COMPUTATIONAL AID FOR DESIGNING PV CANOPY FOR SOLAR-POWERED TRANSIT

A Project Presented to

The Faculty of the Department of

Mechanical Engineering

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Mechanical Engineering

by

Jackson B. Fogelquist

May 2019

SAN JOSÉ STATE UNIVERSITY

The Undersigned Committee Approves

Computational Aid for Designing PV Canopy for Solar-Powered Transit

of

Jackson B. Fogelquist

APPROVED FOR THE DEPARTMENT OF MECHANICAL ENGINEERING

Dr. Burford Furman, Committee Chair	Date
Dr. Ping Hsu, Committee Member	Date
Mr. Ron Swenson, Committee Member International Institute of Sustainable Transportation	Date

ABSTRACT

Computational Aid for Designing PV Canopy for Solar-Powered Transit

By Jackson B. Fogelquist

Solar-powered transit is an emerging sector with the potential to make significant reductions in greenhouse gas emissions and urban congestion. One of the largest challenges is the lack of available land for photovoltaic (PV) arrays in urban environments. The solution from the Spartan Superway automated transit network (ATN) research effort is to mount PV modules on canopies above the guideway structure. This poses new constraints on module orientation and canopy design that prevent these systems from being accurately modeled with existing PV design software. Thus, it was necessary to develop a computational aid for the design of PV canopies for transit applications.

The computational aid loads a user-defined transit route, models the hourly solar irradiance, estimates the energy demand of the transit system, determines the hourly shaded regions of the route, optimizes the size and shape of the PV canopy, and simulates the hourly electricity generation. The accuracy of the model was validated through a comparison of simulation results from two industry-accepted PV modeling programs; the maximum difference in annual electricity output was 0.21%. Several simulation studies were conducted to understand the effects of array profile on irradiance collection. Most notably, the collected irradiance of curved arrays was analytically and experimentally observed to decrease exponentially with a linear increase in the angle between modules. An angle of 4° between modules was recommended as an acceptable compromise between array aesthetics and performance. Finally, the computational aid was used to design the PV canopy for a proposed solar-powered ATN route that links the north and south campuses of San José State University. The optimized canopy was only 6.2 meters wide and supplied twice the projected amount of energy required to operate the transit system at maximum throughput. It is thus feasible to power transit systems with canopies of PV modules.

ACKNOWLEDGEMENTS

I would like to thank my Committee Chair, Dr. Burford Furman, for suggesting this project; it has been a fascinating topic to explore. Dr. Furman was always welcoming, accessible, and eager to help. He worked hard to keep the project moving forward and ensured that I had all the resources I needed. I am grateful for his insightful advising—both project-related and beyond.

I would like to thank my committee members, Dr. Ping Hsu and Mr. Ron Swenson, for their commitment to the project. Dr. Hsu offered particularly grounding advice that helped keep the project focused on the objectives. His advice regarding power electronics associated with PV systems was very helpful. Mr. Swenson was encouraging and motivational, keeping in sight the big picture of working toward a future of solar-powered transit. I appreciate his willingness to share his vast experience in the fields of solar energy and personal rapid transit engineering.

I would like to thank my solar energy professor, Dr. Igor Tyukhov, for his enthusiasm toward this project. He shared several useful resources with me and offered helpful advice about PV data acquisition techniques and cell characterization methods. I would also like to thank Dr. Hilary Nixon and Mr. Richard Kos of the Urban and Regional Planning Department for helping me to attain building height data for the shading analysis algorithm.

Finally, I would like to thank my mother, Donna; father, Larry; brother, Spencer; and girlfriend, Kalika for their endless support, patience, and encouragement. A special thanks goes to Spencer, for letting me use his computer to test the code when I was unable to use my laptop.

TABLE OF CONTENTS

NOMENCLATURE

PV	-	Photovoltaic
ATN	-	Automated Transit Network
DNI	-	Direct Normal Irradiance
DHI	-	Diffuse Horizontal Irradiance
GHI	-	Global Horizontal Irradiance
SAM	-	System Advisory Model
NREL	-	National Renewable Energy Laboratory
NSRDB	-	National Solar Radiation Data Base
OPV	-	Organic Photovoltaic
API	-	Application Program Interface
KML	-	Keyhole Markup Language
USGS	-	United States Geological Survey
CSV	-	Comma-Separated Values
TMY	-	Typical Meteorological Year
SPA	-	Solar Position Algorithm
SPTM	-	SAM Physical Trough Model
PFTSW	-	Power from the Sun Woolf Model
PFTSL	-	Power from the Sun Lamm Model
LST	-	Local Standard Time
TST	-	True Solar Time
LC	-	Longitude Correction
EOT	-	Equation of Time
POA	-	Plane of Array
HDKR	-	Hay, Davies, Klucher, Reindl (diffuse POA model)
PCS	-	Projected Coordinate System
GCS	-	Geographic Coordinate System
NOCT	-	Nominal Operating Cell Temperature
CEC	-	California Energy Commission
STC	-	Standard Test Conditions
I-V	-	Current-Voltage

VARIABLES

β	-	Tilt angle
A	-	Azimuth angle, Area
$\mathbf{n}_M$	-	Module normal vector
γ	-	Angle of incidence
γ'	-	Angle of refraction
ψ	-	Longitude
ϕ	-	Latitude
Θ	-	Angular distance
L	-	Length
R	-	Radius, Resistance
θ	-	Leg bearing
Φ	-	Incline angle
h	-	Elevation
$\mathbf{L}$	-	Leg vector
$\mathbf{v}$	-	Orthogonal leg vector
$\mathbf{SL_1}$	-	Station-leg vector
d	-	Distance
α	-	Elevation angle, Temperature coefficient
Z	-	Zenith angle
N	-	Quantity
ω	-	Hour angle
δ	-	Declination angle
G	-	Irradiance
ε	-	Sky clearness parameter
AM	-	Air mass ratio
Δ	-	Sky brightness parameter
D	-	Component of diffuse irradiance
ρ	-	Albedo, Density
k	-	Correction factor, Boltzmann constant

t	-	Time, Thickness
v	-	Velocity
a	-	Acceleration, Ideality factor
J	-	Jerk
η	-	Efficiency
m	-	Mass
C	-	Empirical coefficient
g	-	Acceleration due to gravity
P	-	Power
X	-	PCS abscissa
Y	-	PCS ordinate
SR	-	Shaded ratio
i	-	Index
n	-	Index of refraction
K	-	Extinction coefficient
$\tau\alpha$	-	Transmittance-absorptance product
I	-	Current
V	-	Voltage
T	-	Temperature
E	-	Energy
x	-	Cartesian abscissa
y	-	Cartesian ordinate
z	-	Cartesian applicate
FOS	-	Factor of Safety

LIST OF TABLES

LIST OF FIGURES

1.0 INTRODUCTION

The current state of the global transportation sector is energy intensive, dangerous, and inefficient. It is energy intensive in that it comprises 25% of the total world energy consumption [1]. This statistic rises to 28% within the United States, with the primary energy source being petroleum [2]. The combustion of petroleum creates greenhouse gases like carbon dioxide and methane, which trap heat in the atmosphere and contribute to global warming [3]. Transportation emissions represented 28% of the United States greenhouse gas production in 2016, with 60% of these emissions coming from light-duty passenger vehicles [4]. Besides the environmental impact, transportation emissions have adverse health effects. An example is in the rapid rise of passenger cars in China, illustrated by the doubling of cars in Beijing to 5 million over a five-year span [2]. China subsequently contained 16 of the 20 most air-polluted cities in the world, with one-third of the country's urban residents exposed to unsafe air-pollution levels [2]. Aside from pollution, car-based transportation is susceptible to lethal accidents caused by small distractions. Road traffic crashes caused 1.25 million global fatalities in 2013, becoming the primary cause of death among the 15- to 29-year age group [5]. Finally, urban congestion makes car-based transportation inefficient, as the average U.S. driver spends 443 hours driving per year, 100 of which are in bumper-to-bumper traffic [2].

The global effort to reduce greenhouse gas emissions has faced many obstacles in the transportation sector. While electric vehicles are gaining popularity, alternative modes of transportation are being explored that shift the current car-based paradigm. Examples are electric vertical take-off and landing planes [6], vacuum trains [7], and automated transit networks (ATNs) [8]. The Spartan Superway is a research and

development program at San José State University that is working to design an ATN powered entirely by solar photovoltaic (PV) modules [9]. The modules are to be installed on canopies above the guideway, which pose constraints on module orientation. It is important to optimize the orientation of the modules to maximize electricity generation and economic payback. This can be done by developing a computational aid that simulates the hourly electricity generation and optimizes the module orientations within the transit application constraints. The computational aid is developed in the context of the Spartan Superway ATN research effort, though its application can be extended to vacuum trains, light rail, heavy rail, and other solar-powered transit systems with PV canopies above the route.

The subsequent literature review discusses the current state of solar-powered transit, technologies involved, and associated challenges. This information is used to derive a detailed problem statement, with project objectives presented afterward.

1.1 Literature Review

The following literature review provides relevant background information about photovoltaic (PV) systems. The current state of solar-powered transit is then discussed, followed by the characteristics of ATNs. Next is a summary of applicable past work from the Spartan Superway program. The final entry is an overview of PV modeling with mention of popular simulation programs.

1.1.1 PV Systems

The largest energy resource is the sun, which annually supplies 1.5 x 10^{17} kWh to the landmass of Earth—near 6,000 times the 2009 U.S. energy usage [10]. The sun is renewable in that it is expected to survive for approximately 4 billion more years [11]. Solar radiant energy is termed *irradiance* and is represented as a quantity of energy per time per area (W/m^2). Irradiance can be directly converted to electricity through a solid-state process called PV conversion. The subsequent PV information is divided into eight topics: PV conversion, decline of cost, limitations, configurations, components of solar irradiance, optimization of module orientation, shading, and cell technologies.

1.1.1.1 PV Conversion

PV conversion is a solid-state process in which irradiance is converted directly to electricity. Most solar cells are made of a semiconductor material such as silicon, which is positively (p) and negatively (n) doped to create a pn junction [12]. The pn junction is formed through diffusion, in which electrons from the n-type side diffuse to occupy holes in the p-type side, and holes from the p-type side diffuse to the electrons in the n-type side [12]. This diffusion process leaves behind a lattice of positively ionized dopants in the n-type side and negatively ionized dopants in the p-type side, which establish a built-in electric field across the junction [12]. An incoming photon with sufficient energy can separate a single electron from its associated atom, creating an electron-hole pair [10]. If this electron-hole pair is produced near the junction region, it may diffuse into the junction before the electron and hole recombine [12]. If this occurs, the built-in electric field will accelerate the hole to the p-type side and the electron to the n-type side [12].

This forces a higher concentration of charge carriers along the edges of the junction, which diffuse toward the contacts as direct current when the circuit is closed [12]. The absorption of photons is therefore the driving mechanism for electricity generation in PV cells.

1.1.1.2 Decline of Cost

The first PV solar cell was originally developed for space applications in 1954, made possible by the advent of the bipolar transistor in 1949 [12]. Advances in the semiconductor industry for the electronics market have exponentially reduced the cost of PV module production, from $101.05/W in 1975 to $0.37/W in 2017 [13]. The corresponding average unsubsidized cost of utility-scale PV electricity is $49.5/MWh—79% less than that of reciprocating diesel engines, 51% less than that of coal power plants, and 18% less than that of gas-combined-cycle power plants [14]. PV conversion does not produce emissions and cells have a lifetime energy production of over 10 times the manufacturing energy consumption [12].

1.1.1.3 Limitations

Despite the economic and environmental benefits of PV conversion over traditional methods of electricity generation, there are three core limitations. The first is the intermittency caused by the daily rotation of the earth, annual declination cycle, and weather variation [10]. The daily intermittency strains the grid in California with a midday period of PV overproduction, followed by a sharp decrease in PV production during peak demand at sunset [15]. An obvious but challenging solution is to store the

excess energy during times of overproduction for use during peak demand. The second limitation is that solar irradiance has a low flow of energy per unit area [10]. This requires large collection areas to generate useful amounts of electricity, which can be scarce or expensive in urban regions. The third limitation is that population centers are generally far from regions with the best solar resource, which are deserts at latitudes within 25° of the equator [10]. This poses transmission challenges in bringing the generated electricity to where it is used.

1.1.1.4 Configurations

Typical PV cells produce less than 5 W at 0.5 V_{DC}, converting less than 22% of the incident irradiance into electricity [12]. To increase the voltage, cells are connected in series to form a module, in which they are encased by a sturdy housing for protection from mechanical damage and water corrosion [16]. Larger PV systems are often created by wiring numerous modules into an array. These modules are usually configured in both series and parallel to achieve the desired voltage and current output [12].

