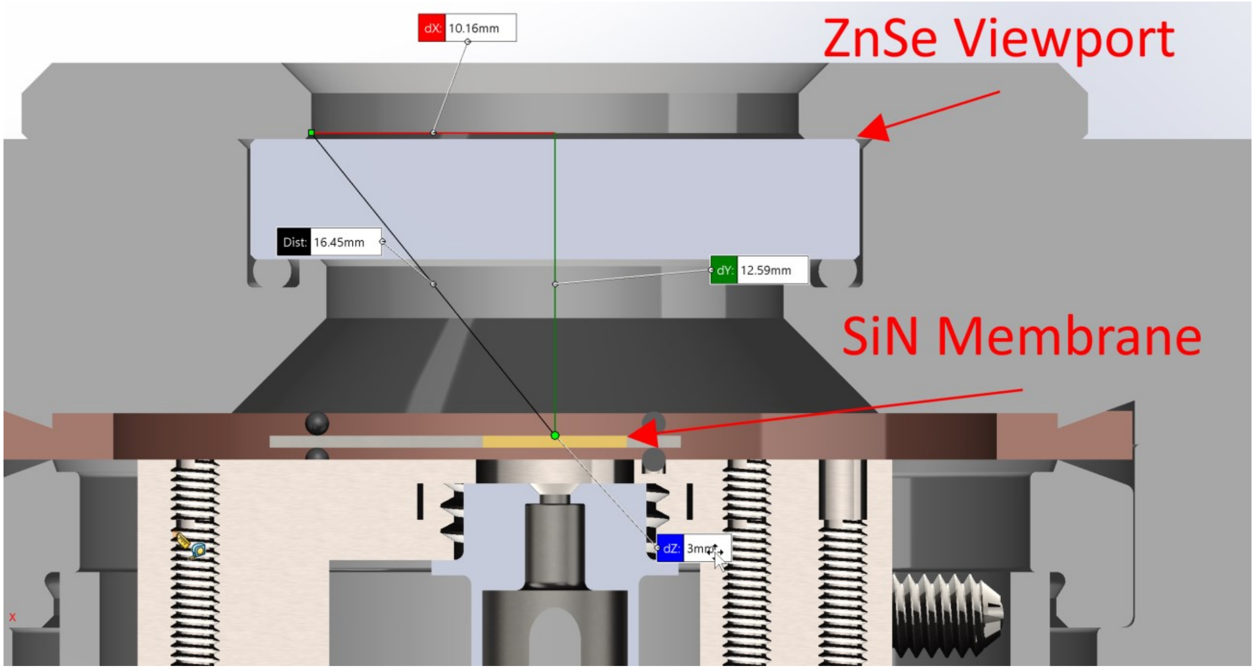


Figure A5. Viewport Angle CAD Measurements.

For experiments 1 and 2 (refer to Table 5.1), the membrane mount was located lower in the nipple of the vacuum chamber, with a dimension dY ≈ 21 mm. This means $\theta_v = \tan^{-1}\left({}^{10.16}/_{21.00}\right) = 0.45$ for these experiments.

I. SILICON NITRIDE MEMBRANE EMISSIVITY ANGLE DEPENDENCY

The spectral directional emissivity ($\varepsilon_{\lambda,\theta,m}$) for the zenith angles (θ) of 0, 30, and 60 degrees, along with the spectral hemispherical emissivity ($\varepsilon_{\lambda,m}$), of a 90 nm thick SiN membrane is shown in Figure A6. This shows that the angle dependency of the SiN membrane emissivity is relatively weak.

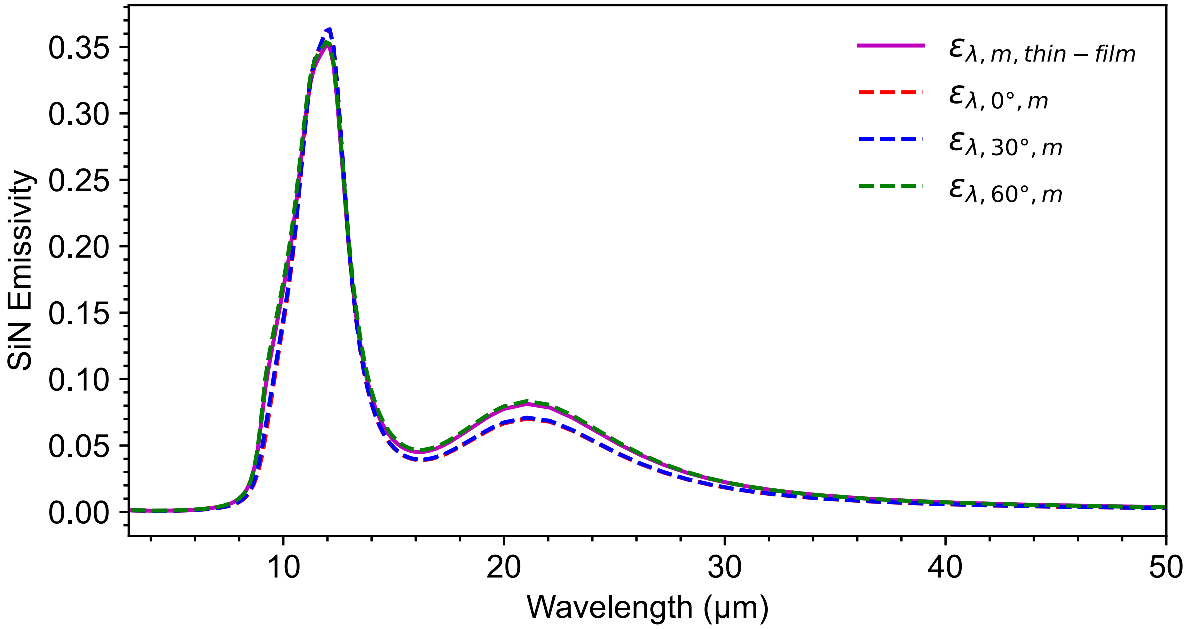


Figure A6. Spectral hemispherical emissivity, and spectral directional emissivity for the zenith angles (θ) of 0, 30, and 60 degrees, of a 90 nm thick SiN thin film membrane.

J. PYTHON SCRIPT: THEORETICAL MODEL NUMERICAL SOLVER

```
"""
Description: This script is a numerical solver for the theoretical temperature drop of a SiN membrane upon radiative
cooling.
Author: Alexandre Bouchard
Date: October 20, 2022
"""

## -- IMPORTED LIBRARIES -- ##
from numpy import sqrt, pi, array, zeros, real, cos, sin, linspace, exp, genfromtxt, conjugate, interp, concatenate
from scipy.integrate import quad, trapz
from scipy.special import iv
import matplotlib
import matplotlib.pyplot as plt
import pandas as pd


## -- USER INPUTS -- ##
thickness = [90e-9, 0.0014] #Enter the thickness of the layers as follows #[SiN, Vacuum] [m].
T_amb = 300 #Ambient temperature [K]
T_cell = 300 #Cell Temperature [K]
theta_v = 0.68 #Viewport angle (where radiation comes from chamber and not atmosphere) [rad]
theta_sun = 0 #Use theta_sun calculator to determine this. Use "1" if nighttime


## -- WAVELENGTH TABLE GENERATION -- ##
"""
Description: Generating the wavelength range and density of points in the numerical solver.
Notes: 1) We want more points in 0.28-20 um range because solar radiation starts at 0.28 um and atmospheric
transparency is mostly null above 20 um.
"""
#Wavelength Table variables
lambda_min = 280e-9 #Start Wavelength (beginning of I_AM1.5 data)
lambda_thresh_1 = 3e-6 #Solar Threshold wavelength.
lambda_thresh_2 = 20e-6 #Atm/ZnSeThreshold wavelength.
lambda_max = 50e-6 #End Wavelength.
iterations_1 = 100 #Number of iterations for first range of wavelength table
iterations_2 = 100 #Number of iterations for second range of wavelength table
```

```
iterations_3 = 30 #Number of iterations for third range of wavelength table

#Generating Wavelength Table
wl_1 = linspace(lambda_min, lambda_thresh_1, iterations_1) #First range of wavelength
wl_2 = linspace(lambda_thresh_1, lambda_thresh_2, iterations_2) #Second range of wavelength
wl_3 = linspace(lambda_thresh_2, lambda_max, iterations_3) #Third range of wavelength
wavelength = concatenate((wl_1, wl_2[1:], wl_3[1:]))
plt.scatter(wavelength*1e6, wavelength*1e6, s=0.5)
plt.title("Density of points in wavelength table")
plt.xlabel(u"Wavelength (\u03bcm)")
plt.ylabel(u"Wavelength (\u03bcm)")
plt.show()

#Finding important indices
#20um (end of atm and ZnSe data)
for i in range(wavelength.size):
    if wavelength[i] > 20e-6:
        index_20um = i
        break
#4um (end of AM1.5 data)
for i in range(wavelength.size):
    if wavelength[i] > 4e-6:
        index_4um = i
        break

#5um (beginning of Lumerical metasurface data)
for i in range(wavelength.size):
    if wavelength[i] > 5e-6:
        index_5um = i
        break

## -- FUNCTION: SiN_complex_refractive_index -- ##
"""
Description: Calculate the complex refractive index of SiN according to the Maxwell-Helmoltz-Drude Model (Cataldo et
al., 2012).