PV canopies are elevated PV arrays that are designed to provide shade and shelter [17]. They are commonly built above parking lots, around the perimeter of buildings, and over transit stations. PV canopies have two benefits over traditional large-scale arrays: they do not compete with other land use and they are located where the generated electricity is used, avoiding transmission losses [2]. Since these canopies are usually in public places, it is important for them to be aesthetically pleasing [12]. They must also be structurally sound throughout various loading modes like module weight, strong winds, and possible accumulation of snow [12]. Figure 1 shows an example of a PV

canopy over a parking lot, with a top view of the PV modules (*a*) and a ground view of the structure beneath (*b*). The canopy is slightly curved for aesthetic purposes.

(*a*) (*b*)

Figure 1—Example of a PV canopy installed over a parking lot. (*a*) is a top view to show the surfaces of the modules; (*b*) is a ground view to depict the supporting structure. Source: Swenson Solar [18]

1.1.1.5 Components of Solar Irradiance

Solar irradiance is divided into three components: direct normal, diffuse horizontal, and albedo. Direct normal irradiance (DNI) strikes the earth in near-parallel beams that have the capacity to be focused [12]. Beam direction is determined by the position of the sun in the sky [19]. Direct horizontal irradiance is the component of the DNI that is perpendicular to the horizontal plane. Diffuse horizontal irradiance (DHI) is scattered sunlight, caused by clouds and other atmospheric conditions [19]. Though it comes from all directions and cannot be focused [12], DHI is anisotropic because of the brightening around the solar disk in the sky [10]. Albedo is the fraction of irradiance that is reflected from the ground [19]. Global horizontal irradiance (GHI) is the total

irradiance incident on a horizontal surface, comprised of the direct horizontal irradiance, DHI, and albedo [19].

1.1.1.6 Optimization of Module Orientation

In the absence of clouds, DNI has approximately five times the energy content of DHI [20]. To ensure maximum electricity production, PV systems should be designed to collect as much DNI as possible. This is achieved by optimizing module orientation, which is described in spherical coordinates by the tilt (β) and azimuth (A_M) angles. The tilt is the angle between the module and the horizontal plane. The azimuth is the angle between true north and the projection of the module normal ($\mathbf{n}_M$) onto the horizontal plane, measured clockwise from north. These angles are illustrated in Figure 2.

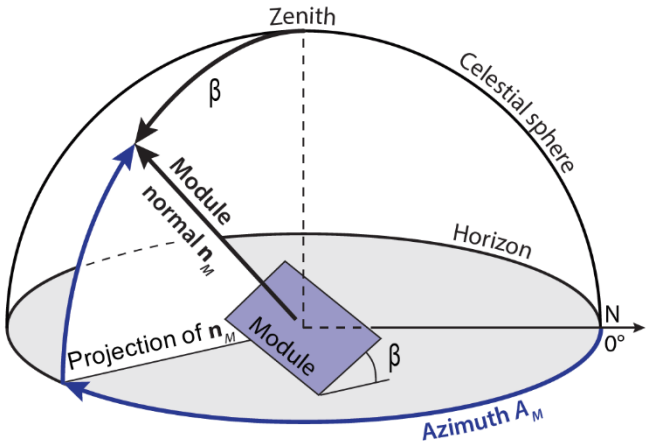


Figure 2—Illustration of module tilt (β) and azimuth (A_M) orientation angles. Adapted from: K. Jäger et al., 2014 [21]

Since DNI travels through space in near-parallel beams, the amount of DNI incident on a module is proportional to the cosine of the angle of incidence between the

module normal and beam direction [21]. Incident DNI is greatest when it is parallel with the module normal, decreasing in magnitude as the angle of incidence increases. Figure 3 illustrates this, with the two-dimensional scenario of the DNI vector on a plane perpendicular to the module. Letting the width of the module be unity, the area of incident DNI is reduced from module length d to $d\cos\gamma$, were γ represents the angle of incidence.

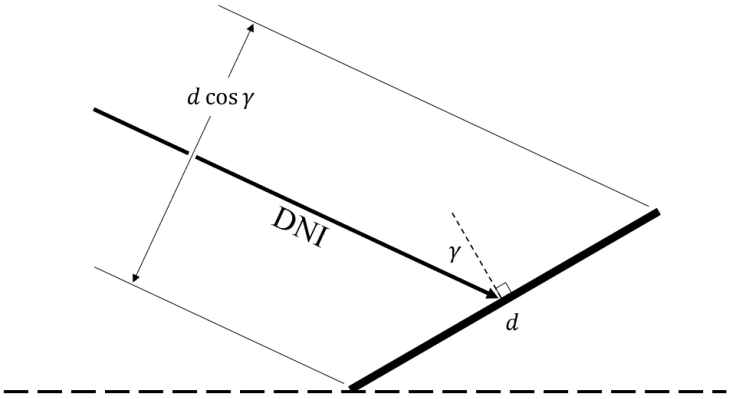


Figure 3—Dependence of incident DNI on angle of incidence with module normal.

Since PV modules generate the most electricity when they are normal to the DNI, tracking modules in dry climates collect approximately 50% more energy than static modules during the summer months, and 20% more in the winter [12]. Tracking modules in wetter climates have lower gains due to the relative increase in DHI, collecting 35% more energy during the summer and 9% more in the winter [22]. Without consideration of weather, static modules are best oriented normal to the position of the sun at the average annual solar noon [12]. This is when the sun is highest in the sky and radiates through the least amount of atmosphere, minimizing absorption and scattering [10].

During solar noon, the sun is perfectly south in the Northern Hemisphere and perfectly north in the Southern Hemisphere, with seasonal exceptions between the Tropics of Cancer and Capricorn [12]. The optimal static module azimuth angle is therefore either true south (180°) or true north (0°), depending on location [12].

1.1.1.7 Shading

Another way to maximize the incident DNI is to reduce shading whenever possible. Shading is caused by an obstruction positioned between the PV module and the sun, which completely blocks the DNI that would have been incident on the module [10]. Without the DNI, the module power output is significantly reduced because it can only collect energy from the lower intensity DHI [10]. The effect of shading is exacerbated according to the module and array wiring, as a shaded cell in series with unshaded cells will restrict the current in the string to that of the shaded cell [23]. Common obstructions are tree branches, buildings, and dust on the module [23].

1.1.1.8 Cell Technologies

Most PV cells are made from crystalline silicon because of the low cost [24]. Monocrystalline silicon is the most efficient but also the most energy intensive to manufacture, leading to the development of polycrystalline silicon, amorphous silicon, and non-silicon cells [12]. Of the silicon varieties, polycrystalline requires less manufacturing energy and amorphous requires the least, with a corresponding decrease in efficiency [12]. Amorphous silicon has applications in thin-film cells, commonly referred to as second generation PV. Thin-film cells have the capability to be flexible but

have remained economically uncompetitive due to their lower efficiencies [21]. Other common types of thin-film cells are gallium arsenide, copper indium gallium diselenide, and cadmium telluride [21]. Third generation PV classifies the emerging technologies, including supertandem, organic PV (OPV), perovskite, and quantum dot cells [12]. Many of these are currently under development in labs with challenges on securing long-term stability [12]. Each technology poses unique advantages over traditional crystalline silicon cells. Notably, the FhG-ISE Soitec supertandem cell achieved a 46% efficiency, which is over 150% of the record silicon cell efficiency [25]. The disadvantage is that supertandem cells are currently very costly to produce, requiring concentrating optics and tracking devices to minimize cell size [26].

OPV cells are perhaps the most promising of the third generation PV technologies. They require 94% less production energy than monocrystalline silicon cells and are flexible, lightweight, and recyclable [27]. This allows them to be manufactured at a lower cost, corresponding to module costs that are projected to be 24% less than crystalline silicon by 2020 [28]. OPV cells have lower efficiencies than silicon cells, with a maximum efficiency of 15.6% achieved by the South China University of Technology [25]. The efficiency of an OPV cell degrades two to four times faster than a silicon cell, reducing the operational lifetime to five to ten years [28]. This high degradation rate makes this technology currently unsuitable for transit applications, as frequent maintenance intervals would reduce rider throughput. However, this is an emerging technology and should not be discounted for future consideration.

1.1.2 Current State of Solar-Powered Transit

Solar-powered transit is a new effort to reduce greenhouse gas emissions in the transportation sector. Automobiles powered by PV cells do not exist outside of one-off designs for competitions like the Formula Sun Grand Prix and World Solar Challenge [29]. While these designs prove that solar-powered automobiles are possible, they are not practical because of their cramped cockpits and high costs of up to $300,000 [29]. Ford and Toyota have begun integrating small PV arrays in the roofs of the 2014 C-Max Solar Energi and 2010 Prius, to charge the hybrid battery and power auxiliary electronics [29]. However, the limited surface area of the cab restricts the capacity for PV cells. The popular alternative to onboard PV cells is to charge electric vehicles at home with the energy collected from larger rooftop PV arrays [29].

There are currently two fully solar-powered public transit vehicles in existence, shown in Figure 4: the 2017 Byron Bay train (*a*) and the 2007 Tindo bus (*b*), both in Australia. The Byron Bay train is a shuttle across a three-kilometer rail corridor, featuring 6.5 kW of PV modules on top of the two carriages [30]. These modules offset the energy consumed during operation, and the batteries are fully charged by the 30 kW PV array atop the train storage shed [30]. The Tindo electric bus features a 68 W onboard PV module, but is primarily charged by the PV array on the roof of the central station [31]. It has a 200 km range between charges and operates every day [31]. The limiting factor for solar-powered transit is the surface area of vehicles, which necessitates battery charging from larger PV arrays. No transit applications currently exist with PV arrays built over the top of the route.

(*a*) (*b*)

Figure 4—Existing solar-powered public transit vehicles. (*a*) is the Byron Bay train in Australia; (*b*) is the Tindo bus, also in Australia. Source (*a*): G. Parkinson, 2017 [32]; Source (*b*): MPower [33]

1.1.3 Automated Transit Networks (ATNs)

An ATN is a fully connected network of guideways and stations, within which small autonomous vehicles carry passengers directly to their destinations without transfers or stops [8]. The concept has been in existence since the 1950s and five ATNs are currently in operation around the world [8]. ATNs are more efficient than car-based transportation because they travel directly from origin to destination with no possibility of congestion. They are safer than cars because they operate on isolated guideways, eliminating the possibility of collision with pedestrians, cyclists, and vehicles [8].

All existing and proposed ATNs are powered electrically, with either onboard battery storage or dynamic wayside pickup [8]. The electric nature allows these systems to be powered by solar or wind energy [34], which can be incorporated into guideway and station infrastructure according to resource availability [35]. Economic feasibility of solar-powered ATNs has improved from the exponentially decreasing cost of PV modules [36] and associated low energy costs [14]. PV integration would allow these

transit networks to be energetically self-sufficient with zero emissions, yielding potential to drastically reduce greenhouse gas production.

1.1.4 Prior Spartan Superway PV Design Work

The Spartan Superway program was initiated in 2012 after successful efforts in a solar-powered ATN design challenge from the International Institute of Sustainable Transportation [8]. The Superway design comprises a canopy of PV modules mounted above the guideway, with vehicles suspended underneath [9]. The canopy is static, as opposed to tracking, to reduce maintenance and associated system downtime [37]. The propulsive assembly that supports the vehicle, termed the bogie, is enclosed in the guideway for protection from the elements [9]. Vehicles are pod-shaped and have the capacity for four to six passengers [9]. Throughout the continued research and development efforts, two pieces of work have been prominent in the design of the PV system.

The first work was a case study in 2016 on a proposed ATN route between the north and south San José State University campuses [35]. This route was defined as having a length of 14.4 km with 150 vehicles in constant operation between 14 stations [35]. Energy modeling was performed and yielded a daily system requirement of 28.5 MWh, which was used to size the PV canopy [35]. This was conducted with the System Advisory Model (SAM) solar simulation software from the National Renewable Energy Laboratory (NREL) [35]. It was predicted that 19,600 PV modules would be required, spanning an area of 38,300 m^2 [35]. A three-meter wide PV canopy along the length of the guideway would provide 43,200 m^2 of area, delivering ample energy to power the

ATN [35]. The conclusion was that ATNs have sufficient area on top of the guideway for a PV array to sustain 24/7 operation [35]. One consideration is that the PV simulation was conducted with all modules set to the same orientation of south-facing with a 30° tilt [35]. This is unrealistic because the PV canopy follows the direction of the route, changing the orientation of the modules at each turn. To more accurately represent the different orientations, future work may entail simulating the PV array on each leg of the route.

The second work was the design and fabrication of a three-meter sample section of PV canopy, completed in 2018. The project identified five requirements for the canopy design [38]:

1. Integration with the top surface of the guideway.
2. Curved PV array for aesthetics.
3. Closely connected PV modules for rainwater management and bogie shelter.
4. Withstand loading scenarios caused by high winds.
5. Easy to manufacture.

The design solution was a modified Double Howe Truss with six PV modules arranged in a semi-circular segment [38]. The tilt progression of the modules was 10°, 27.5°, and 35° from horizontal on each side [38]. Unistrut® channel was used for the construction and ANSYS structural simulations returned a von Mises-stress safety factor of 15 [38]. The conclusion was that the prototype successfully fulfilled all design requirements [38]. One consideration, however, is that no mention was made toward optimizing the PV module orientations for collection of DNI. To improve system

efficiency and economic payback, future work may involve determining the module tilt progression that maximizes incident DNI.