Input(s): wavelength - wavelength of radiation [m]
Return(s): N - complex refractive index of SiN for input wavelength
"""
#Constants
M = 5 #Number of Oscillators
e_inf = 4.562+0.0124j #Relative permittivity
```

```
e = array([7.582, 6.754+0.3759j, 6.601+0.0041j, 5.43+0.1179j, 4.601+0.2073j, 4.562+0.0124j]) #Difference in relative
complex dielectric constant between adjacent oscillators
w_T = 2*pi*1E12*array([13.913, 15.053, 24.521, 26.440, 31.724]) #Oscillator resonance frequencies
gamma_SiN = 2*pi*1E12*array([5.810, 6.436, 2.751, 3.482, 5.948]) #Damping coefficient
alpha = array([0.0001, 0.3427, 0.0006, 0.0002, 0.0080]) #Interpolation term
c = 299792458 #speed of light [m/s]
def SiN_complex_refractive_index(wavelength):
    #Philip Dispersion Model for wavelengths below 3 um.
    if wavelength < 3e-6:
        n = sqrt(1+(2.8939*(wavelength/1000000)**2)/((wavelength/1000000)**2-0.13967**2))
        N = n #k = 0  in this range

    #Maxwell-Helmoltz-Drude Model for wavelengths in the mid-far infrared range (3-1000um)
    elif wavelength >= 3e-6:
        w = (2*pi*c)/(wavelength) # Converting wavelength to frequency
        e_w = e_inf
        delta_e = zeros(M, dtype=complex)
        for i in range(delta_e.size):
            delta_e[i] = e[i]-e[i+1]
            gamma_SiN_w = gamma_SiN[i]*exp(-alpha[i]*((w_T[i]**2-w**2)/(w*gamma_SiN[i]))**2)
            e_w = e_w+(delta_e[i]*w_T[i]**2)/(w_T[i]**2-w**2-1j*w*gamma_SiN_w)
        N = sqrt(e_w)

    return N


## -- FUNCTION: aluminium_refractive_index -- ##
"""
Description: Calculate the complex refractive index of aluminium using the Drude model
Input(s): wavelength - wavelength of radiation [m]
Return(s): N - complex refractive index of aluminium for input wavelength
"""
#Constants
wp = 1.035e5 #Plasma frequency of aluminium between 0-200 um [cm-1]
gamma_al = 5.4e2 #Collision Frequency of aluminium between 0-200 um [cm-1]
def aluminium_refractive_index(wavelength):
    wavenumber = 1/(wavelength*100) #x100 for m to cm
    e_w = 1-((wp**2)/(wavenumber**2+1j*gamma_al*wavenumber)) #dielectric function (permittivity)
    N = sqrt(e_w) #complex refractive index
    return N
```

```
## -- FUNCTION: absorption -- ##
"""
Description: Calculate the absorptivity of the SiN membrane according to the characteristic matrix formalism (Macleod,
2018).
Input(s): thickness - thickness of each layer in the multilayer (SiN, vacuum, aluminium) [m]
          wavelength - wavelength of radiation [m]
          inc_angle - incidence angle of radiation [rad]
          case - Specification of which emissivity model to use (thin film or multilayer)
Return(s): A_spect_dir - Spectral directional absorptivity of thin film or multilayer
           T_spect_dir - Spectral directional transmissivity of thin film or multilayer
           R_spect_dir - Spectral directional reflectivity of thin film or multilayer
"""
def absorption(thickness, wavelength, inc_angle, case):
    opt_admittance = 1 #2.6544e-3 in free space units and 1 in gaussian units
    ##LAYER CHARACTERISTICS CALCULATIONS
    #Incident medium - Vacuum
    N_inc = 1+0j
    eta_inc_s = opt_admittance*real(N_inc*cos(inc_angle))
    eta_inc_p = opt_admittance*real(N_inc/cos(inc_angle))
    #Thin Film Layer 1 - SiN
    thickness_1 = thickness[0] #[m]
    N1 = conjugate(SiN_complex_refractive_index(wavelength)) #Conjugate because of Mcleod convention
    cos_theta_1 = sqrt(1-(N_inc*sin(inc_angle)/N1)**2)
    eta_1_s = opt_admittance*N1*cos_theta_1
    eta_1_p = opt_admittance*N1/cos_theta_1
    delta_1 = 2*pi*N1*thickness_1*cos_theta_1/wavelength
    layer_matrix_1_s = array([[cos(delta_1), sin(delta_1)*1j/eta_1_s], [eta_1_s*sin(delta_1)*1j, cos(delta_1)]],
dtype=complex)
    layer_matrix_1_p = array([[cos(delta_1), sin(delta_1)*1j/eta_1_p], [eta_1_p*sin(delta_1)*1j, cos(delta_1)]],
dtype=complex)
    if case == "multi": #Multliayer Emissivity Model
        #Thin Film Layer 2 - Vacuum
        thickness_2 = thickness[1]
        N2 = 1-0j
        cos_theta_2 = sqrt(1-(N_inc*sin(inc_angle)/N2)**2)
        eta_2_s = opt_admittance*N2*cos_theta_2
        eta_2_p = opt_admittance*N2/cos_theta_2
        delta_2 = 2*pi*N2*thickness_2*cos_theta_2/wavelength
        layer_matrix_2_s = array([[cos(delta_2), sin(delta_2)*1j/eta_2_s], [eta_2_s*sin(delta_2)*1j, cos(delta_2)]],
dtype=complex)
```

```
        layer_matrix_2_p = array([[cos(delta_2), sin(delta_2)*1j/eta_2_p], [eta_2_p*sin(delta_2)*1j, cos(delta_2)]],
dtype=complex)
        #Emergent medium/Substrate - Aluminium
        N_m = conjugate(aluminium_refractive_index(wavelength)) #Conjugate because of Mcleod convention
        cos_theta_m = sqrt(1-(N_inc*sin(inc_angle)/N_m)**2)
        eta_m_s = opt_admittance*N_m*cos_theta_m
        eta_m_p = opt_admittance*N_m/cos_theta_m
        m_matrix_s = array([[1], [eta_m_s]], dtype=complex)
        m_matrix_p = array([[1], [eta_m_p]], dtype=complex)
        #Characteristic matric calculation
        char_matrix_s = layer_matrix_1_s.dot(layer_matrix_2_s).dot(m_matrix_s)
        char_matrix_p = layer_matrix_1_p.dot(layer_matrix_2_p).dot(m_matrix_p)
    elif case == "SiN": #Thin Film Emissivity Model
        # Emergent medium/Substrate - Vacuum
        N_m = 1+0j
        cos_theta_m = sqrt(1-(N_inc*sin(inc_angle)/N_m)**2)
        eta_m_s = opt_admittance*N_m*cos_theta_m
        eta_m_p = opt_admittance*N_m/cos_theta_m
        m_matrix_s = array([[1], [eta_m_s]], dtype=complex)
        m_matrix_p = array([[1], [eta_m_p]], dtype=complex)
        #Characteristic matric calculation
        char_matrix_s = layer_matrix_1_s.dot(m_matrix_s)
        char_matrix_p = layer_matrix_1_p.dot(m_matrix_p)

    #p-polarized and s-polarized characteristic matrix
    B_s = char_matrix_s[0][0]
    C_s = char_matrix_s[1][0]
    B_p = char_matrix_p[0][0]
    C_p = char_matrix_p[1][0]
    #s-polarized transmissivity and reflectivity
    T_s = (4*eta_inc_s*real(eta_m_s))/((eta_inc_s*B_s+C_s)*conjugate(eta_inc_s*B_s+C_s))
    R_s = ((eta_inc_s*B_s-C_s)/(eta_inc_s*B_s+C_s))*(conjugate((eta_inc_s*B_s-C_s)/(eta_inc_s*B_s+C_s)))
    #p-polarized transmissivity and reflectivity
    T_p = (4*eta_inc_p*real(eta_m_p))/((eta_inc_p*B_p+C_p)*conjugate(eta_inc_p*B_p+C_p))
    R_p = ((eta_inc_p*B_p-C_p)/(eta_inc_p*B_p+C_p))*(conjugate((eta_inc_p*B_p-C_p)/(eta_inc_p*B_p+C_p)))
    #unpolarized transmissivity and reflectivity
    T_spect_dir = (real(T_s)+real(T_p))/2
    R_spect_dir = (real(R_s)+real(R_p))/2
    #unpolarized absorptivity
    A_spect_dir = 1-R_spect_dir-T_spect_dir
```

```
    return A_spect_dir,  T_spect_dir,  R_spect_dir


## -- FUNCTION: BB_Intensity -- ##
"""
Description: Calculate the spectral radiative intensity of a blackbody
Input(s): wavelength - wavelength of radiation [m]
          T - Temperature [K]
Return(s): I_BB - Blackbody intensity [W/sqm/m/sr]
"""
h = 6.626e-34 #Planck constant [J/s]
k_b = 1.381e-23 #Boltzmann constant [J/K]
def BB_Intensity(wavelength, T):
    I_BB = (2*h*c**2)/(wavelength**5*(exp((h*c)/(wavelength*k_b*T))-1))
    return I_BB


## -- FUNCTION: interpolation -- ##
"""
Description: Interpolate the wavelength values (x-axis) of all relevant data arrays to match with the input wavelength
array.
Input(s): wavelength - wavelength of radiation [m]
          index_4um - wavelength table index for 4 um
          index_20um - wavelength table index for 20 um
          index_5um - wavelength table index for 5 um
Return(s): t_atm_mid_summer - interpolated array for mid-summer atmospheric transmissivity as a function of wavelength
           t_atm_arc_winter - interpolated array for subarctic winter atmospheric transmissivity as a function of
wavelength
           t_ZnSe - interpolated array for ZnSe transmissivity as a function of wavelength
           I_AM1_5 - interpolated array for Am1.5 solar intensity as a function of wavelength
           A_met_match - interpolated array for absorptivity of metasurface as a function of wavelength
"""
#MODTRAN Atmospheric Transmissivity data
t_atm_table_mid_summer = genfromtxt(r"C:\Users\alexandre.bouchard\t_atm - Mid-Latitude Summer.csv", delimiter=',',
skip_header=1)
t_atm_table_mid_winter = genfromtxt(r"C:\Users\alexandre.bouchard\t_atm - Mid-Latitude Winter.csv", delimiter=',',
skip_header=1)
t_atm_table_arc_summer = genfromtxt(r"C:\Users\alexandre.bouchard\t_atm - Sub-Arctic Summer.csv", delimiter=',',
skip_header=1)
t_atm_table_arc_winter = genfromtxt(r"C:\Users\alexandre.bouchard\t_atm - Sub-Arctic Winter.csv", delimiter=',',
skip_header=1)
```

```
#Thorlabs ZnSe viewport transmissivity data
t_ZnSe_table = genfromtxt("C:\\Users\\alexandre.bouchard\\t_ZnSe.csv", delimiter=',', skip_header=1)
#AM1.5 solar intensity spectrum data
I_AM1_5_table = genfromtxt("C:\\Users\\alexandre.bouchard\\I_AM1.5.csv", delimiter=',', skip_header=1)
#Lumerical (Arthur's) metasurface data
R_met_table = genfromtxt(r"C:\Users\alexandre.bouchard\R_data_90nmSiN_wcross.txt", delimiter=',', skip_header=1)
T_met_table = genfromtxt(r"C:\Users\alexandre.bouchard\T_data_90nmSiN_wcross.txt", delimiter=',', skip_header=1)
wavelength_met = 2.998e8/(R_met_table[:,0]*1e12) #converting frequency (THz) to wavelength (m)
R_met = R_met_table[:,1]
T_met = T_met_table[:,1]
A_met = abs(1-R_met-T_met) #metasurface absorptivity calculation from reflectivity and transmissivity data
def interpolation(wavelength, index_4um, index_20um, index_5um):
    #Interpolating data arrays to match input wavelength array
    t_atm_mid_summer = interp(wavelength, t_atm_table_mid_summer[:,0], t_atm_table_mid_summer[:,1])
    t_atm_arc_winter = interp(wavelength, t_atm_table_arc_winter[:,0], t_atm_table_arc_winter[:,1])
    t_ZnSe = interp(wavelength, t_ZnSe_table[:,0], t_ZnSe_table[:,1])
    I_AM1_5 = interp(wavelength, I_AM1_5_table[:,0], I_AM1_5_table[:,1])

    #Setting transmission values to 0 after 20um
    for i in range(index_20um, wavelength.size):
        t_atm_mid_summer[i] = 0
        t_atm_arc_winter[i] = 0
        t_ZnSe[i] = 0

    #Setting I_AM_1.5 values to 0 after 4um (end of standard data)
    for i in range(index_4um, wavelength.size):
        I_AM1_5[i] = 0

    #Cutting lumerical data to match wavelength array
    A_met_match = zeros(wavelength.size)
    for i in range(wavelength.size):
        for j in range (wavelength_met.size):
            if wavelength[i] > wavelength_met[j]:
                A_met_match[i] = A_met[j-1]
                break

    return t_atm_mid_summer, t_atm_arc_winter, t_ZnSe, I_AM1_5, A_met_match


## -- FUNCTION: T_m_solution -- ##
"""
```

```
Description: Determine the temperature drop of the SiN membrane upon radiative cooling. This is done by numerically
solving the energy balance with iterations of T_m values until steady state is reached.
Input(s): wavelength - wavelength of radiation [m]
          thickness - thickness of each layer in the multilayer (SiN, vacuum, aluminium) [m]
          T_amb - Ambient temperature input [K]
          T_cell - vacuum chamber temperature input [K]
          theta_v - viewport angle [rad]
          theta_sun - angle at which the sun is facing the membrane [rad]
          index_4um - wavelength table index for 4 um
          index_20um - wavelength table index for 20 um
          index_5um - wavelength table index for 5 um
Return(s): T_m_(best, best, ideal, met) - Membrane temperature drops (K) per case
           e_spect_hem_(best, worst, ideal) - Membrane spectral hemispherical emissivity per case
           e_spect_(0, 30, 60) - Membrane spectral direction emissivity for incidence angles of 0, 30, and  60
degrees.
           e_tot_hem - Membrane total hemispherical emissivity
           t_atm_mid_summer - interpolated array for mid-summer atmospheric transmissivity as a function of wavelength
           t_atm_arc_winter - interpolated array for subarctic winter atmospheric transmissivity as a function of
wavelength
           t_ZnSe - interpolated array for ZnSe transmissivity as a function of wavelength
           I_AM1_5 - interpolated array for Am1.5 solar intensity as a function of wavelength
           A_met - interpolated array for absorptivity of metasurface as a function of wavelength
           (worst, best, ideal, met)_case_df - dataframes with heat transfer rate values at each iteration.
Notes: The cases are represented by:
            best: calculated using the multilayer model emissivity and sub-arctic winter atmospheric transparency
            worst: calculated using thin film model emissivity and mid-latitude summer atmospheric transparency
            ideal: calculated using the emissivity of an ideal emitter and sub-arctic winter atmospheric transparency
            met: calculated using the emissivity from Lumerical data and mid latitude summer atmospheric transparency
        The heat transfer rates are represented by (in W/sqm):
            q_m - Membrane outward radiation
            q_atm - Atmospheric irradiation
            q_c - Topside chamber irradiation from chamber walls portion
            q_v - Topside chamber irradiation from viewport portion
            q_backside - Chamber irradiation on backside of membrane
            q_sun - Solar irradiation
            q_cond - Conduction from silicon frame to membrane
"""
def T_m_solution(wavelength, thickness, T_amb, T_cell, theta_v, theta_sun, index_4um, index_20um, index_5um):

    #Grabbing interpolated data tables
```

```
    t_atm_mid_summer, t_atm_arc_winter, t_ZnSe, I_AM1_5, A_met  = interpolation(wavelength, index_4um, index_20um,
index_5um)

    #Determining the emissivity terms, as well as BB intesities, for each heat transfer rate
    e_ideal = zeros(wavelength.size)
    e_m_all_best = zeros(wavelength.size)
    e_m_all_worst = zeros(wavelength.size)
    e_m_all_ideal = zeros(wavelength.size)
    e_m_all_met = zeros(wavelength.size)
    e_atm_best = zeros(wavelength.size)
    e_atm_worst = zeros(wavelength.size)
    e_atm_ideal = zeros(wavelength.size)
    e_atm_met = zeros(wavelength.size)
    e_m_v_best = zeros(wavelength.size)
    e_m_v_worst = zeros(wavelength.size)
    e_m_v_ideal = zeros(wavelength.size)
    e_m_v_met = zeros(wavelength.size)
    e_m_c_best = zeros(wavelength.size)
    e_m_c_worst = zeros(wavelength.size)
    e_m_c_ideal = zeros(wavelength.size)
    e_m_c_met = zeros(wavelength.size)
    e_m_backside = zeros(wavelength.size)
    e_m_backside_met = zeros(wavelength.size)
    e_sun_best = zeros(wavelength.size)
    e_sun_worst = zeros(wavelength.size)
    e_sun_ideal = zeros(wavelength.size)
    e_sun_met = zeros(wavelength.size)
    e_spect_hem_best = zeros(wavelength.size)
    e_spect_hem_worst = zeros(wavelength.size)
    e_spect_hem_ideal = zeros(wavelength.size)
    e_spect_0_best = zeros(wavelength.size)
    e_spect_0_worst = zeros(wavelength.size)
    e_spect_30_best = zeros(wavelength.size)
    e_spect_30_worst = zeros(wavelength.size)
    e_spect_60_best = zeros(wavelength.size)
    e_spect_60_worst = zeros(wavelength.size)
    I_BB_amb = zeros(wavelength.size)
    I_BB_cell = zeros(wavelength.size)
    print("Starting Emissivity Loop:")
    for i in range(wavelength.size):
        print("Emissivity Loop iteration",i," of ",wavelength.size)