1.1.5 Modeling PV Systems

For PV systems of every scale, it is useful to model the performance and cost to optimize the design and economic feasibility. This begins with simulating the changing direction of incident irradiance on the PV modules throughout the annual solar cycle. An hourly interval is sufficient [39], requiring 8,760 sun-position calculations per cycle. It is therefore practical to develop a computer program to efficiently perform these computations and manage the results. Weather effects can be incorporated by utilizing a data base of weather station records [12]. One such archive is the National Solar Radiation Data Base (NSRDB) from NREL, which uses a combination of weather station data and meteorological models to predict the weather effects on irradiance [40]. Once the incident irradiance is modeled, losses from shading, module efficiency, and electrical component efficiencies are incorporated [41]. The project performance data is then often used to calculate financial metrics like payback period and internal rate of return [22]. Many commercial and free programs exist for designing residential, commercial, and utility-scale PV systems. Three popular options are SAM by NREL [22], HelioScope by Folsom Labs [42], and SunDAT by FTC Solar [43]. These programs are limited to the design of planar PV arrays, in which modules are mounted on single or parallel planes across a rooftop or parcel of land. No commercial PV design software was discovered that can model arrays in non-planer configurations, as would be required for the design of PV canopies for transit applications.

1.2 Problem Statement

The application of PV canopies for solar-powered transit is unique in that the canopy extends over the top of the route. This protects the track and vehicles from exposure to sunlight and weather, with opportunities for rainwater management through gutters or storage tanks. The canopy bearing is defined by the direction of the route, creating situations for non-ideal west- or east-facing modules. Despite this azimuthal constraint, the module tilt on each leg of the route should be optimized for DNI collection. Existing PV design software cannot perform these optimizations, establishing the need to develop a program. Additionally, the aesthetics of the canopy are of great importance because transit infrastructure becomes part of the associated city image. Applying curvature to the PV canopy is an aesthetic solution that causes deviations from the optimal module tilt. Thus, it is necessary to investigate the reduction of irradiance collection caused by applying array curvature. Another topic for study is whether irradiance collection can be enhanced by using array profiles that are dependent on the bearing of the route.

1.3 Objectives

The objectives of the project are to:

- Document the development of a MATLAB program that inputs route information and component properties, and outputs the optimal PV canopy configuration and annual-hourly electricity generation.
- Investigate the efficiencies of curved versus planar PV arrays.
- Investigate the use of different PV array profiles for legs of the route with different bearings.
- Provide a basis for future ATN developments on electrical system design, energy storage, vehicle throughput, and cost analysis.

2.0 METHODOLOGY

The methodology describes the actions taken to achieve the project objectives through the avenues of analytical work, experimental work, and computer simulation.

2.1 Analytical Work

The analytical work comprises the development of the computational aid with emphasis on the mathematical representation of the route, irradiance, and PV modules. All programming is done with MATLAB, which was selected for its versatility, convenient built-in functions, capability to communicate with application program interfaces (APIs), and availability of learning resources. The developmental steps for the computational aid are:

1. Input the route information.
2. Model the irradiance at the route location.
3. Model the energy demand of the transit system.
4. Model the shading from surrounding structures.
5. Conduct simulation studies to quantify performance of different array profiles.
6. Model the module power output.
7. Simulate and optimize the PV canopy.
8. Integrate the algorithms into a single script.

2.1.1 Route Input

The purpose of the route input algorithm is to determine the PV canopy length, incline angle, and azimuth angle for each leg and station of a user-defined route.

Additionally, the average elevation and central coordinates of the route are determined for the module and irradiance models, respectively. The route length and average distance between stations are also computed for use in the energy demand model. Appendix A features the loadRoute() function for reference, which is executed in the following seven steps:

1. Route Definition
2. Calculation of Leg Lengths
3. Calculation of Leg Bearings
4. Calculation of Incline Angles
5. Corner Representation
6. Station Representation
7. PV Canopy Representation

2.1.1.1 Route Definition

Route definition is most conveniently performed graphically, by marking the route and stations on a map. Google Maps [44] was selected as the platform because it has a user-friendly interface, free accessibility, and offers quality geographic information with satellite imagery. The legs of the route can be drawn with the line tool, while station locations can be set with the marker tool. The route definition can then be downloaded as a Keyhole Markup Language (KML) file, which is a file format used to store geographic information for Google programs [45]. Figure 5 shows an example of the Google Maps interface, with an ATN route fully defined by lines and markers. Note that a closed loop

of line segments forms a polygon, which is depicted with a gray shaded region. Though polygons have a different appearance, they are processed as lines.

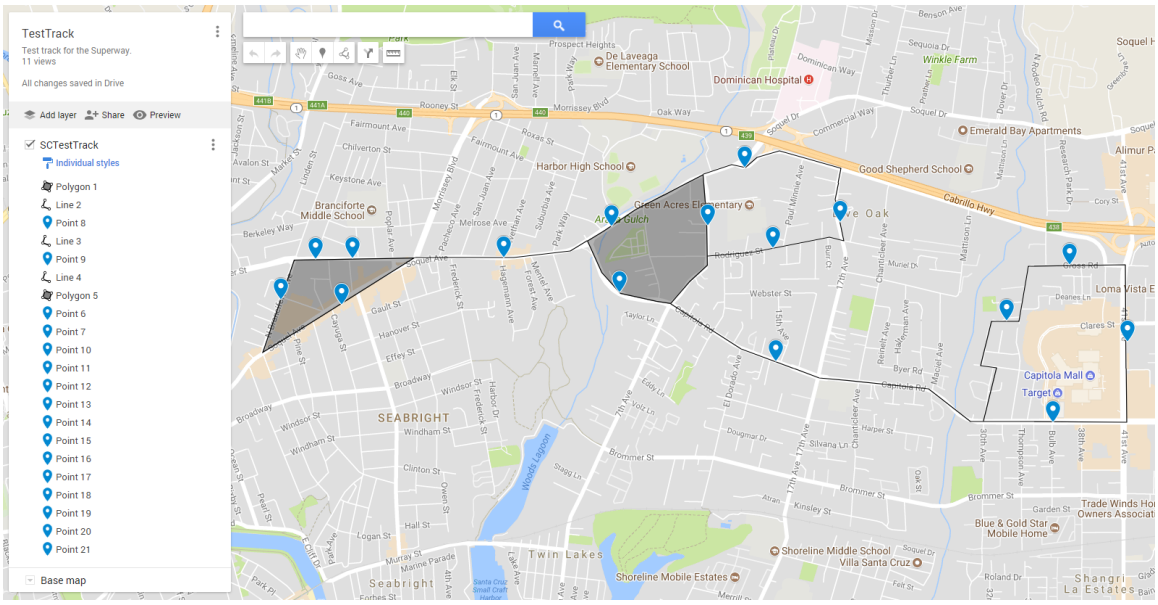


Figure 5—ATN route graphically defined with Google Maps.

Among other information, KML files store the longitude and latitude of each vertex in the user-defined route. These coordinates can be extracted with MATLAB using the kml_shapefile() function, which was created by Michael Toomey in 2010 and is available via the MATLAB File Exchange data base [46]. The imported route coordinates are then sorted into legs and stations, based on the number of vertices per feature.

2.1.1.2 Calculation of Leg Lengths

Each leg of the route is defined by its start and end point coordinates. The length of each leg (L) is calculated as the great-circle distance between two points on the surface of an Earth-sized sphere, as illustrated in Figure 6.

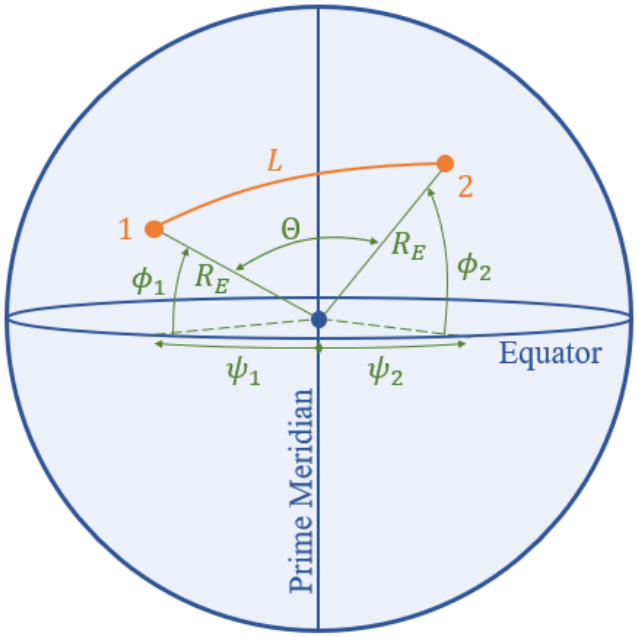


Figure 6—Illustration of the leg length (L) calculation parameters. These include the start and end point longitudes (ψ) and latitudes (ϕ), measured from the prime meridian and equator, respectively. The radius of Earth (R_E) and angular distance between the start and end points (Θ) are also shown.

The leg length computation is executed with the subsequent haversine spherical trigonometric relations from [47], [48]. The inputs are the start and end point longitudes (ψ) and latitudes (ϕ), which are used to compute the haversine of the angular distance ($\mathrm{hav}(\Theta)$).

$$\mathrm{hav}(\Theta) = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\Delta\psi}{2}\right) \quad (1)$$

The angular distance is then calculated with the following haversine identity. The $\mathrm{atan2}(Y, X)$ function is the four-quadrant inverse tangent, which extends the range of the traditional inverse tangent from $[-\pi/2\,, \pi/2]$ to $[-\pi, \pi]$ [49].

$$\Theta = 2\,\mathrm{atan2}\left[\sqrt{\mathrm{hav}(\Theta)}, \sqrt{1 - \mathrm{hav}(\Theta)}\right] \quad (2)$$

Finally, the leg length is the product of the radius of Earth (R_E) and the angular distance between the start and end points. The average radius of Earth is 6371 km [48].

$$L = R_E\Theta \quad (3)$$

This leg length algorithm was implemented in the LonLatDistAz() function, presented in Appendix B. This function is called in the route input algorithm for each leg.

2.1.1.3 Calculation of Leg Bearings

The leg bearing (θ) is the angular direction of the leg, measured clockwise from true north in degrees. Unless the leg is perfectly latitudinal, the bearing will change between the start and end points [48]. Since the bearing is calculated from one point to another, it is termed the initial bearing (θ_{12}) when measured from the start point, and final bearing (θ_{21}) when measured from the end point, as illustrated in Figure 7.

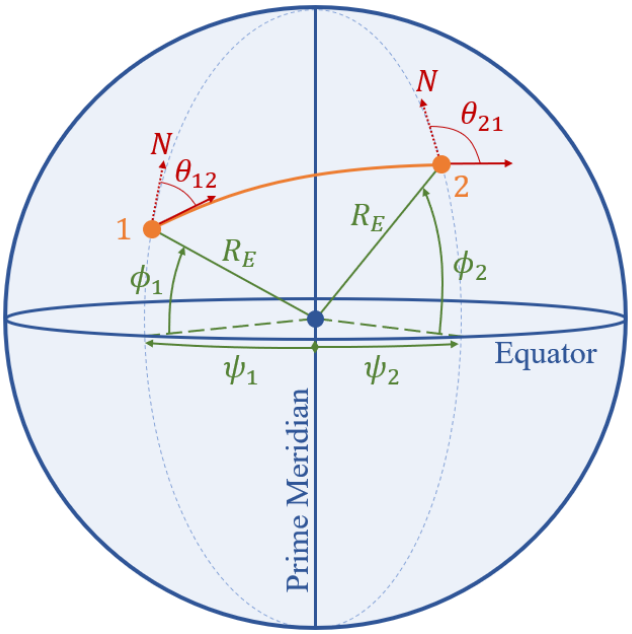


Figure 7—Illustration of the leg bearing calculation parameters. The initial bearing (θ_{12}) and final bearing (θ_{21}) are shown as measured clockwise from true north. Also included are the start and end point longitudes (ψ) and latitudes (ϕ), measured from the prime meridian and equator, respectively. The radius of Earth (R_E) is shown for reference.

The initial bearing is computed according to the following equation from [48]. As with Equation (1), the inputs are the start and end point longitudes and latitudes.

$$\theta_{12} = \text{atan2}[\sin(\Delta\psi)\cos(\phi_2), \cos(\phi_1)\sin(\phi_2) - \sin(\phi_1)\cos(\phi_2)\cos(\Delta\psi)] \quad (4)$$

The final bearing is also calculated with Equation (4), but since it is measured from the end to start point, all subscripts are reversed. As mentioned earlier, the range of the $\text{atan2}(Y, X)$ function is $[-180°, 180°]$, which must be shifted to $[0°, 360°]$ according to convention. This is done with the modulus function, according to [48].

$$\text{Initial Bearing:} \quad \theta_{12} = \text{mod}(\theta_{12} + 360°, 360°) \tag{5}$$

$$\text{Final Bearing:} \quad \theta_{21} = \text{mod}(\theta_{21} + 180°, 180°) \tag{6}$$

Finally, for a single value, the initial and final bearings are averaged.

$$\theta = \frac{\theta_{12} + \theta_{21}}{2} \tag{7}$$

This leg bearing algorithm follows the leg length computation in the LonLatDistAz() function, featured in Appendix B.