```

```
        #Defining ideal radiative cooler emissivity
        if wavelength[i]<8e-6 or wavelength[i]>13e-6:
            e_ideal[i] = 0
        else:
            e_ideal[i] = 1

        #Membrane outward radiation emissivities
        e_m_all_best[i] = quad(lambda inc_angle: absorption(thickness, wavelength[i], inc_angle,
"multi")[0]*cos(inc_angle)*sin(inc_angle), 0, pi/2)[0]
        e_m_all_worst[i] = quad(lambda inc_angle: absorption(thickness, wavelength[i], inc_angle,
"SiN")[0]*cos(inc_angle)*sin(inc_angle), 0, pi/2)[0]
        e_m_all_ideal[i] = quad(lambda inc_angle: e_ideal[i]*cos(inc_angle)*sin(inc_angle), 0, pi/2)[0]
        e_m_all_met[i] = quad(lambda inc_angle: A_met[i]*cos(inc_angle)*sin(inc_angle), 0, pi/2)[0]

        #Atmospheric irradiation emissivities
        e_atm_best[i] = quad(lambda inc_angle: (absorption(thickness, wavelength[i], inc_angle, "multi"))[0]*(1-
t_atm_arc_winter[i]**(1/cos(inc_angle)))*cos(inc_angle)*sin(inc_angle), 0, theta_v)[0]
        e_atm_worst[i] = quad(lambda inc_angle: (absorption(thickness, wavelength[i], inc_angle, "SiN"))[0]*(1-
t_atm_mid_summer[i]**(1/cos(inc_angle)))*cos(inc_angle)*sin(inc_angle), 0, theta_v)[0]
        e_atm_ideal[i] = quad(lambda inc_angle: e_ideal[i]*(1-
t_atm_arc_winter[i]**(1/cos(inc_angle)))*cos(inc_angle)*sin(inc_angle), 0, theta_v)[0]
        e_atm_met[i] = quad(lambda inc_angle: A_met[i]*(1-
t_atm_mid_summer[i]**(1/cos(inc_angle)))*cos(inc_angle)*sin(inc_angle), 0, theta_v)[0]

        #Viewport irradiation emissivities:
        e_m_v_best[i] = quad(lambda inc_angle: absorption(thickness, wavelength[i], inc_angle, "multi")[0]*(1-
t_ZnSe[i])*cos(inc_angle)*sin(inc_angle), 0, theta_v)[0]
        e_m_v_worst[i] = quad(lambda inc_angle: absorption(thickness, wavelength[i], inc_angle, "SiN")[0]*(1-
t_ZnSe[i])*cos(inc_angle)*sin(inc_angle), 0, theta_v)[0]
        e_m_v_ideal[i] = quad(lambda inc_angle: e_ideal[i]*(1-t_ZnSe[i])*cos(inc_angle)*sin(inc_angle), 0, theta_v)[0]
        e_m_v_met[i] = quad(lambda inc_angle: A_met[i]*(1-t_ZnSe[i])*cos(inc_angle)*sin(inc_angle), 0, theta_v)[0]

        #Chamber irradiation emissivities:
        if theta_v == pi/2:
            e_m_c_best[i] = 0
            e_m_c_worst[i] = 0
            e_m_c_ideal[i] = 0
            e_m_c_met[i] = 0
        else:
            e_m_c_best[i] = quad(lambda inc_angle: absorption(thickness, wavelength[i], inc_angle,
"multi")[0]*cos(inc_angle)*sin(inc_angle), theta_v, pi/2)[0]
```

```
            e_m_c_worst[i] = quad(lambda inc_angle: absorption(thickness, wavelength[i], inc_angle,
"SiN")[0]*cos(inc_angle)*sin(inc_angle), theta_v, pi/2)[0]
            e_m_c_ideal[i] = quad(lambda inc_angle: e_ideal[i]*cos(inc_angle)*sin(inc_angle), theta_v, pi/2)[0]
            e_m_c_met[i] = quad(lambda inc_angle: A_met[i]*cos(inc_angle)*sin(inc_angle), theta_v, pi/2)[0]

        #Backside Irradiation
        e_m_backside[i] = quad(lambda inc_angle: absorption(thickness, wavelength[i], inc_angle,
"SiN")[0]*cos(inc_angle)*sin(inc_angle), 0, pi/2)[0]
        e_m_backside_met[i] = quad(lambda inc_angle: A_met[i]*cos(inc_angle)*sin(inc_angle), 0, pi/2)[0]

        #Emissivity at solar angle
        if theta_sun != 1: #"1" represents nighttime
            e_sun_best[i] = absorption(thickness, wavelength[i], theta_sun, "multi")[0]
            e_sun_worst[i] = absorption(thickness, wavelength[i], theta_sun, "SiN")[0]
            e_sun_ideal[i] = e_ideal[i]
            e_sun_met[i] = A_met[i]

        #Spectral Hemispherical Emissivity
        e_spect_hem_best[i] = 2*quad(lambda inc_angle: absorption(thickness, wavelength[i], inc_angle,
"multi")[0]*cos(inc_angle)*sin(inc_angle),0,pi/2)[0]
        e_spect_hem_worst[i] = 2*quad(lambda inc_angle: absorption(thickness, wavelength[i], inc_angle,
"SiN")[0]*cos(inc_angle)*sin(inc_angle),0,pi/2)[0]
        e_spect_hem_ideal[i] = 2*quad(lambda inc_angle: e_ideal[i]*cos(inc_angle)*sin(inc_angle),0,pi/2)[0]

        #Directional Emissivities
        e_spect_0_best[i] = absorption(thickness, wavelength[i], 0, "multi")[0] #Normal
        e_spect_0_worst[i] = absorption(thickness, wavelength[i], 0, "SiN")[0] #Normal
        e_spect_30_best[i] = absorption(thickness, wavelength[i], 30*pi/180, "multi")[0] #30 degrees
        e_spect_30_worst[i] = absorption(thickness, wavelength[i], 30*pi/180, "SiN")[0] #30 degrees
        e_spect_60_best[i] = absorption(thickness, wavelength[i], 60*pi/180, "multi")[0] #60 degrees
        e_spect_60_worst[i] = absorption(thickness, wavelength[i], 60*pi/180, "SiN")[0] #60 degrees

        # BB Intensities
        I_BB_amb[i] = BB_Intensity(wavelength[i], T_amb) #at ambient temperature
        I_BB_cell[i] = BB_Intensity(wavelength[i], T_cell) #at chamber temperature

    #Total Hemispherical Emissivity for SiN thin film
    e_tot_hem = trapz(e_spect_hem_worst*pi*I_BB_amb, wavelength)/trapz(pi*I_BB_amb, wavelength)

    #Calculating x_cond for conduction
    r_eff = 1.252*6e-3/2 #membrane equivalent radius
```

```
    B = sqrt(8*5.67e-8*e_tot_hem*T_cell**3/(2.7*thickness[0]))
    B_reff = B*r_eff
    x_cond = (2/B_reff)*(iv(1, B_reff)/iv(0, B_reff))

    ## NUMERICAL SOLVER FOR T_m FOR EACH CASE
    #initializing variables
    T_m_worst = T_cell #Initial membrane temperature (assume it is initially at ambient temperature) [K]
    T_m_best = T_cell
    T_m_ideal = T_cell
    T_m_met = T_cell
    T_m_i = 0.1 #Membrane Temperature increment [K]

    # BEST CASE SCENARIO LOOP
        #calculated using the multilayer model emissivity and sub-arctic winter atmospheric transparency
    T_m_best_list = []
    q_m_best_list = []
    q_atm_best_list = []
    q_c_best_list = []
    q_v_best_list = []
    q_cond_best_list = []
    q_sun_best_list = []
    stead_state_test_best_list = []
    steady_state_test_best = 100  # (0 = steady state)
    while steady_state_test_best > 0:
        I_BB_m_best = zeros(wavelength.size)
        for i in range(wavelength.size):
            I_BB_m_best[i] = BB_Intensity(wavelength[i], T_m_best)

        q_m_best = 2*pi*trapz(I_BB_m_best*e_m_all_best, wavelength)
        q_atm_best = 2*pi*trapz(t_ZnSe*I_BB_amb*e_atm_best, wavelength)
        q_c_best = 2*pi*trapz(I_BB_cell*e_m_c_best, wavelength)
        q_v_best = 2*pi*trapz(I_BB_cell*e_m_v_best, wavelength)
        q_cond_best = x_cond*(2*e_tot_hem*5.67e-8*(T_cell**4-T_m_best**4))/(1-x_cond)
        if theta_sun !=1:
            q_sun_best = trapz(e_sun_best*I_AM1_5, wavelength)
        else:
            q_sun_best = 0

        steady_state_test_best = q_m_best - q_atm_best - q_c_best - q_v_best - q_cond_best - q_sun_best
        if steady_state_test_best > 0:
            T_m_best -= T_m_i
```

```
        T_m_best_list.append(T_m_best)
        q_m_best_list.append(q_m_best)
        q_atm_best_list.append(q_atm_best)
        q_c_best_list.append(q_c_best)
        q_v_best_list.append(q_v_best)
        q_cond_best_list.append(q_cond_best)
        q_sun_best_list.append(q_sun_best)
        stead_state_test_best_list.append(steady_state_test_best)

    # WORST CASE SCENARIO lOOP
        #calculated using thin film model emissivity and mid-latitude summer atmospheric transparency
    T_m_worst_list = []
    q_m_worst_list = []
    q_atm_worst_list = []
    q_c_worst_list = []
    q_v_worst_list = []
    q_backside_list = []
    q_cond_worst_list = []
    q_sun_worst_list = []
    stead_state_test_worst_list = []
    steady_state_test_worst = 100  # (0 = steady state)
    while steady_state_test_worst > 0:
        I_BB_m_worst = zeros(wavelength.size)
        for i in range(wavelength.size):
            I_BB_m_worst[i] = BB_Intensity(wavelength[i], T_m_worst)

        q_m_worst = 2*(2*pi*trapz(I_BB_m_worst*e_m_all_worst, wavelength))
        q_atm_worst = 2*pi*trapz(t_ZnSe*I_BB_amb*e_atm_worst, wavelength)
        q_c_worst = 2*pi*trapz(I_BB_cell*e_m_c_worst, wavelength)
        q_v_worst = 2*pi*trapz(I_BB_cell*e_m_v_worst, wavelength)
        q_backside = 2*pi*trapz(I_BB_cell*e_m_backside, wavelength)
        if theta_sun !=1:
            q_sun_worst = trapz(e_sun_worst*I_AM1_5, wavelength)
        else:
            q_sun_worst = 0
        q_cond_worst = x_cond*(2*e_tot_hem*5.67e-8*(T_cell**4-T_m_worst**4))/(1-x_cond)

        steady_state_test_worst = q_m_worst - q_atm_worst - q_c_worst - q_v_worst - q_backside - q_cond_worst -
q_sun_worst
        if steady_state_test_worst > 0:
```

```
            T_m_worst -= T_m_i

        T_m_worst_list.append(T_m_worst)
        q_m_worst_list.append(q_m_worst)
        q_atm_worst_list.append(q_atm_worst)
        q_c_worst_list.append(q_c_worst)
        q_v_worst_list.append(q_v_worst)
        q_backside_list.append(q_backside)
        q_cond_worst_list.append(q_cond_worst)
        q_sun_worst_list.append(q_sun_worst)
        stead_state_test_worst_list.append(steady_state_test_worst)

    # IDEAL CASE SCENARIO LOOP
        #calculated using the emissivity of an ideal emitter and sub-arctic winter atmospheric transparency
    T_m_ideal_list = []
    q_m_ideal_list = []
    q_atm_ideal_list = []
    q_c_ideal_list = []
    q_v_ideal_list = []
    q_sun_ideal_list = []
    stead_state_test_ideal_list = []
    steady_state_test_ideal = 100  # (0 = steady state)
    while steady_state_test_ideal > 0:
        I_BB_m_ideal = zeros(wavelength.size)
        for i in range(wavelength.size):
            I_BB_m_ideal[i] = BB_Intensity(wavelength[i], T_m_ideal)

        q_m_ideal = 2*pi*trapz(I_BB_m_ideal*e_m_all_ideal, wavelength)
        q_atm_ideal = 2*pi*trapz(t_ZnSe*I_BB_amb*e_atm_ideal, wavelength)
        q_c_ideal = 2*pi*trapz(I_BB_cell*e_m_c_ideal, wavelength)
        q_v_ideal = 2*pi*trapz(I_BB_cell*e_m_v_ideal, wavelength)
        if theta_sun !=1:
            q_sun_ideal = trapz(e_sun_ideal*I_AM1_5, wavelength)
        else:
            q_sun_ideal = 0

        steady_state_test_ideal = q_m_ideal - q_atm_ideal - q_c_ideal - q_v_ideal - q_sun_ideal
        if steady_state_test_ideal > 0:
            T_m_ideal -= T_m_i

        T_m_ideal_list.append(T_m_ideal)
```

```
        q_m_ideal_list.append(q_m_ideal)
        q_atm_ideal_list.append(q_atm_ideal)
        q_c_ideal_list.append(q_c_ideal)
        q_v_ideal_list.append(q_v_ideal)
        q_sun_ideal_list.append(q_sun_ideal)
        stead_state_test_ideal_list.append(steady_state_test_ideal)

    # METASURFACE CASE SCENARIO lOOP
        #calculated using the emissivity from Lumerical data and mid latitude summer atmospheric transparency
    T_m_met_list = []
    q_m_met_list = []
    q_atm_met_list = []
    q_c_met_list = []
    q_v_met_list = []
    q_backside_met_list = []
    q_cond_met_list = []
    q_sun_met_list = []
    stead_state_test_met_list = []
    steady_state_test_met = 100  # (0 = steady state)
    while steady_state_test_met > 0:
        I_BB_m_met = zeros(wavelength.size)
        for i in range(wavelength.size):
            I_BB_m_met[i] = BB_Intensity(wavelength[i], T_m_met)

        q_m_met = 2*(2*pi*trapz(I_BB_m_met*e_m_all_met, wavelength))
        q_atm_met = 2*pi*trapz(t_ZnSe*I_BB_amb*e_atm_met, wavelength)
        q_c_met = 2*pi*trapz(I_BB_cell*e_m_c_met, wavelength)
        q_v_met = 2*pi*trapz(I_BB_cell*e_m_v_met, wavelength)
        q_backside_met = 2*pi*trapz(I_BB_cell*e_m_backside_met, wavelength)
        if theta_sun !=1:
            q_sun_met = trapz(e_sun_met*I_AM1_5, wavelength)
        else:
            q_sun_met = 0
        q_cond_met = x_cond*(2*e_tot_hem*5.67e-8*(T_cell**4-T_m_met**4))/(1-x_cond)

        steady_state_test_met = q_m_met - q_atm_met - q_c_met - q_v_met - q_backside_met - q_cond_met - q_sun_met
        if steady_state_test_met > 0:
            T_m_met -= T_m_i

        T_m_met_list.append(T_m_met)
        q_m_met_list.append(q_m_met)
```

```
        q_atm_met_list.append(q_atm_met)
        q_c_met_list.append(q_c_met)
        q_v_met_list.append(q_v_met)
        q_backside_met_list.append(q_backside)
        q_cond_met_list.append(q_cond_met)
        q_sun_met_list.append(q_sun_met)
        stead_state_test_met_list.append(steady_state_test_met)
        #print('T_m_met =', round(T_m_met, 2), 'K')

    #Printing heat Trasnfer rate values at steady state
    print("BEST")
    print("m (topside only)= ",q_m_best,", atm = ",q_atm_best,", c = ",q_c_best,", v = ",q_v_best," cond =
",q_cond_best," sun = ",q_sun_best," W/m^2")
    print("\nWORST")
    print("m = ",q_m_worst,", atm = ",q_atm_worst,", c = ",q_c_worst,", v = ",q_v_worst,", back = ",q_backside,", cond
= ",q_cond_worst," sun = ",q_sun_worst," W/m^2")
    print("\nIDEAL")
    print("m = ",q_m_ideal,", atm = ",q_atm_ideal,", c = ",q_c_ideal,", v = ",q_v_ideal," sun = ",q_sun_ideal,"
W/m^2")
    print("\nMETASURFACE")
    print("m = ",q_m_met,", atm = ",q_atm_met,", c = ",q_c_met,", v = ",q_v_met,", back = ",q_backside_met,", cond =
",q_cond_met," sun = ",q_sun_met," W/m^2")

    #Populating iterative HT rate value dataframes for each case
    worst_case_df = pd.DataFrame({"T_m": T_m_worst_list, "q_m": q_m_worst_list, "q_atm": q_atm_worst_list, "q_c":
q_c_worst_list, "q_v": q_v_worst_list, "q_back": q_backside_list, "q_cond": q_cond_worst_list, "q_sun":
q_sun_worst_list, "e_ss": stead_state_test_worst_list})
    best_case_df = pd.DataFrame({"T_m": T_m_best_list, "q_m": q_m_best_list, "q_atm": q_atm_best_list, "q_c":
q_c_best_list, "q_v": q_v_best_list, "q_cond": q_cond_best_list, "q_sun": q_sun_best_list, "e_ss":
stead_state_test_best_list})
    ideal_case_df = pd.DataFrame({"T_m": T_m_ideal_list, "q_m": q_m_ideal_list, "q_atm": q_atm_ideal_list, "q_c":
q_c_ideal_list, "q_v": q_v_ideal_list, "q_sun": q_sun_ideal_list, "e_ss": stead_state_test_ideal_list})
    met_case_df = pd.DataFrame({"T_m": T_m_met_list, "q_m": q_m_met_list, "q_atm": q_atm_met_list, "q_c":
q_c_met_list, "q_v": q_v_met_list, "q_back": q_backside_met_list, "q_cond": q_cond_met_list, "q_sun": q_sun_met_list,
"e_ss": stead_state_test_met_list})

    return T_m_best, T_m_worst, T_m_ideal, T_m_met, e_spect_hem_best, e_spect_hem_worst, e_ideal, e_spect_0_best,
e_spect_0_worst, e_spect_30_best, e_spect_30_worst, e_spect_60_best, e_spect_60_worst, e_tot_hem, A_met,
t_atm_mid_summer, t_atm_arc_winter, t_ZnSe, I_AM1_5, worst_case_df, best_case_df, ideal_case_df, met_case_df
```

```
## -- COMMAND -- ##
"""
Description: Calling T_m solution function to determine T_m upon radiative cooling for conitions given by the user
inputs
"""
#Calling T_m solution
T_m_best, T_m_worst, T_m_ideal, T_m_met, e_spect_hem_best, e_spect_hem_worst, e_ideal, e_spect_0_best,
e_spect_0_worst, e_spect_30_best, e_spect_30_worst, e_spect_60_best, e_spect_60_worst, e_tot_hem, A_met,
t_atm_mid_summer, t_atm_arc_winter, t_ZnSe, I_AM1_5, worst_case_df, best_case_df, ideal_case_df, met_case_df =
T_m_solution(wavelength, thickness, T_amb, T_cell, theta_v, theta_sun, index_4um, index_20um, index_5um)

## -- RESULTS -- ##
print('\n## -- SIMULATION PARAMETERS -- ##')
print('T_amb =',round(T_amb, 2),'K')
print('T_cell =',round(T_cell, 2),'K (Membrane assumed to start at this temperature)')
print('theta_v =',round(theta_v, 2),'rad')
print("\n## -- MEMBRANE TEMPERATURE RESULTS -- ##")
print('T_m_worst =',round(T_m_worst, 2),'K')
print('T_m_best =',round(T_m_best, 2),'K')
print('The temperature drop from T_cell is',round(T_cell - T_m_worst, 2),'to',round(T_cell - T_m_best, 2),'K')
print("\n## -- MEMBRANE HEMISPHERICAL EMISSIVITY RESULTS -- ##")
print('e_tot_hem =',round(e_tot_hem, 3))
print("\n## -- IDEAL TEMPERATURE RESULTS -- ##")
print('T_m_ideal =',round(T_m_ideal, 2),'K')
print('The temperature drop from T_cell is',round(T_cell - T_m_ideal, 2),'K')
print("\n## -- METASURFACE TEMPERATURE RESULTS -- ##")
print('T_m_met =',round(T_m_met, 2),'K')
print('The temperature drop from T_cell is',round(T_cell - T_m_met, 2),'K')


## -- PLOTS -- ##
"""
Description: Generating plots for thesis
"""
##Thesis plot settings
plot_width = 6.5 #in.