A study was conducted to validate the accuracy of representing the leg bearing with a single value. Since bearing variation depends on latitude, according to Equation (4), a latitude domain of ±60° was selected. This is because 99.7% of the world population lives within this range [50]. Since ATNs are designed to function in urban environments, the maximum possible longitudinal leg length was taken as the longitudinal length of Los Angeles—the widest urban area in the world [51], [52]. Los Angeles has a maximum longitudinal length of 130 km between Santa Monica and San Bernardino [52]. According to Equation (1), Equation (2), and Equation (3), 130 km is a longitudinal span of 2.34° at the maximum latitude of 60°. These parameters were used in Equation (4), Equation (5), and Equation (6) to yield an initial bearing of 89.0° and a final bearing of 91.0°. The average bearing from Equation (7) is 90.0°, which is only 1.0° different from the initial and final bearings for this worst-case scenario. It is thus acceptable to represent the leg bearing with a single value.

2.1.1.4 Calculation of Incline Angles

The KML file from Google Maps does not include elevation information, causing the route to be represented on level ground. However, some urban areas, like San Francisco and Seattle, feature steep hills that would significantly alter the level-route module orientations. To quantify the error of ignoring route incline, a study was conducted to compare the annual incident irradiance on a module before and after a 25% incline was applied. This grade was selected because 25 streets in San Francisco have inclines of over 25% [53]. A PV simulation software called Solrad [20] was used to compute the annual incident irradiances for module azimuth angles of 90°, 120°, 150°, 180°, 210°, 240°, and 270° at latitudes of 0°, 15°, 30°, 45°, and 60°. The pre-incline module tilt was the optimal tilt at each azimuth angle and latitude. The incline was applied counterclockwise about the projection of the module normal onto the horizontal plane. Figure 8 shows the results, indicating the incline error increases with latitude and with deviation of azimuth angle from true south. This was expected because higher latitudes have larger optimal module tilts to account for the lower position of the sun in the sky. This increases the sensitivity of module orientation on annual incident irradiance. East- and west-biased azimuth angles also deviate from the optimal south-facing position, increasing module orientation sensitivity. The maximum annual irradiance error in the study was an underrepresentation of 17.2% for the 60° latitude and 270° azimuth angle scenario. This error is large enough to necessitate the incorporation of route incline effects on module orientation, requiring the acquisition of elevation information.

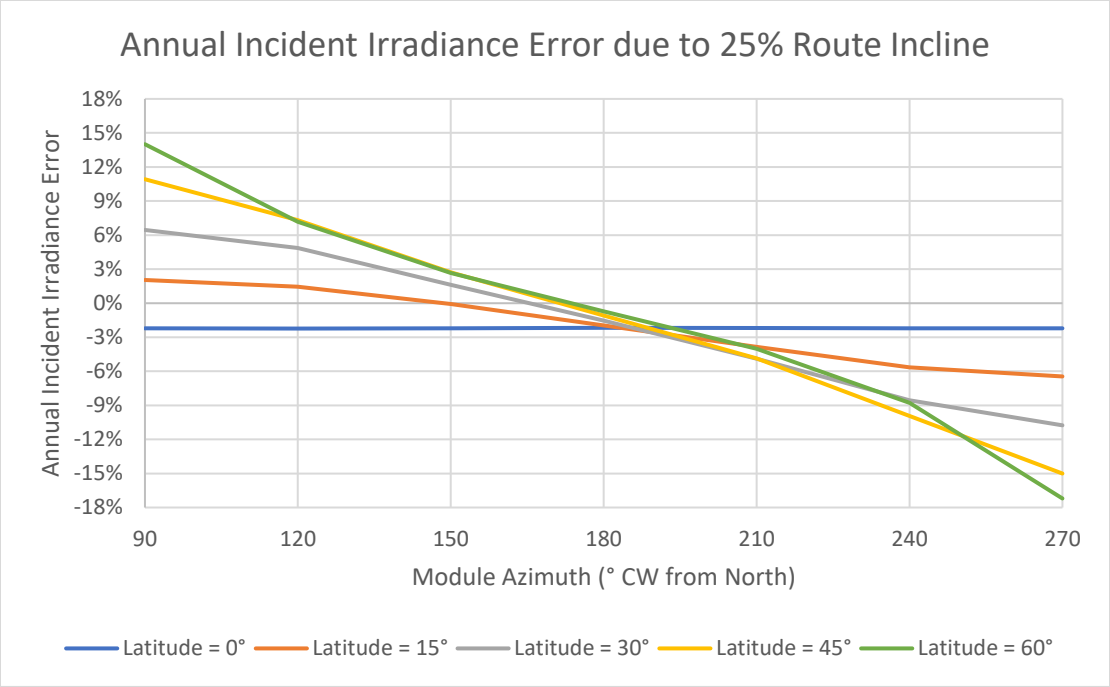


Figure 8—Annual incident irradiance error caused by applying a 25% route incline.

Since Google Maps does not export elevation information, the getUSGSelevation() function was created to utilize the United States Geological Survey (USGS) Elevation Point Query Service [54]. This function sends a request with point coordinates and receives the corresponding elevation in meters. The service covers most of the United States and will respond if points are queried outside the domain, returning a flag value to indicate that the requested elevation is unknown [54]. The getUSGSelevation() function is listed in Appendix C for reference.

The elevation for each leg vertex is queried with the getUSGSelevation() function, which takes approximately two seconds per point. To reduce execution time, an algorithm was constructed to check if the start point of the next leg is the end point of the current leg, as is the case for a chain of legs. If this condition is true, the current end point elevation is assigned to the start point of the next leg, to avoid querying the value.

This reduced the execution time by approximately 50%, though this reduction is dependent on the number of chains of legs. To further reduce execution time, the elevation values are saved in a comma-separated values (CSV) file, which is loaded at the beginning of future executions. If the leg vertices match those stored in this file, the corresponding elevation values will be loaded instead of queried.

The incline angle (Φ) is defined as the counterclockwise angular displacement about the projection of the module normal onto the horizontal plane. It is computed for each leg through a simple tangent relationship between the length of the leg on level ground (L_{level}) and the difference between vertex elevations (Δh). This is illustrated in Figure 9. If the elevation of either vertex is undefined, the incline angle is set to zero. The incline angle of each station canopy is set to the value of the associated leg.

$$\Phi = \tan^{-1}\left(\frac{\Delta h}{L_{level}}\right) \tag{8}$$

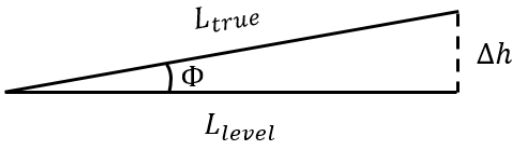


Figure 9—Illustration of incline angle (Φ) and true leg length (L_{true}).

To ensure the sign of the incline angle is correct, the difference between vertex elevations is dependent on the hemisphere of the route. These relations are subsequently listed, with special conditions for legs that are oriented exactly latitudinally.

Northern Hemisphere:

General Case: $\Delta h = h_{East\ Vertex} - h_{West\ Vertex}$

Latitudinal Leg: $\Delta h = h_{North\ Vertex} - h_{South\ Vertex}$

Southern Hemisphere:

General Case: $\Delta h = h_{West\ Vertex} - h_{East\ Vertex}$

Latitudinal Leg: $\Delta h = h_{South\ Vertex} - h_{North\ Vertex}$

Each leg length must also be corrected because the true length (L_{true}) is longer than the level length, as indicated in Figure 9. This is conducted with the Pythagorean Theorem, though the correction is omitted if the elevation of either vertex is undefined.

$$L_{true} = \sqrt{L_{level}^2 + \Delta h^2} \tag{9}$$

Finally, the mean of all defined vertex elevations is determined to return the average elevation of the route. This value is required for the module model.

2.1.1.5 Corner Representation

A constructed ATN route will have curved corner transitions between legs. In order to accurately represent these curved corners, tangent arcs must be applied to each leg intersection. This is illustrated in Figure 10 with two 100 m legs, 110° apart, connected by a tangent arc of 30 m radius. The red shaded regions indicate the lengths of each leg that need to be removed to represent the true leg lengths.

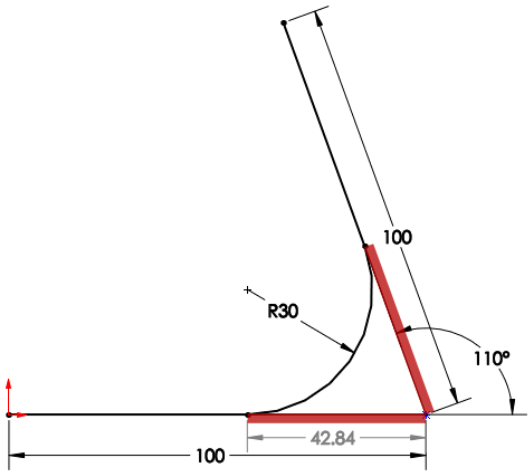


Figure 10—Leg trimming regions for accurate corner representation.

An algorithm was created to trim the ends of each leg for accurate corner representation. First, the change in bearing at the start and end of each leg is calculated. This is done by subtracting each leg bearing from that of the next leg in the route, correcting the result by taking the absolute value and subtracting it from 360° if it is larger than 180°. In the case of an open-loop chain of legs, the start and end changes in bearing are defaulted to 90°, assuming a perpendicular intersection with another leg. The trimming length (L_{trim}) is then calculated as follows, in function of the corner radius (R_c) and change in bearing ($\Delta\theta$).

$$L_{trim} = \frac{R_c}{\tan\left(\frac{180 - \Delta\theta}{2}\right)} \tag{10}$$

The corner radius is left as a user input because it will depend on the design of the route. Once the trimming lengths at the beginning and end of each leg are determined, they are subtracted from the original leg length. If the resulting trimmed leg length is

negative, it is set to zero and a warning is issued to inform the user. Figure 11 illustrates the application of the corner representation algorithm, in which sharp corners are appropriately trimmed more than gradual ones. The original route is displayed in solid black and the trimmed legs are overlaid in dashed orange.

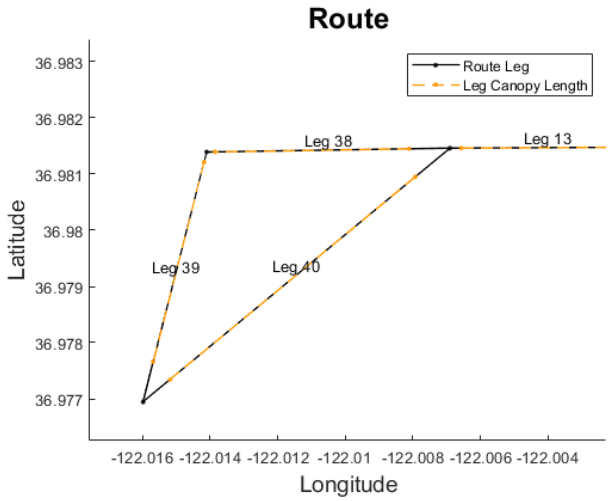
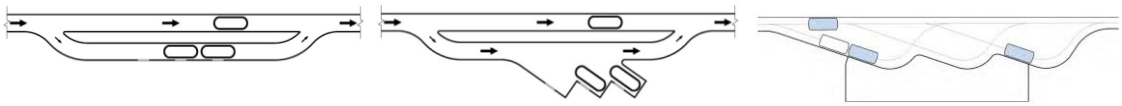


Figure 11—Application of the corner representation algorithm.

2.1.1.6 Station Representation

A constructed ATN route will have stations that are wider than the legs to allow space for vehicles to go offline for loading and unloading passengers [8]. The lengths of stations can vary and should depend on the estimated volume of passengers for each stop [8]. Figure 12 shows three examples of station configurations: series (*a*), parallel (*b*), and combination (*c*). Each configuration requires a larger PV canopy for shelter, which also increases the generated electricity to offset the large energy demand from the accelerating loaded vehicles.



(*a*) (*b*) (*c*)

Figure 12—Examples of ATN station configurations. (*a*) is a series configuration; (*b*) is a parallel configuration; (*c*) is a combination of series and parallel configurations. Source: B. Furman et al., 2014 [8]

A four-step algorithm was created to accurately represent the stations based on the single-point coordinates imported from the KML file. The first step is to snap each station to its associated leg for proper location representation. This begins by associating each station with the nearest leg through a series of vector projections. The vector definitions are illustrated in Figure 13 with the leg vector (**L**) from the start point (L_1) to end point (L_2), orthogonal leg vector (**v**) from the station point (S), and station-leg vector (**SL₁**) from the station point to the leg start point. The perpendicular distance ($d_\perp$) from the station point to the leg vector is also presented, as well as the parallel distance ($d_\parallel$) along the leg vector from the start point to the intersection of the orthogonal leg vector.