plot_font_name = "Arial"
plot_font_size = 12
plot_dpi = 300
matplotlib.rcParams.update({'font.size': plot_font_size})
matplotlib.rcParams['font.sans-serif'] = plot_font_name
```

```
##Emissivity Models Plot
fig, ax = plt.subplots(figsize=[plot_width, 2.6], dpi=plot_dpi)
ax.plot(wavelength*1e6, e_spect_hem_worst, color='m', label=u'$\u03B5_{\u03BB,m,thin-film}$')
ax.plot(wavelength*1e6, e_spect_hem_best, color='c', label=u'$\u03B5_{\u03BB,m,multilayer}$')
ax.set_xlabel(u"Wavelength (\u03bcm)")
ax.set_ylabel("Membrane Spectral\nHemispherical Emissivity")
ax.minorticks_on()
# Legend
lines, labels = ax.get_legend_handles_labels()
ax.legend(lines, labels, loc=1, frameon=False)
#save fig
plt.tight_layout()
plt.savefig(r"C:\Users\alexandre.bouchard\Downloads\emissivity_models.svg", format="svg")
plt.show()

##Emissivity of metasurface
fig, ax = plt.subplots(figsize=[plot_width, 2.6], dpi=plot_dpi)
ax.plot(wavelength*1e6, e_spect_hem_worst, color='m', label=u'$\u03B5_{\u03BB,m,thin-film}$')
ax.plot(wavelength*1e6, A_met, color='c', label= u"$\u03B5_{\u03BB,0\u00B0,metasurface}$")
ax.set_xlabel(u"Wavelength (\u03bcm)")
ax.set_ylabel("Emissivity")
ax.minorticks_on()
# Legend
lines, labels = ax.get_legend_handles_labels()
ax.legend(lines, labels, loc=1, frameon=False)
#save fig
plt.tight_layout()
plt.savefig(r"C:\Users\alexandre.bouchard\Downloads\emissivity_metasurface.svg", format="svg")
plt.show()

##Emissivity with atmospheric transparency plot
fig, ax = plt.subplots(figsize=[plot_width, 2.6], dpi=plot_dpi)
ax.plot(wavelength*1e6, t_atm_mid_summer, color='b', label=u'$\u03C4_{\u03BB,atm}$')
ax.fill_between(wavelength*1e6, t_atm_mid_summer, color='b', alpha=0.3)
ax.plot(wavelength*1e6, e_spect_hem_worst, color='m', label=u'$\u03B5_{\u03BB,SiN}$')
ax.set_xlabel(u"Wavelength (\u03bcm)")
ax.set_ylabel("Transmissivity/Emissivity")
ax.minorticks_on()
ax.set_xlim(6,20)
# Legend
```

```
lines, labels = ax.get_legend_handles_labels()
ax.legend(lines, labels, loc=1, frameon=False)
#save fig
plt.tight_layout()
plt.savefig(r"C:\Users\alexandre.bouchard\Downloads\emissivity_vs_atm_transparency.svg", format="svg")
plt.show()

##Emissivity directionality plot
fig, ax = plt.subplots(figsize=[plot_width, 3.6], dpi=plot_dpi)
ax.plot(wavelength*1e6, e_spect_hem_worst, color='m', label=u'$\u03B5_{\u03BB,m,thin-film}$')
ax.plot(wavelength*1e6, e_spect_0_worst, color='r', label= u"$\u03B5_{\u03BB,0\u00B0,m}$", linestyle = "dashed")
ax.plot(wavelength*1e6, e_spect_30_worst, color='b', label= u"$\u03B5_{\u03BB,30\u00B0,m}$", linestyle = "dashed")
ax.plot(wavelength*1e6, e_spect_60_worst, color='g', label= u"$\u03B5_{\u03BB,60\u00B0,m}$", linestyle = "dashed")
ax.set_xlabel(u"Wavelength (\u03bcm)")
ax.set_ylabel(u"SiN Emissivity")
ax.minorticks_on()
ax.set_xlim(3,50)
# Legend
lines, labels = ax.get_legend_handles_labels()
ax.legend(lines, labels, loc=1, frameon=False)
#save fig
plt.tight_layout()
plt.savefig(r"C:\Users\alexandre.bouchard\Downloads\e_directional.svg", format="svg")
plt.show()

#Atmoshpheric transparency comparison plot with ZnSe transparency
fig, ax = plt.subplots(figsize=[plot_width, 3.6], dpi=plot_dpi)
ax.plot(wavelength*1e6, t_atm_mid_summer, color='b', label=u'$\u03C4_{\u03BB,atm,mid-latitude}$')
ax.plot(wavelength*1e6, t_atm_arc_winter, color='c', label=u'$\u03C4_{\u03BB,atm,sub-arctic}$')
ax.plot(wavelength*1e6, t_ZnSe, color='y', label=u'$\u03C4_{\u03BB,v}$')
ax.set_xlabel(u"Wavelength (\u03bcm)")
ax.set_ylabel("Transmissivity")
ax.minorticks_on()
ax.set_xlim(0,20)
# Legend
lines, labels = ax.get_legend_handles_labels()
ax.legend(lines, labels, loc='upper center', bbox_to_anchor=(0.5, 1.2), ncol=4, columnspacing=1, frameon=False)
#save fig
plt.tight_layout()
plt.savefig(r"C:\Users\alexandre.bouchard\Downloads\t_atm_t_ZnSe_comparison.svg", format="svg")
plt.show()
```

```
##Solar Am1.5 spectrum
fig, ax = plt.subplots(figsize=[plot_width, 2.6], dpi=plot_dpi)
ax.plot(wavelength*1e6, I_AM1_5*1e-6, color='r', label='$I_{AM1.5}$')
ax.fill_between(wavelength*1e6, I_AM1_5*1e-6, color='r', alpha=0.3)
ax.set_xlabel(u"Wavelength (\u03bcm)")
ax.set_ylabel(u"Solar Intensity (Wm$^{-2}$\u03bcm$^{-1}$)")
ax.minorticks_on()
ax.set_xlim(0,4)
# Legend
lines, labels = ax.get_legend_handles_labels()
ax.legend(lines, labels, loc=1, frameon=False)
#save fig
plt.tight_layout()
plt.savefig(r"C:\Users\alexandre.bouchard\Downloads\IAM1_5.svg", format="svg")
plt.show()

##Pie chart - weight of each heat trasnfer rate against q_m at steady state - Thin Film Model & Mid-lat t_atm
worst_q_atm = worst_case_df.iloc[-1]['q_atm']
worst_q_chtop = worst_case_df.iloc[-1]['q_c']+worst_case_df.iloc[-1]['q_v']
worst_q_back =worst_case_df.iloc[-1]['q_back']
worst_q_sun =worst_case_df.iloc[-1]['q_sun']
worst_q_cond =worst_case_df.iloc[-1]['q_cond']
worst_q_m = worst_case_df.iloc[-1]['q_m']
#calculating percentages
p_back = worst_q_back/worst_q_m*100
p_top = worst_q_chtop/worst_q_m*100
p_atm = worst_q_atm/worst_q_m*100
p_cond = worst_q_cond/worst_q_m*100
p_sun = worst_q_sun/worst_q_m*100
#Defining Labels
l_back = "$q_{ch,back}$, "+str(int(round(p_back,0)))+" %"
l_top = "$q_{ch,top}$, "+str(int(round(p_top, 0)))+" %"
l_atm = "$q_{atm}$, "+str(int(round(p_atm,0)))+" %"
l_cond = "$q_{cond}$, "+str(int(round(p_cond,0)))+" %"
l_sun = "$q_{sun}$, "+str(int(round(p_sun,0)))+" %"
#Defining table values
worst_labels = [l_back, l_top, l_atm, l_cond, l_sun]
worst_slices = [worst_q_back, worst_q_chtop, worst_q_atm, worst_q_cond, worst_q_sun]
#plot
fig, ax = plt.subplots(figsize=[plot_width, 2.5], dpi=plot_dpi)
```

```
ax.pie(worst_slices, wedgeprops={'linewidth': 0.5,'edgecolor': 'black'})
ax.axis('equal')
plt.title("Worst-Case")
#Legend
plt.legend(labels=worst_labels, loc='center left', bbox_to_anchor=(1,0.5), frameon=False)
#save fig
plt.tight_layout()
plt.savefig(r"C:\Users\alexandre.bouchard\Downloads\pie_chart_thin_film.svg", format="svg")
plt.savefig(r"C:\Users\alexandre.bouchard\Downloads\pie_chart_thin_film.pdf", format="pdf")
plt.show()

##Pie chart - weight of each heat trasnfer rate against q_m at steady state - Multilayer Model & sub-arct t_atm
best_q_atm = best_case_df.iloc[-1]['q_atm']
best_q_chtop = best_case_df.iloc[-1]['q_c']+best_case_df.iloc[-1]['q_v']
best_q_sun =best_case_df.iloc[-1]['q_sun']
best_q_cond =best_case_df.iloc[-1]['q_cond']
best_q_m = best_case_df.iloc[-1]['q_m']
#calculating percentages
p_top = best_q_chtop/best_q_m*100
p_atm = best_q_atm/best_q_m*100
p_cond = best_q_cond/best_q_m*100
p_sun = best_q_sun/best_q_m*100
#Defining Labels
l_top = "$q_{ch,top}$, "+str(int(round(p_top, 0)))+" %"
l_atm = "$q_{atm}$, "+str(int(round(p_atm,0)))+" %"
l_cond = "$q_{cond}$, "+str(int(round(p_cond,0)))+" %"
l_sun = "$q_{sun}$, "+str(int(round(p_sun,0)))+" %"
#Defining table values
best_labels = [l_top, l_atm, l_cond, l_sun]
best_slices = [best_q_chtop, best_q_atm, best_q_cond, best_q_sun]
#plot
fig, ax = plt.subplots(figsize=[plot_width, 2.5], dpi=plot_dpi)
ax.pie(best_slices, wedgeprops={'linewidth': 0.5,'edgecolor': 'black'})
ax.axis('equal')
plt.title("Best-Case")
#Legend
plt.legend(labels=best_labels, loc='center left', bbox_to_anchor=(1,0.5), frameon=False)
#save fig
plt.tight_layout()
plt.savefig(r"C:\Users\alexandre.bouchard\Downloads\pie_chart_thin_film.svg", format="svg")
plt.show()
```

K. PYTHON SCRIPT: EXPERIMENTAL DATA ANALYSIS

```
"""
Description: This script determines the membrane resonance frequency shift, membrane temperature shift, and silicon
frame temperature shift from a radiative cooling experiment (from Waveforms and LabJack data).
Author: Alexandre Bouchard
Date: October 20, 2022
"""

## -- IMPORTED LIBRARIES -- ##
from numpy import genfromtxt, zeros, array, round, sqrt
from datetime import datetime, timedelta
import matplotlib.pyplot as plt
import matplotlib
import matplotlib.transforms as mtransforms


## -- USER INPUTS -- ##
"""
Description: User inputs to the Waveforms and LabJack data folders/files for a single experiment.
NOTE:
    The printed resonance frequency shift result is only valid once the right file number range is selected. You must
select the file number range that corresponds to a single viewport uncovering (use the first plot to determine this).
"""
folder_link_waveforms = r"C:\Users\alexandre.bouchard\2022-08-27 Day\WF_peaks\Test 9" #Link to Waveforms folder
start_file = 1 #Waveforms data file number to start at
end_file = 50 #Waveforms data file number to end at
approx_res_freq = 18.0 #Approximate membrane resonance frequency before radiative cooling [kHz]
file_type = "txt" #csv or txt
file_link_labjack = r"C:\Users\alexandre.bouchard\2022-08-27 Day\LJ_temp\data_0.dat" #Link to LabJack file


## -- WAVEFORMS DATA -- ##
"""
Description: Creating array (membrane resonance frequency vs. time) from all Waveforms data files.
Important variables:
    res_frequency - resonance frequency of membrane
    WF_time_arr_H_M_S_uS - time in H:M:S:uS corresponding to resonance frequency data
    WF_time_arr_H_M_S - time in H:M:S corresponding to resonance frequency data
```

```
"""
num_files = end_file - start_file
table_waveforms = zeros(num_files)
res_frequency = zeros(num_files)
WF_file_num = zeros(num_files)
WF_time_H_M_S_uS = []
WF_time_H_M_S = []
print("Printing Resonance Frequency (kHz) per File Number:")
for i in range(num_files):
    ## Defining acquisition file number for each loop (since file numbers start at 1 and have three digits)
    if i+start_file<9:
        acq_number = f"000{i+start_file}"
    elif i+start_file>9 and i+start_file<=99:
        acq_number = f"00{i+start_file}"
    elif i+start_file >99 and i+start_file<999:
        acq_number = f"0{i+start_file}"
    WF_file_num[i] = acq_number

    #Get Resonance Frequency of Acquisition File
    for j in range(6,8):
        text = open(f"{folder_link_waveforms}\\acq{acq_number}.{file_type}")
        freq_str = text.readlines()[j]
        res_freq_str = freq_str.split()[0]
        if abs(float(res_freq_str)-approx_res_freq) < 2:
            res_frequency[i] = float(res_freq_str)
            print(acq_number, res_frequency[i])
            break
        else:
            print("Peak Gone")

    #Getting Time of Acquisition File in H:M:S:uS
    text = open(f"{folder_link_waveforms}\\acq{acq_number}.{file_type}")
    time_str = text.readlines()[3]
    time_value_list = time_str.split()
    time_value_datetime = datetime.strptime(time_value_list[2]+' '+time_value_list[3], "%Y-%m-%d %H:%M:%S.%f")
    time_value_string = datetime.strftime(time_value_datetime, "%H:%M:%S:%f")
    WF_time_H_M_S_uS.append(time_value_string)
    #Getting time of acquisition file in H:M:S (for matching of LabJack later in script)
    time_value_string_second = datetime.strftime(time_value_datetime, "%H:%M:%S")
    WF_time_H_M_S.append(time_value_string_second)
```

```
WF_time_arr_H_M_S_uS = array(WF_time_H_M_S_uS)
WF_time_arr_H_M_S = array(WF_time_H_M_S)


## -- LABJACK DATA -- ##
"""
Description: Creating array (Silicon/Ambient/Chamber Temperature vs. time) from all LabJack data files.
Important variables:
    LJ_amb_temp - Ambient temperature (from temperature probe; AIN0) [Celsius]
    LJ_cell_temp - Chamber temperature (from thermistor; AIN2) [Celsius]
    LJ_Si_temp - Silicon frame temperature (from RTD; AIN5) [Celsius]
    LJ_time_H_M_S - time in H:M:S corresponding to temperature data
"""
table_labjack = genfromtxt(file_link_labjack, delimiter='\t')
LJ_amb_temp = table_labjack[:,1]
LJ_cell_temp = table_labjack[:,3]
LJ_Si_temp = table_labjack[:,4]
labjack_excel_time = table_labjack[:,0]/86400+1462-0.1666667
LJ_time_H_M_S = []
#Converting Excel time to Date Format
def from_excel_ordinal(ordinal, epoch=datetime(1900, 1, 1)):
    if ordinal > 59:
        ordinal -= 1  # Excel leap year bug, 1900 is not a leap year!
    inDays = int(ordinal)
    frac = ordinal - inDays
    inSecs = int(round(frac * 86400.0))

    return epoch + timedelta(days=inDays - 1, seconds=inSecs) # epoch is day 1

for i in range(labjack_excel_time.size):
    LJ_datetime = from_excel_ordinal(labjack_excel_time[i])
    LJ_datetime_string = datetime.strftime(LJ_datetime, "%H:%M:%S")  # %H = hour; %M = minute; %S = second; %f =
microsecond
    LJ_time_H_M_S.append(LJ_datetime_string)


## -- MATCHING WAVEFORMS AND LABJACK DATA -- ##
"""
Description: Matching the LabJack and Waveforms data to the same time stamp.
Important variables:
    amb_temperature - Ambient temperature data matching time stamp of WF time [Celsius]
```

```
    cell_temperature - Chamber temperature matching time stamp of WF time [Celsius]
    Si_temperature - Silicon frame temperature matching time stamp of WF time [Celsius]
    WF_seconds_elapsed - Time elapsed during experiment
"""
#Matching LJ data to WF time data
amb_temperature = zeros(WF_time_arr_H_M_S.size)
cell_temperature = zeros(WF_time_arr_H_M_S.size)
Si_temperature = zeros(WF_time_arr_H_M_S.size)
for i in range(amb_temperature.size):
    temperature_index = LJ_time_H_M_S.index(WF_time_arr_H_M_S[i])
    amb_temperature[i] = LJ_amb_temp[temperature_index]
    cell_temperature[i] = LJ_cell_temp[temperature_index]
    Si_temperature[i] = LJ_Si_temp[temperature_index]

#Creating an array of the time elapsed during the experiment (instead of absolute time)
WF_seconds_elapsed = zeros(WF_time_arr_H_M_S_uS.size)
time_0 = datetime.strptime(WF_time_arr_H_M_S_uS[0], "%H:%M:%S:%f")
for i in range(WF_time_arr_H_M_S_uS.size):
    time_loop = datetime.strptime(WF_time_arr_H_M_S_uS[i], "%H:%M:%S:%f")
    time_difference = time_loop - time_0
    WF_seconds_elapsed[i] = time_difference.total_seconds()


## -- DETERMINE MINIMUM AND MAXIMUM FREQUENCY -- ##
"""
Description: The minimum and maximum frequency is determined to calculate the frequency shift
Important variables:
    max_freq - Resonance frequency after radiative cooling [kHz]
    min_freq - Resonance frequency before radiative cooling [kHz]
    freq_shift - Resonance frequency shift (max_freq - min_freq) [Hz]
NOTE:
    freq_shift is only once the right file number range is selected. You must select the file number range that
corresponds to a single viewport uncovering.