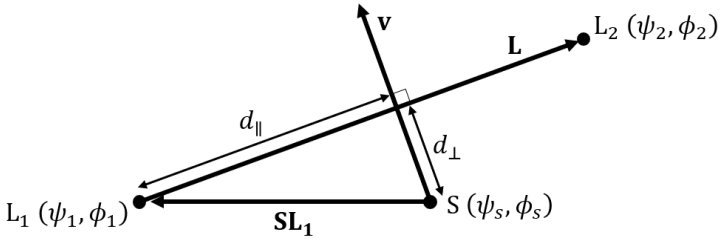


Figure 13—Definitions of the leg (**L**), orthogonal leg (**v**), and station-leg (**SL₁**) vectors. The perpendicular ($d_\perp$) and parallel ($d_\parallel$) distances are also illustrated.

The leg, orthogonal leg, and station-leg vectors are represented mathematically as follows.

$$\mathbf{L} = \begin{bmatrix} \psi_2 - \psi_1 \\ \phi_2 - \phi_1 \end{bmatrix}, \qquad \mathbf{v} = \begin{bmatrix} \phi_1 - \phi_2 \\ \psi_2 - \psi_1 \end{bmatrix}, \qquad \mathbf{SL_1} = \begin{bmatrix} \psi_1 - \psi_S \\ \phi_1 - \phi_S \end{bmatrix}$$

The perpendicular distance is calculated as the absolute scalar projection of the station-leg vector onto the orthogonal leg vector.

$$d_{\perp} = \frac{|\mathbf{SL_1} \cdot \mathbf{v}|}{\|\mathbf{v}\|} \tag{11}$$

The parallel distance is the scalar projection of the reverse station-leg vector onto the leg vector.

$$d_{\parallel} = \frac{-\mathbf{SL_1} \cdot \mathbf{L}}{\|\mathbf{L}\|} \tag{12}$$

These distances are used to calculate the distance from the station to the leg. This is the perpendicular distance if the parallel distance is between zero and the leg length. If the parallel distance is negative, the station distance is the distance between the station point and the leg start point. Otherwise, it is the distance between the station point and the leg end point. The station distance is computed for each leg, allowing the station to be associated with the nearest leg. Each station is then snapped onto its associated leg by replacing the station coordinates with the nearest coordinates on the leg.

At this stage, the station is represented as a point and therefore has no dimensions. A length is thus assigned to each station, which is left as a user input because the value will depend on the route design. The station point is set as the midpoint of the station

length. This process is illustrated in Figure 14, as Station 17 is snapped to associated Leg 43. The thick orange line above the station point in the right image is the station length.

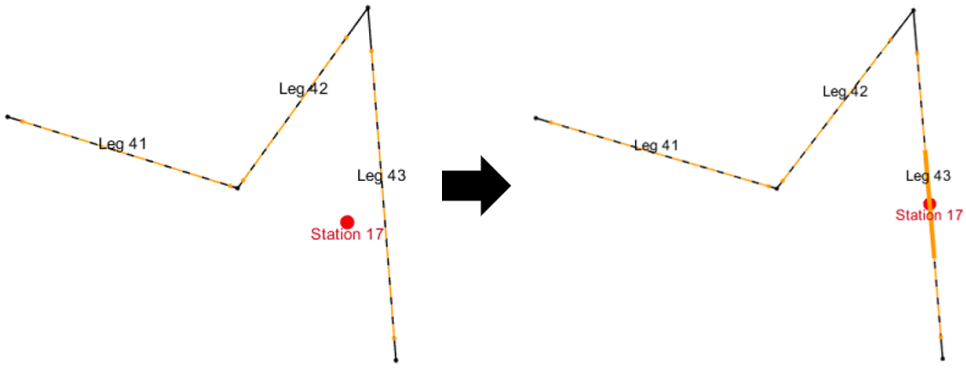


Figure 14—Implementation of station snapping and lengthening algorithm.

The next step of the algorithm is to translate each station if either end extends beyond the associated trimmed leg. This is executed by calculating the distance from the station midpoint to each end of the trimmed leg. If the distance is less than half of the station length, the station endpoint is set to the overlapping trimmed leg endpoint and the rest of the station is translated accordingly. This is exemplified in Figure 15, in which Station 45 is translated so that it does not extend beyond the trimmed leg.

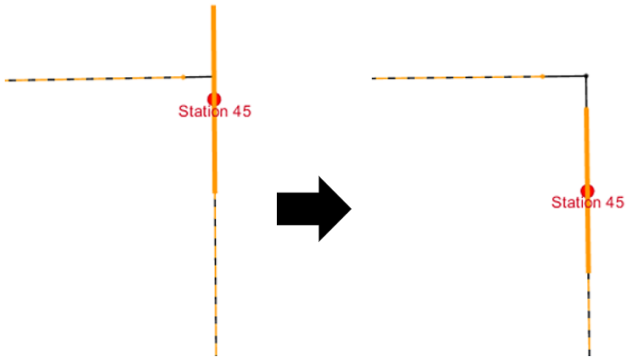


Figure 15—Implementation of station translation algorithm.

The third step is to manage overlapping stations, which are most accurately represented as a single station of combined length. Overlapping stations are detected by computing the distance between each station on each leg. If this distance is less than the station length, the stations are flagged as overlapping and grouped in an array. Then the two stations in the array that are furthest from each other are identified and the distance between their midpoints is calculated. The station length is added to this value for the total length of the overlapping group of stations. This length is assigned to the station in the group with the lowest name number, while the lengths of the other stations in the group are set to zero. This ensures the station length is not overrepresented. Figure 16 depicts the implementation of this algorithm, in which Stations 29, 32, 35, and 40 are identified as overlapping and combined into a single station. The end points of the individual stations are shown in the left image, which are replaced by the single thick line in the right image.

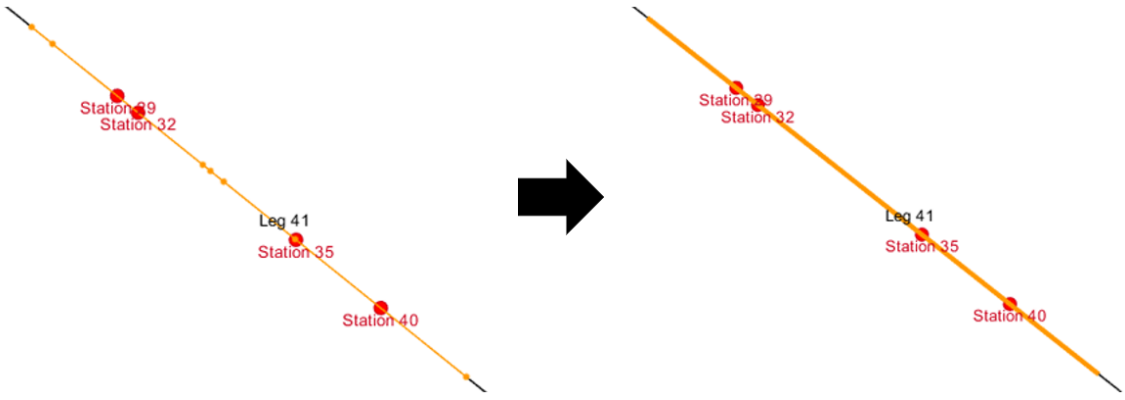


Figure 16—Implementation of the station overlap algorithm.

The final step of the station representation algorithm is to ensure that each station fits on its associated trimmed leg. If the length of any station is greater than that of its associated trimmed leg, the station length is reset to the trimmed leg length. In this event, a warning is issued to notify the user of the trimmed station, should they want to adjust the route to increase the length of the associated leg. However, subsequent execution will continue. All station lengths are then subtracted from the lengths of their associated legs. Figure 17 shows the use of this algorithm, as Station 15 is set to the trimmed length of associated Leg 24.

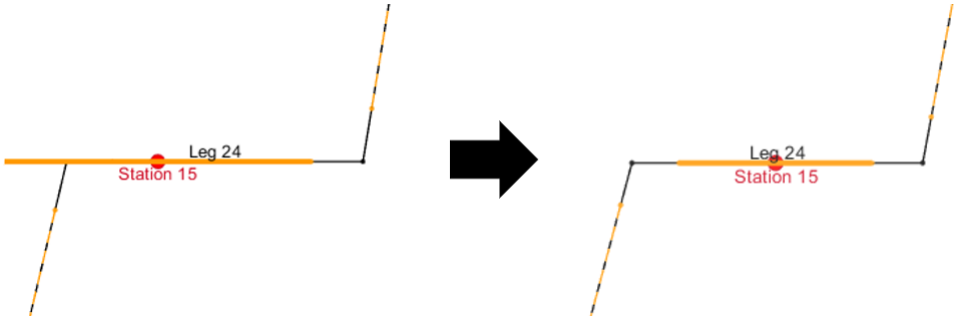


Figure 17—Implementation of the station trimming algorithm.

2.1.1.7 PV Canopy Representation

At this point, the route legs and stations are accurately represented and have known lengths and bearings. The central coordinates are determined next, which mark the center-point of the route for canopy azimuth calculations and irradiance modeling. The central longitude and latitude are the averages of the maximum and minimum trimmed leg longitudes and latitudes, respectively.

The canopy azimuth angle is the azimuth of the modules it supports. Since the optimal azimuth is 180° for modules in the Northern Hemisphere, and 0° in the Southern Hemisphere, the route hemisphere is first determined from the central coordinates. Then each leg bearing is modified by subtracting 180° if it is greater than or equal to 180°. This removes the start-and-end orientation of the legs, representing them as lines described by the modified bearing angle measured clockwise from true north. If the route is in the Northern Hemisphere, 90° is added to the modified bearing. Otherwise, 90° is subtracted from the modified bearing, according to Figure 18. The canopy azimuth angles for the stations are set to the values of their associated legs.

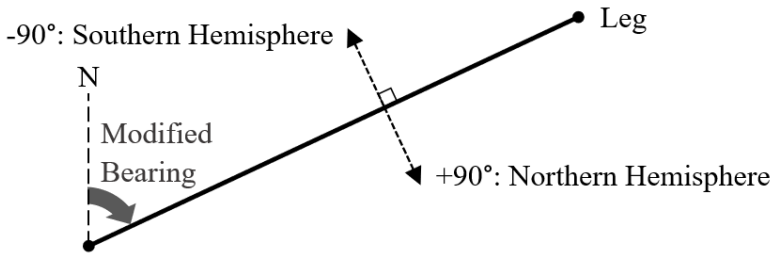


Figure 18—Hemispherical convention for calculating PV canopy azimuth angles.

The length of the route and average distance between stations are computed for use in the energy demand model. The length of the route is returned as the sum of the untrimmed leg lengths. The formula for the average distance between stations was derived for the ideal ATN route configuration detailed in [55]. This configuration is comprised of legs arranged in a grid with stations between intersections, as illustrated in Figure 19.

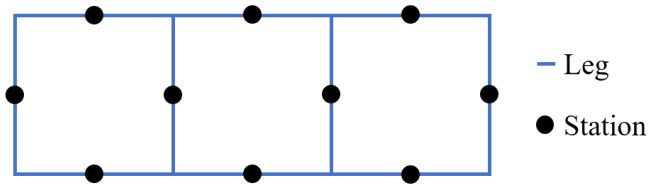


Figure 19—Ideal ATN route configuration.

The number of route segments between stations $(N_{segments})$ is mathematically represented as a function of the number of stations $(N_{stations})$. As an example, the configuration in Figure 19 has 10 stations that divide the route into 8 segments.

$$N_{segments} = \frac{2}{3}N_{stations} + \frac{4}{3} \tag{13}$$

The floor() function is applied to Equation (13) to accommodate larger routes that lack symmetry, like that of Figure 20. This simply rounds the number of segments down to the greatest integer if it is fractional.

$$N_{segments} = \text{floor}\left(\frac{2}{3}N_{stations} + \frac{4}{3}\right) \tag{14}$$

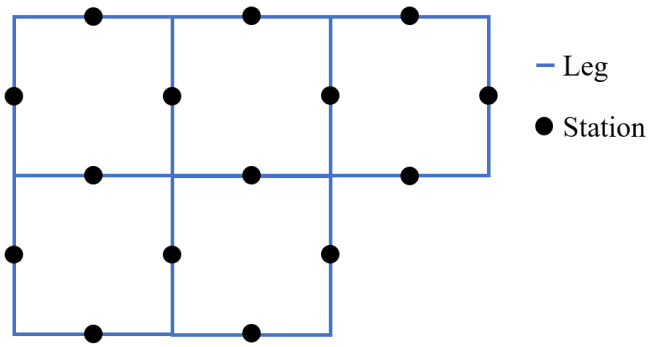


Figure 20—Asymmetric ATN route configuration.

Since ATN routes may have configurations that are less uniform than the example in Figure 20, Equation (14) only serves as an approximation. Thus, the average distance between stations (d_s) is the route length (L_{route}) divided by the approximate number of route segments between stations.

$$d_s = \frac{L_{route}}{N_{segments}} = \frac{L_{route}}{\text{floor}\left(\frac{2}{3}N_{stations} + \frac{4}{3}\right)} \tag{15}$$

2.1.2 Irradiance Model

The purpose of the irradiance model is to determine the amount of irradiance that is incident upon the PV canopy. Appendix D features the irradianceModel() function for reference, which is executed in the following five steps:

1. Acquisition of Weather Data
2. Time Array Construction
3. Sun Position Modeling
4. Plane of Array (POA) Irradiance Modeling
5. Incorporation of Soiling Losses

2.1.2.1 Acquisition of Weather Data

Weather data is the foundation for the irradiance model because it describes the solar resource. Among other data, weather files generally have DNI, DHI, and GHI values for all 8,760 hours of a non-leap year. This information is derived from either weather station measurements, satellite data, or a combination of the two [56]. Since the

listed irradiance values will never be exact predictions for the future, the most reliable weather data is based on measurements recorded over many years [12]. This type of weather file is called a typical meteorological year (TMY) and is comprised of weather data from the 12 months in the entire multiyear data set that most accurately represent typical conditions [57]. TMY weather files are freely accessible from the NSRDB, the European Commission's Photovoltaic Geographical Information System [58], and other sources listed in [56]. The SAM 2017 download [22] also includes a library of almost 1,700 TMY weather files from around the world.

The NSRDB's latest model is the Physical Solar Model, which utilizes meteorological data from four geostationary satellites [40]. This allows weather files to be created at a spatial resolution of 4 × 4 km between latitudes of 60° and -20° in the Western Hemisphere [40]. The data spans the years 1998 to 2016 and is updated annually to maintain reliable TMYs [40]. A validation study with nine ground measurement stations across the Unites States yielded maximum bias errors of ±5% for GHI and ±10% for DNI [59].

The getNSRDBweather() function was created to download weather data through the NSRDB API [60]. A request string is constructed to specify the coordinates and format for the weather file, which is downloaded in response to the call. The weather file is set to be an hourly TMY in local time that contains DNI, DHI, GHI, and albedo data. The getNSRDBweather() function is listed in Appendix E.

The irradiance modeling algorithm begins with the acquisition of a weather file. The user can either specify a weather file or leave the field blank. The specified weather file can be from the SAM solar resource library or the NSRDB. Four different types of

weather files exist in the SAM solar resource library and they are all compatible. The weather data is loaded if the file exists, otherwise, a search is conducted for a previously saved NSRDB file at the desired coordinates. If found, the matching NSRDB file is loaded. If not, the getNSRDBweather() function is used to download the appropriate file. This new file is saved in CSV format to avoid future API calls on repeated coordinates. Since some weather files do not contain albedo information, missing values are defaulted to 0.2 [10], [41]. The mean square wind speed is also computed for use in the energy demand model, which is the mean of the squared hourly wind speed values.

2.1.2.2 Time Array Construction

Since the irradiance model is dependent on the time of day, a time array is created as a shell for the subsequent sun position calculations. The irradiance modeling algorithm operates on non-leap years in standard local time, in accordance with the weather data. The time array is built for every hour of the year, listing the year, month, day, and time. The listed year is the current year, which is decremented if this is a leap year.

2.1.2.3 Sun Position Modeling

The sun position algorithm involves computing the position of the sun in the sky for every hour of the year. This position is described in spherical coordinates with the elevation (α), zenith (Z), and azimuth (A_S) angles. The elevation is the angle between the sun vector and the horizontal plane. The zenith angle is the complement of the elevation angle, measured from the vertical zenith axis. The azimuth is the angle measured

clockwise from true north to the projection of the sun vector on the horizontal plane. These angles are illustrated in Figure 21.

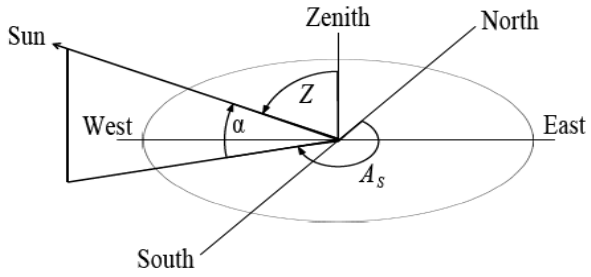


Figure 21—Illustration of the sun elevation (α), zenith (Z), and azimuth (A_S) angles. Adapted from: P. Gilman et al., 2018 [41]

Several methods exist for determining hourly sun positions, each with varying complexity and accuracy. The following four methods were compared to identify the model that should be used: Solar Position Algorithm (SPA) [61], SAM Physical Trough Model (SPTM) [62], Power From The Sun Woolf Model (PFTSW) [63], and Power From The Sun Lamm Model (PFTSL) [63]. These models were also compared with the outputs from the Solrad and SAM simulation programs. The hourly sun positions from each model were used to compute the annual incident DNI on a south-facing module of 37.37° tilt and latitude. The SPA model was set as the benchmark for comparison, as it has a validated sun angle accuracy of ±0.0003° [61]. Table 1 shows the results from the study, indicating close agreeance across all models. The Solrad and SAM simulation values compared closest with the SPA results, verifying the selection of this model as the benchmark. The SPTM had the largest error, while the PFTSL and PFTSW models had equal errors. The PFTSW model was selected for implementation because it is the most computationally efficient and has an acceptable accuracy.

Table 1—Comparison of sun position models.

Model	Annual Incident DNI (Wh/m^2)	Percent Error
SPA	1,775,534	-
SPTM	1,771,419	0.232%
PFTSL	1,774,357	0.066%
PFTSW	1,774,368	0.066%
Solrad	1,775,552	-0.001%
SAM	1,776,230	-0.039%

The sun position algorithm is subsequently detailed, according to the PFTSW model in [63]. The objective of these computations is to generate the solar elevation and azimuth angles for each hour of the year. The first step is to convert the local standard time (LST) to true solar time (TST). LST is the standard clock time in the local time zone, whereas TST is based on the position of the sun in the sky. Noon in TST is when the sun is perfectly due south when viewed from the Northern Hemisphere, or due north when viewed from the Southern Hemisphere, regardless of clock time. Before conversion, the LST must be modified if the weather file is from the SAM solar resource library. This is because the associated irradiance values are averaged over each hourly time step [64], necessitating the sun position to be averaged accordingly. NSRDB weather files have instantaneous irradiance values for the beginning of each time step [64], so no correction is necessary. The sun position algorithm detects the use of SAM weather files and adjusts the LST to the average sun-up value for each time step.

To convert from LST to TST, a longitude correction (LC) and equation of time (EOT) must be applied. The LC is represented in hours and adjusts the LST to be dependent on longitude instead of a politically defined time zone. The time zone

parameter in Equation (16) is the number of hours that the location is ahead-of or behind Coordinated Universal Time (e.g., -8 hours for California). This must be defined by the user.

$$LC = time\ zone - \frac{\psi}{15} \tag{16}$$

The EOT is an empirical relation that compensates for the factors that cause the length of a day to deviate from 24 hours. The Woolf EOT is implemented from [63], [65] as a function of the number of days since January 1 (N). The units of variable x and the EOT are degrees and minutes, respectively.

$$EOT = 0.258\cos(x) - 7.416\sin(x) - 3.648\cos(2x) - 9.228\sin(2x) \tag{17}$$

$$\text{where,} \qquad x = \frac{360(N-1)}{365.242}$$

The LST can be converted to TST by adding the EOT, in hours, and subtracting the LC.

$$TST = LST + \frac{EOT}{60} - LC \tag{18}$$

With the TST determined, the hour angle (ω) is calculated as the earthly rotation, in degrees, for the associated solar time.

$$\omega = 15(TST - 12) \tag{19}$$

The declination angle (δ) of the earth describes the tilt of the equatorial plane relative to the orbital plane. It is listed in degrees and computed in function of the day number.

$$\delta = \sin^{-1}\{0.39795\cos[0.98563(N-173)]\} \quad (20)$$

The elevation angle, represented in degrees from the horizon, is calculated in function of the declination angle, latitude, and hour angle.

$$\alpha = \sin^{-1}[\sin(\delta)\sin(\phi) + \cos(\delta)\cos(\omega)\cos(\phi)] \quad (21)$$

Finally, the solar azimuth angle is determined, represented in degrees clockwise from true north. This requires the use of a test condition to ensure the azimuth is assigned to the correct quadrant. The untested azimuth (A_S') is calculated with the declination angle, latitude, hour angle, and elevation angle. It is then tested and corrected as follows.

$$A_S' = \cos^{-1}\left[\frac{\sin(\delta)\cos(\phi) - \cos(\delta)\cos(\omega)\sin(\phi)}{\cos(\alpha)}\right] \quad (22)$$

If: $\sin(\omega) > 0 \quad \rightarrow \quad A_S = 360° - A_S'$

Else: $A_S = A_S'$

These calculations are repeated for each of the 8,760 time steps in the weather file. The resulting solar elevation and azimuth angles are important parameters for the subsequent steps.

2.1.2.4 Plane of Array (POA) Irradiance Modeling

With hourly weather data and the associated sun positions, the plane of array (POA) irradiance can be determined. POA irradiance is the amount of incident irradiance

on the surface of a PV array. It is comprised of three components: direct, diffuse, and ground-reflected.

The direct POA irradiance $(G_{dir,A})$ is calculated in function of the angle of incidence (γ), according to [21].

$$G_{dir,A} = DNI \cos(\gamma) \tag{23}$$

The cosine of the angle of incidence is equivalent to the dot product of the array normal and sun position unit vectors. The resulting expression allows the direct POA irradiance to be calculated with the DNI, array tilt (β), array azimuth (A_A), sun elevation (α), and sun azimuth (A_S) angles [21]. These parameters are illustrated in Figure 22.

$$G_{dir,A} = DNI\,[\sin(\beta)\cos(\alpha)\cos(A_A - A_S) + \cos(\beta)\sin(\alpha)] \tag{24}$$

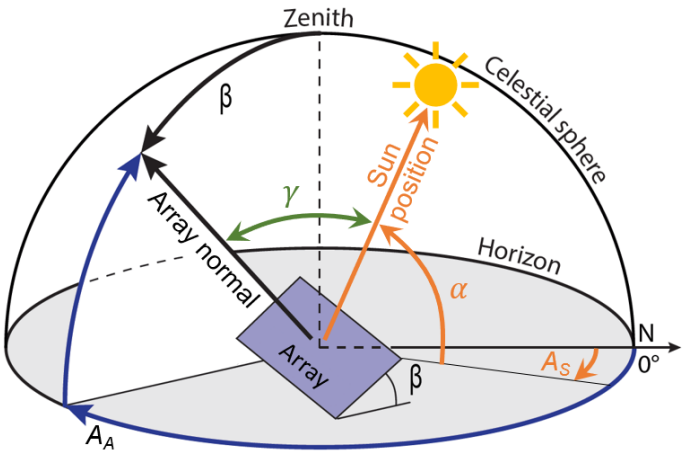


Figure 22—Illustration of the direct POA irradiance parameters. The array normal is described by the array azimuth (A_A) and tilt (β) angles. The sun position is described by the sun azimuth (A_S) and elevation (α) angles. The angle of incidence (γ) between the array normal and sun position vector is also included. Adapted from: K. Jäger et al., 2014 [21]

Two physical conditions must be applied to Equation (24). The first is that there will be no DNI if the sun is below the horizon. The second is that no DNI will be incident if the sun is behind the collecting face of the array. Therefore, if either the sun elevation or the cosine of the angle of incidence is negative, the direct POA irradiance is set to zero.

Several methods exist for computing the diffuse POA irradiance $\left(G_{dif,A}\right)$, such as the isotropic model [10], Hay, Davies, Klucher, Reindl (HDKR) model [66], and Perez model [67]. A validation study indicated the Perez model was 60% more accurate than the isotropic model, and 33% more accurate than the HDKR model [68]. It was thus selected for the diffuse POA irradiance calculations and is subsequently detailed according to [41]. The first step of the model is to define the sun zenith angle (Z) in degrees, as well as parameters a and b.

$$Z = 90° - \alpha \qquad a = \max[0, \cos(\gamma)] \qquad b = \max[\cos(85°), \cos(Z)]$$

The sky clearness parameter (ε) is calculated as follows.

$$\varepsilon = \frac{(DHI + DNI)/DHI + (5.534 \times 10^{-6})Z^3}{1 + (5.534 \times 10^{-6})Z^3} \tag{25}$$

The air mass ratio (AM) is then computed to represent the relative length of atmosphere that the irradiance passes through. By definition, an air mass ratio of unity represents the shortest atmospheric distance, which is when the sun is directly overhead.

$$AM = \frac{1}{\cos(Z) + 0.15(93.9 - Z)^{-1.253}} \tag{26}$$

The sky brightness parameter (Δ) is calculated with the assumption that the extraterrestrial irradiance is 1,367 W/m^2.

$$\Delta = DHI \frac{AM}{1367\ W/m^2} \tag{27}$$

The empirical f coefficients are then determined from Table 2, based on the sky clearness parameter.