"""
min_freq_index = 0
max_freq_index = res_frequency.size-1
max_freq = res_frequency[max_freq_index]
min_freq = res_frequency[min_freq_index]
freq_shift = abs(max_freq - min_freq)*1000 #Hz
```

```
## -- CALCULATING TEMPERATURE SHIFT FROM RESONANCE FREQUENCY SHIFT -- ##
"""
Description: Calculating the temperature shift from the resonance frequency shift.
Important variables:
    temp_shift (float) - Membrane temperature shift [K]
    u_T (float) - Membrane temperature shift uncertainty [K]
    membrane_temp_shift (array) - array of membrane temperature shift [K] per time stamp
    uncertainty (array) - array of membrane temperature shift uncertainty [K] per time stamp
NOTE:
    temp_shift is only once the right file number range is selected. You must select the file number range that
corresponds to a single viewport uncovering.
"""
##Material Properties & Variables
m = 1 #Modeshape antinode m
n = 1 #Modeshape antinode n
f_0 = min_freq*1000 #Initial resonance frequency (before shift) [Hz]
error = 0.12182581676593424 #Error calculated from B_reff and (n,j), see Thesis
rho = 3000 #Density [kg/m^3]
L = 0.006 #side length of membrane [m]
alpha = 2.22e-6 #thermal expansion coeff [K^-1]
E = 290e9 #Young's modulus [Pa]
mu = 0.25 #Poisson ratio
sigma_0 = (4*f_0**2*rho*L**2)/(m**2+n**2) #Tensile stress [Pa]

##Temperature Shift Calculation [K]
temp_shift = (freq_shift/((alpha*E*f_0)/(2*sigma_0*(1-mu))))*(1+error)

##Temperature Shift Uncertainty Calculation [K]
#Material Property Uncertainties
u_alpha = 0.07e-6 #uncertainty of thermal expansion coefficient
u_L = 15e-6 #uncertainty of side length
u_E = 90e9 #uncertainty of Young's Modulus
u_mu = 0.03 #uncertainty of Poisson ratio
u_rho = 100 #uncertainty of density
#Partial derivatives in propagation uncertainty rule
T_diff_alpha = (8*(error+1)*L**2*freq_shift*rho*(mu-1)*f_0)/(E*(m**2+n**2)*alpha**2)
T_diff_L = (16*(error+1)*freq_shift*rho*(1-mu)*f_0*L)/(E*(m**2+n**2)*alpha)
T_diff_E = (8*(error+1)*L**2*freq_shift*rho*(mu-1)*f_0)/(E**2*(m**2+n**2)*alpha)
T_diff_mu = (8*(error+1)*L**2*freq_shift*rho*f_0)/(E*(m**2+n**2)*alpha)
T_diff_rho = (8*(error+1)*L**2*freq_shift*(1-mu)*f_0)/(E*(m**2+n**2)*alpha)
#Propagation uncertainty rule
```

```
u_T = sqrt((T_diff_alpha*u_alpha)**2+(T_diff_L*u_L)**2+(T_diff_E*u_E)**2+(T_diff_mu*u_mu)**2+(T_diff_rho*u_rho)**2)
#Populating incremental temperature shift array for plot
membrane_temp_shift = zeros(WF_seconds_elapsed.size)
uncertainty = zeros(WF_seconds_elapsed.size)
for i in range(WF_seconds_elapsed.size):
    freq_shift = (res_frequency[i]-min_freq)*1000
    #temp shift
    membrane_temp_shift[i] = (freq_shift/((alpha*E*f_0)/(2*sigma_0*(1-mu))))*(1+error) #K
    #uncertainty
    T_diff_alpha = (8*(error+1)*L**2*freq_shift*rho*(mu-1)*f_0)/(E*(m**2+n**2)*alpha**2)
    T_diff_L = (16*(error+1)*freq_shift*rho*(1-mu)*f_0*L)/(E*(m**2+n**2)*alpha)
    T_diff_E = (8*(error+1)*L**2*freq_shift*rho*(mu-1)*f_0)/(E**2*(m**2+n**2)*alpha)
    T_diff_mu = (8*(error+1)*L**2*freq_shift*rho*f_0)/(E*(m**2+n**2)*alpha)
    T_diff_rho = (8*(error+1)*L**2*freq_shift*(1-mu)*f_0)/(E*(m**2+n**2)*alpha)
    uncertainty[i] =
sqrt((T_diff_alpha*u_alpha)**2+(T_diff_L*u_L)**2+(T_diff_E*u_E)**2+(T_diff_mu*u_mu)**2+(T_diff_rho*u_rho)**2)


## -- RESULTS -- ##
print("\nRESULTS:")
print("Membrane resonance frequency shift = ", round(freq_shift,2), "Hz")
print("Membrane temperature shift = ", round(temp_shift,2), "K")
print("Membrane temperature shift Uncertainty = ", round(u_T,2), "K")


## -- PLOTS -- ##
"""
Description: Generating plots
"""
##Thesis plot settings
plot_width = 6.5 #in.
plot_font_name = "Arial"
plot_font_size = 12
plot_dpi = 300
matplotlib.rcParams.update({'font.size': plot_font_size})
matplotlib.rcParams['font.sans-serif'] = plot_font_name

#Resonance frequency versus file number (USE THIS PLOT TO DETERMINE THE RIGHT FILE NUMBER RANGE)
fig, ax = plt.subplots(figsize=[plot_width, 4], dpi=plot_dpi)
ax.plot(WF_file_num, res_frequency, label='Resonance Frequency [kHz]', color='r')
ax.annotate("Maximum Frequency", (max_freq_index, max_freq), arrowprops=dict(facecolor='grey'))
```

```
ax.annotate("Minimum Frequency", (min_freq_index, min_freq), arrowprops=dict(facecolor='grey'))
ax.set_ylabel('Resonance Frequency (kHz)', color='r')
ax.set_xlabel('WF File Number')
plt.title("USE THIS PLOT TO:\n1) Determine file number range for a single viewport uncovering\n2) Validate the max and
min frequency indices")
ax.minorticks_on()
plt.tight_layout()
plt.show()

#Membrane resonance frequency/temperature & silicon temprature vs time elapsed
fig, ax = plt.subplots(figsize=[plot_width, 2.85], dpi=plot_dpi)
ax.plot(WF_seconds_elapsed, 0-membrane_temp_shift, label=r'$\overline{ΔT_{m}}$ (membrane)', color='b',
linestyle='dashed')
ax.plot(WF_seconds_elapsed, Si_temperature-Si_temperature[0], label='$ΔT_{Si}$ (frame)', color='b')
ax.set_xlabel('Time Elapsed (s)')
ax.set_ylabel("Temperature Shift (K)", color='b')
ax.tick_params(axis='y', which='both', colors='b')
# Twin Axes for temp shift
ax2 = ax.twinx()
ax2.plot(WF_seconds_elapsed, res_frequency, label='$f_{m}$ (membrane)', color='m')
ax2.set_ylabel('Resonance Frequency (kHz)', color='m')
ax2.tick_params(axis='y', which='both', colors='m')
#Viewport line
plt.axvline(0.7, 0, 1, linestyle='dashed', color='k', alpha=0.3)
trans = mtransforms.blended_transform_factory(ax2.transData, ax2.transAxes)
ax.fill_between(WF_seconds_elapsed, 0, 1, where=WF_seconds_elapsed <= 0.78, facecolor='k', alpha=0.1, transform=trans)
# Display
ax.minorticks_on()
ax2.minorticks_on()
#ax11.set_ylim(-7,1.65)
plt.tight_layout()
# Legend
lines, labels = ax.get_legend_handles_labels()
lines2, labels2 = ax2.get_legend_handles_labels()
ax2.legend(lines+lines2, labels+labels2, loc=5, frameon=False)
plt.savefig(r"C:\Users\alexandre.bouchard\Downloads\Si vs SiN temperature shift.svg", format="svg")
plt.show()

#Ambient/Si/Chamber temperature during time elapsed
fig, ax = plt.subplots(figsize=[plot_width, 3], dpi=plot_dpi)
ax.plot(WF_seconds_elapsed, amb_temperature, label='$T_{amb}$', color='r')
```

```
ax.plot(WF_seconds_elapsed, cell_temperature, label='$T_{ch}$', color='g')
ax.plot(WF_seconds_elapsed, Si_temperature, label='$T_{Si}$', color='b')
ax.set_ylabel('Temperature (C)')
ax.set_xlabel('Time Elapsed (s)')
# Display
ax.minorticks_on()
# Legend
lines, labels = ax.get_legend_handles_labels()
ax.legend(lines, labels, loc=1, frameon=False)
plt.show()

## -- PLOT Ambient & Cell Temperature -- ##
fig3, ax31 = plt.subplots(figsize=[plot_width, 3], dpi=plot_dpi)
ax31.plot(LJ_time_H_M_S, LJ_amb_temp, label='$T_{amb}$', color='r')
ax31.plot(LJ_time_H_M_S, LJ_cell_temp, label='$T_{ch}$', color='g')
ax31.plot(LJ_time_H_M_S, LJ_Si_temp, label='$T_{Si}$', color='b')
ax31.set_xticks(LJ_time_H_M_S[::120])
ax31.set_xticklabels(LJ_time_H_M_S[::120], rotation=45)
ax31.set_ylabel('Temperature (C)')
ax31.set_xlabel('Time Elapsed (s)')
ax31.minorticks_on()
# Legend
lines31, labels31 = ax31.get_legend_handles_labels()
ax31.legend(lines31, labels31, loc=1, frameon=False)
plt.show()
```