Table 2—Empirical *f* coefficients for Perez diffuse model.

	f_{11}	f_{12}	f_{13}	f_{21}	f_{22}	f_{23}
$\varepsilon \leq 1.065$	-0.0083117	0.5877285	-0.0620636	-0.0596012	0.0721249	-0.0220216
$\varepsilon \leq 1.23$	0.1299457	0.6825954	-0.1513752	-0.0189325	0.065965	-0.0288748
$\varepsilon \leq 1.5$	0.3296958	0.4868735	-0.2210958	0.055414	-0.0639588	-0.0260542
$\varepsilon \leq 1.95$	0.5682053	0.1874525	-0.295129	0.1088631	-0.1519229	-0.0139754
$\varepsilon \leq 2.8$	0.873028	-0.3920403	-0.3616149	0.2255647	-0.4620442	0.0012448
$\varepsilon \leq 4.5$	1.1326077	-1.2367284	-0.4118494	0.2877813	-0.8230357	0.0558651
$\varepsilon \leq 6.2$	1.0601591	-1.5999137	-0.3589221	0.2642124	-1.127234	0.1310694
$\varepsilon > 6.2$	0.677747	-0.3272588	-0.2504286	0.1561313	-1.3765031	0.2506212

Source: P. Gilman et al., 2018 [41]

The F_1 and F_2 coefficients are determined from the sky clearness, sky brightness, empirical f coefficients, and the zenith angle in radians.

$$F_1 = \max(0, f_{11} + \Delta f_{12} + Z f_{13}) \tag{28}$$

$$F_2 = f_{21} + \Delta f_{22} + Z f_{23} \tag{29}$$

The isotropic (D_i), circumsolar (D_c), and horizon brightening (D_h) components of the diffuse irradiance are then computed. The model is switched to isotropic when the zenith angle is greater than 87.5°.

For $0° \leq Z \leq 87.5°$:

$$D_i = DHI(1 - F_1)\frac{1 + \cos(\beta)}{2}, \qquad D_c = DHI\ F_1\frac{a}{b}, \qquad D_h = DHI\ F_2 \sin(\beta) \tag{30}$$

For $Z > 87.5°$:

$$D_i = DHI\ \frac{1 + \cos(\beta)}{2}, \qquad D_c = 0, \qquad D_h = 0 \tag{31}$$

Finally, the three components are summed for the diffuse POA irradiance.

$$G_{dif,A} = D_i + D_c + D_h \tag{32}$$

The ground-reflected POA irradiance $\left(G_{ref,A}\right)$ is the final component of sunlight reaching the surface of the array. Ground-reflected irradiance is diffuse and can be described in function of the albedo (ρ), DNI, elevation angle, DHI, and array tilt, according to [10].

$$G_{ref,A} = \rho[DNI \sin(\alpha) + DHI] \sin^2\left(\frac{\beta}{2}\right) \tag{33}$$

The nominal POA irradiance (G_A) is the sum of the direct, diffuse, and ground-reflected components.

$$G_A = G_{dir,A} + G_{dif,A} + G_{ref,A} \tag{34}$$

The calcPOA() function was created to make POA calculations more modular within the simulation algorithm. It is listed in Appendix F.

2.1.2.5 Incorporation of Soiling Losses

As PV modules operate, dust particles accumulate on the surface and cause partial shading of the cells. This reduces the incident irradiance on the cells, resulting in an associated reduction of power output. This loss is termed *soiling* and is dependent on many factors that include wind patterns, rain frequency, module tilt, electrostatic particle attraction, particle size, particle transmittance, and particle geometry [69].

The soiling loss algorithm was developed from the results of studies conducted by Cano et al. [70] and Nahar and Gupta [71]. The Cano et al. study measured the soiling loss associated with modules at tilts ranging from 0° to 40° [70]. The results were modified to be more conservative by adding a soiling loss of 2.0% to all values. This was done because the study was conducted during the winter months of January through March. There were six rain events throughout the duration of the experiment that temporarily eliminated or drastically reduced the soiling loss. Consequently, the reported average soiling loss values were likely lower than the annual average values that include the dry season. Thus, the 2.0% bias increases the soiling loss toward values that are expected to be more frequent throughout the rest of the year. For reference, the biased average value for each tilt is approximately 85% of the associated peak value. To account for steeper tilts, the results from Nahar and Gupta were applied. This study concluded that the soiling losses of glass plates tilted at 45° and 90° were 72% and 30% of the horizontal configuration value, respectively [71]. These results were plotted

alongside the biased measurements from Cano et al. and a third-order polynomial curve fit was applied, shown in Figure 23. The coefficient of determination was 0.97, indicating an acceptable fit.

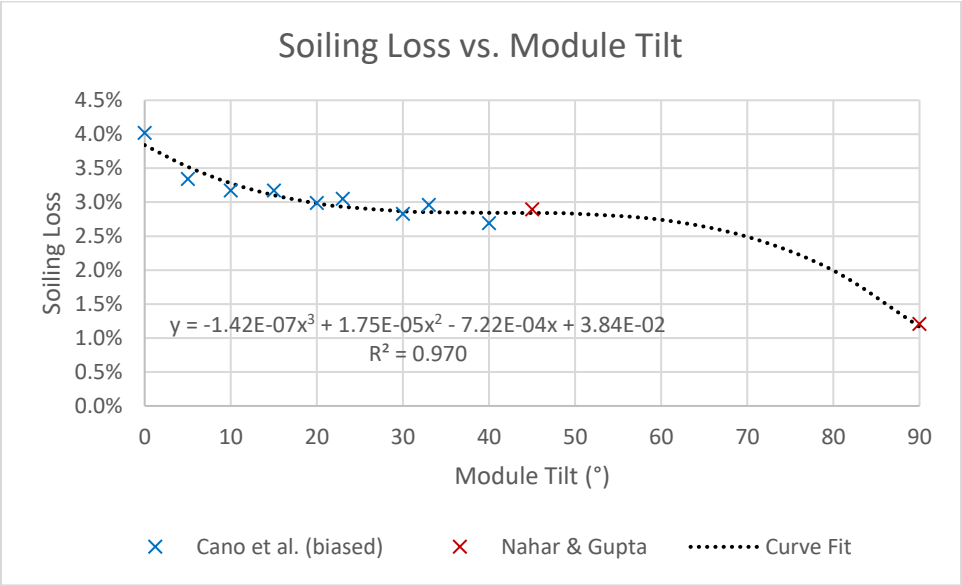


Figure 23—Curve fit correlation between soiling loss and module tilt.

Sources: Cano et al., 2014 [70]; Nahar and Gupta, 1990 [71]

The steep reduction of soiling loss at low module tilts was attributed to the improved rainwater flow that occurs as the module tilt is increased from horizontal, as observed in [70]. The soiling loss plateaus through the mid-range tilts and decreases rapidly as the tilt approaches 90°. This second steep reduction was attributed to the diminution of the static friction coefficient between the particles and module surface, which decreases with the cosine of the tilt angle. The curve fit correlation from Figure 23 is implemented at the end of the calcPOA() function to determine the soiling loss (k_{soil})

associated with the tilt of the array. This soiling loss value is applied to the three POA irradiance components to return the soiled irradiance incident upon the array.

$$k_{soil} = (-1.42 \times 10^{-7})\beta^3 + (1.75 \times 10^{-5})\beta^2 - (7.22 \times 10^{-4})\beta + (3.84 \times 10^{-2}) \quad (35)$$

2.1.3 Energy Demand Model

The purpose of the energy demand model is to determine the annual energy demand of the transit route. This provides a basis for the sizing computations performed in the canopy simulation and optimization algorithm. The average trip duration, average system power demand, and number of vehicles operating on the route are also returned. The energyDemand() function is presented in Appendix G and is subsequently discussed in three sections:

1. Trip Time Equation
2. Transit Energy Equation
3. Computation of Outputs

2.1.3.1 Trip Time Equation

The trip time equation is used to calculate the elapsed time during vehicle transit from one location to another. In the context of ATNs and other transit systems with prescribed routes, it is the elapsed time during transit between two stations. The trip time (t_s) is comprised of an acceleration interval, constant velocity interval, deceleration interval, and dwell interval, as detailed by Anderson in [72]. It can be computed with Equation (36) from [72].

$$t_s = t_D + \frac{d_s}{v_L} + \frac{v_L}{a_m} + \frac{a_m}{J_1} + \frac{a_m^3}{24 v_L}\left(\frac{1}{J_2^2} - \frac{1}{J_1^2}\right) \tag{36}$$

The variables of Equation (36) are described as follows:

- t_D – Dwell time: Time that the vehicle is stopped at the station during passenger loading and unloading. This value is typically between 0 and 40 seconds [72].
- d_s – Average trip distance: Average distance that the vehicle travels between two stations. This term is computed in the loadRoute() function for the most energy intensive scenario of stopping at every station.
- v_L – Line speed: Operating speed of the fully accelerated vehicle.
- a_m – Maximum acceleration: Maximum acceleration of the vehicle. This is set to $0.25g$, which is the upper limit for rider comfort in a vehicle with seated passengers [72].
- J – Jerk: Time derivative of acceleration. J_1 represents the rate at which a_m is reached. J_2 represents the rate at which a_m diminishes to zero. J should be maximized to minimize the acceleration interval but should not exceed a_m for passenger comfort [72]. For transit systems with off-line stations, such as ATNs, J_1 is typically equal to J_2 [72]. Thus, it is acceptable to let $J_1 = J_2 = a_m$ for ATNs.

Equation (36) can be simplified by applying the ATN approximation that $J_1 = J_2 = a_m$.

$$t_s = t_D + \frac{d_s}{v_L} + \frac{v_L}{a_m} + 1 \tag{37}$$

2.1.3.2 Transit Energy Equation

The transit energy equation is used to calculate the energy $\left(E(t_s)\right)$ required to move a single vehicle from one station to another in the trip time. It was developed by Anderson in [72] and is presented as Equation (38). From left to right, the terms on the right side of the equation represent the energy required to overcome the inertial resistance, aerodynamic drag, rolling resistance, grade resistance, and auxiliary load.

$$E(t_s) = \frac{1}{\overline{\eta}_m}\left\{\left(1-\eta_{regen}\right)N_T\frac{m_v v_L^2}{2} + \frac{1}{2}\rho_{air}C_D A_v\left[\left(v_L^2 + \langle v_w^2\rangle\right)d_s - \frac{v_L^4}{2a_m}\right]\right.$$
$$\left. + N_T m_v\left[\frac{C_{srr} g d_s}{R_w} + C_{drr} v_L\left(d_s - \frac{v_L^2}{3a_m}\right) + \Delta h_{avg} g\right]\right\} + N_T P_{aux} t_s \qquad (38)$$

In a study on the energy demand of ATNs, Furman modified Equation (38) by inserting a correction factor to increase the static rolling resistance term [73]. This was done to incorporate the rolling resistance of the switching wheels and wayside pickup shoes—both characteristic features of ATN vehicles [73]. The modified transit energy equation with the static rolling resistance correction factor (k_{srr}) is presented as Equation (39).

$$E(t_s) = \frac{1}{\overline{\eta}_m}\left\{\left(1-\eta_{regen}\right)N_T\frac{m_v v_L^2}{2} + \frac{1}{2}\rho_{air}C_D A_v\left[\left(v_L^2 + \langle v_w^2\rangle\right)d_s - \frac{v_L^4}{2a_m}\right]\right.$$
$$\left. + N_T m_v\left[k_{srr}\frac{C_{srr} g d_s}{R_w} + C_{drr} v_L\left(d_s - \frac{v_L^2}{3a_m}\right) + \Delta h_{avg} g\right]\right\} + N_T P_{aux} t_s \qquad (39)$$

The variables of Equation (39) are subsequently detailed:

- $\overline{\eta}_m$ – Average motor efficiency: Average efficiency of the vehicle motor.

- η_{regen} – Regenerative braking efficiency: Proportion of kinetic energy recovered by the regenerative braking system during a single trip.
- N_T – Number of vehicles in a train: Quantity of connected vehicles traveling together in a train. Since ATN vehicles operate independently, this term is set to one.
- m_v – Vehicle mass: Mass of a single vehicle.
- ρ_{air} – Density of air: This term is set to the 15 °C value of 1.225 kg/m^3 at standard atmospheric pressure [74].
- C_D – Drag coefficient: Empirical coefficient for the aerodynamic drag calculation.
- A_v – Frontal area: Frontal area of a single vehicle.
- $\langle v_w^2 \rangle$ – Mean square wind speed: Average of the squared hourly wind speed values. This term is computed in the irradianceModel() function from the loaded weather data.
- k_{srr} – Static rolling resistance correction factor: Correction factor applied in the static rolling resistance calculation to incorporate the additional friction forces from the switching wheels and wayside pickup shoes of an ATN vehicle. This term is set to 1.2, according to [73].
- C_{srr} – Dimensioned static rolling resistance coefficient: Empirical coefficient for the static rolling resistance calculation, dimensioned in units of length. Typical values are 0.00053 m for cast iron on steel and 0.00076 m – 0.0014 m for polyurethane on steel [73].
- g – Acceleration due to gravity: This term is set to 9.81 m/s^2.

- R_w – Wheel radius: Radius of the wheels on the vehicle.
- C_{drr} – Dimensioned dynamic rolling resistance coefficient: Empirical coefficient for the dynamic rolling resistance calculation, dimensioned in units of force per mass-velocity. This is set to the conservative value of 0.0004935 N/(kg-m/s) from [75].
- Δh_{avg} – Average elevation change: Change in elevation for an average station-to-station trip. For closed-loop transit networks like ATNs, the net change in elevation along the route is zero. Thus, the change in elevation of each trip averages to zero.
- P_{aux} – Auxiliary power: Auxiliary power demand of one vehicle (e.g., load from air conditioning unit, cabin lights, automatic doors, etc.).

2.1.3.3 Computation of Outputs

The four outputs of the energy demand model are the average trip duration, number of vehicles operating on the route, average system power demand, and annual system energy demand. The average trip duration is computed from Equation (37) with user-defined values for the line speed and dwell time. The number of vehicles operating on the route (N_v) is determined for the maximum practical route usage of 70% (i.e. 70% of the route is occupied by vehicles and their associated headways) [73]. This is calculated in function of the route length (L_{route}), line speed, and headway time (t_h). The floor() function is used to round the result down to the greatest integer if it is fractional.

$$N_v = \text{floor}\left(0.7\frac{L_{route}}{v_L t_h}\right) \tag{40}$$

The average system power demand $\left(P_{dem,avg}\right)$ is the single vehicle trip energy divided by the trip time, scaled up by the number of vehicles operating on the route.

$$P_{dem,avg} = N_v \frac{E(t_s)}{t_s} \tag{41}$$

Finally, the annual system energy demand (E_{dem}) is the product of the average system power demand and the number of hours in a year, assuming 24/7 operation.

$$E_{dem} = \left(8760 \frac{hr}{yr}\right) P_{dem,avg} \tag{42}$$

2.1.4 Shading Analysis

The purpose of the shading analysis algorithm is to determine the proportion of each leg and station canopy that is shaded by the surrounding buildings. This shaded proportion is termed the *shaded ratio* and is computed for each hourly time step of the annual cycle. The shadingAnalysis() function is featured in Appendix H and is executed in the following four steps:

1. Acquisition of Building Height Data
2. Coordinate System Transformation
3. Shadow Modeling
4. Computation of Shaded Ratios

2.1.4.1 Acquisition of Building Height Data

Since transit systems operate in urban areas, the dominant sources of shading are the surrounding buildings. The necessary parameters for modeling the shadow of each building are the building footprint, height, and distance relative to the transit route. This information is most readily accessible through the municipal website of the associated city, typically in a common geographic information system file format called a shapefile [76]. Shapefiles can be read with MATLAB through the Mapping Toolbox Add-On [77], which is required for the execution of the shadingAnalysis() function. For this reason, the shading analysis is optional, and its omission defaults all canopies to the fully unshaded state.

The shapefile is selectively imported to load information for buildings that are taller than the PV canopy and within a specific proximity. The proximity is defined by a 0.0015° longitude and latitude offset beyond the most north, south, east, and west vertices of the route. This spherical offset is approximately equivalent to one and a half city blocks, ensuring all surrounding buildings are loaded. Each building footprint is then simplified by retaining only the most north, south, east, and west vertices. The purpose for this simplification is to reduce the complexity of the algorithm and the associated computing time. Figure 24 shows an example of the imported and simplified building footprints, shown in yellow and green, respectively. The PV canopies are represented in black for reference.

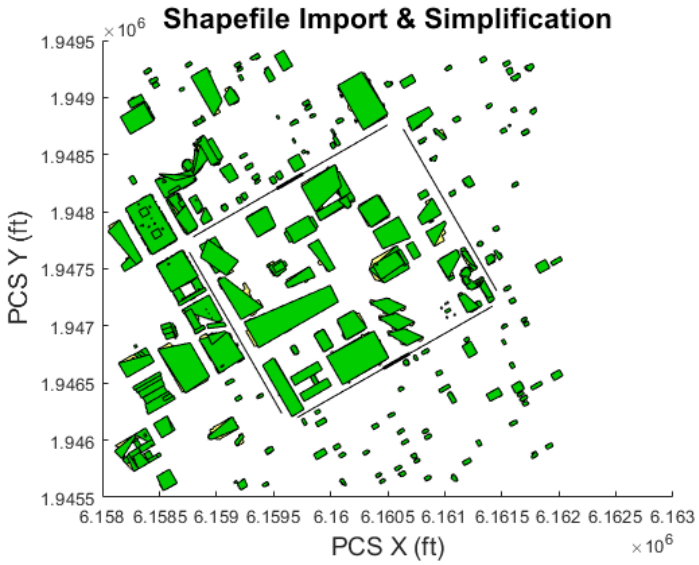


Figure 24—Imported (yellow) and simplified (green) building footprints.

2.1.4.2 Coordinate System Transformation

Shapefiles are often generated in a projected coordinate system (PCS) with linear coordinates on an X-Y axis. Up to this point, the route has been represented in the geographic coordinate system (GCS), marked by spherical longitude and latitude values. To perform the shading analysis, the GCS route coordinates must be transformed to the PCS of the shapefile. This is conducted through the ESPG.io API [78], which uses projection codes to represent the source and target coordinate systems. These codes are sent with the source coordinates as a query, and the transformed coordinates are returned. The .PRJ file associated with the shapefile lists the titles of the corresponding PCS and GCS, which can be pasted into the search function of [79] to identify the appropriate projection codes.

2.1.4.3 Shadow Modeling

With the route represented alongside the building footprints in the PCS, the next step is to model the shadows. For each building, the outer bounds of the shadow are computed as vectors connected to the vertices of the simplified footprint. The length of each vector (L_{shadow}) is represented in function of the sun elevation angle (α) and elevation difference between the building (h_b) and canopy (h_c), according to [63]. Note that the canopy height must be represented in the unit system of the PCS, which may differ from that of the previous computations.

$$L_{shadow} = \frac{h_b - h_c}{\tan \alpha} \tag{43}$$

The azimuth of each shadow vector (A_{shadow}) is calculated by subtracting 180° from the sun azimuth angle (A_S) [63].

$$A_{shadow} = A_S - 180° \tag{44}$$

The X- and Y-coordinates of the end point of each shadow vector (X_{shadow}, Y_{shadow}) are determined from the building footprint vertex coordinates (X_{vertex}, Y_{vertex}) and the shadow length and azimuth angle.

$$X_{shadow} = L_{shadow} \sin A_{shadow} + X_{vertex} \tag{45}$$

$$Y_{shadow} = L_{shadow} \cos A_{shadow} + Y_{vertex} \tag{46}$$

Finally, the shadow bounds are completed by inserting four additional vectors. These simply connect the end points that are determined from Equation (45) and

Equation (46). Figure 25 illustrates how these vectors, shown in grey, are interconnected to represent the outer bounds of the shadow.

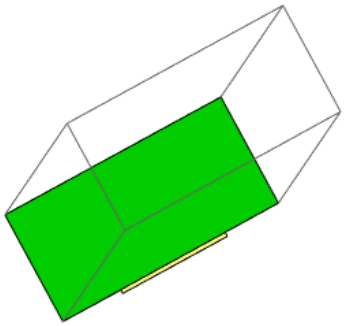


Figure 25—Example of vector representation of shadow bounds.

2.1.4.4 Computation of Shaded Ratios

The shaded regions of the leg and station canopies can be identified from the intersection points with the shadow vectors. These points are computed with the polyxpoly() function for each shadow, which returns the intersection points between the canopy line and shadow vectors. The canopy line is then extended westward and the intersection points are computed again. If new intersection points are found, the shadow extends beyond the west vertex of the canopy. In this case, the western shadow intersection point is set to the west canopy vertex. This process is repeated for the east vertex of the canopy. Both canopy extensions are 305 meters, which is a conservative length selected to exceed the shadow width of most buildings. The outer pair of intersection points of each shadow mark the associated shaded regions on the canopy. These values are stored in an array for processing.

The array of outer intersection points is processed with an algorithm that combines shaded regions if they overlap. This is done by comparing the locations of the outer intersection points of two different shadows at a time. If they overlap, the two outermost intersection points are stored, and the two innermost points are truncated. If no overlap is detected, another comparison is made. This process continues until all possible shadow comparisons have been made, yielding an array of end points for each set of overlapping shadows.

The processed array of shadow end points (X_1, Y_1, X_2, Y_2) is used to determine the length of each shaded region on the canopy. These lengths are summed and divided by the canopy length $\left(L_{canopy}\right)$ to return the shaded ratio (SR). This value is stored in the associated leg or station table and the entire shading computation is repeated for the next hourly time step.

$$SR = \frac{1}{L_{canopy}}\left[\sum_{i=1}^{N_{shaded\ regions}} \sqrt{\left(X_{2,i} - X_{1,i}\right)^2 + \left(Y_{2,i} - Y_{1,i}\right)^2}\right] \tag{47}$$

Figure 26 presents a visualization of this computation for the arbitrary instance of 2:00 PM on January 1. The shaded regions are marked in red on the black canopies.

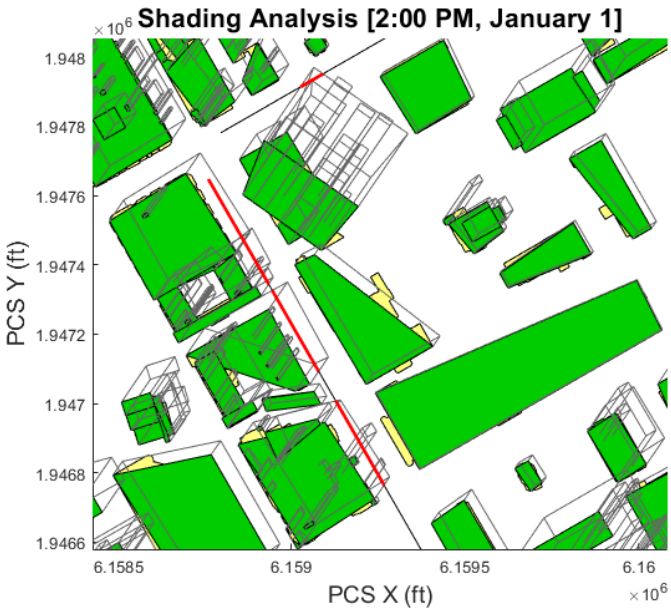


Figure 26—Visualization of the shaded regions (red) on the canopies (black).

As a final step, the annual average shaded ratio (SR_{avg}) is determined for each canopy as the sum of the hourly shaded ratios divided by the number of hours that the sun is above the horizon $(N_{sun\ up\ hrs})$. This value indicates the average proportion of the canopy that is shaded throughout the year, which serves as a metric for comparing the solar resource of different canopies.

$$SR_{avg} = \frac{1}{N_{sun\ up\ hrs}} \left(\sum_{i=1}^{8760} SR_i \right) \tag{48}$$

2.1.5 Array Profile Design Studies

Four array profile design studies were conducted to quantify the influence of array parameters on the quantity of annual soiled POA irradiation. All studies were conducted for arrays in the Northern Hemisphere, though the results are valid for the Southern Hemisphere due to symmetry. The results of each study were used to design the four

array profile options for the canopy simulation and optimization algorithm. An additional study was conducted to investigate the benefits of using different array profiles for canopies with different bearings. The five studies are subsequently listed, and the simulation algorithms are included in Appendix I:

1. Tilted Planar Array Study
2. Tilted Curved Array Study
3. Horizontal Planar Array Study
4. Horizontal Curved Array Study
5. Dual Array Profile Study

2.1.5.1 Tilted Planar Array Study

The tilted planar array is the simplest of the four profiles, comprised of uniformly oriented modules. The only design variable that can be optimized for irradiance collection is the tilt angle, since the azimuth angle is defined by the bearing of the canopy. Figure 27 is an example of a tilted planar array profile with six modules.

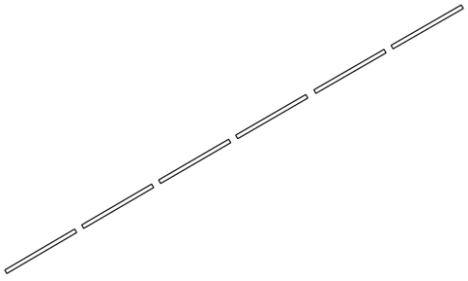


Figure 27—Tilted planar array profile.

A study was conducted to investigate the influence of latitude and weather on the optimal tilt angle. The methodology entailed computing the annual soiled POA irradiation on a module surface of varying tilt and azimuth angle. The tilt was adjusted from 0° to 90° at 1° increments, at azimuth angles ranging from 90° to 270° at 10° increments. The tilt angle associated with the maximum annual soiled POA irradiation value was identified and plotted for each azimuth angle. This was done for three locations of latitude 0°, 37°, and 60° with associated weather data from the NSRDB. The results are shown in Figure 28.

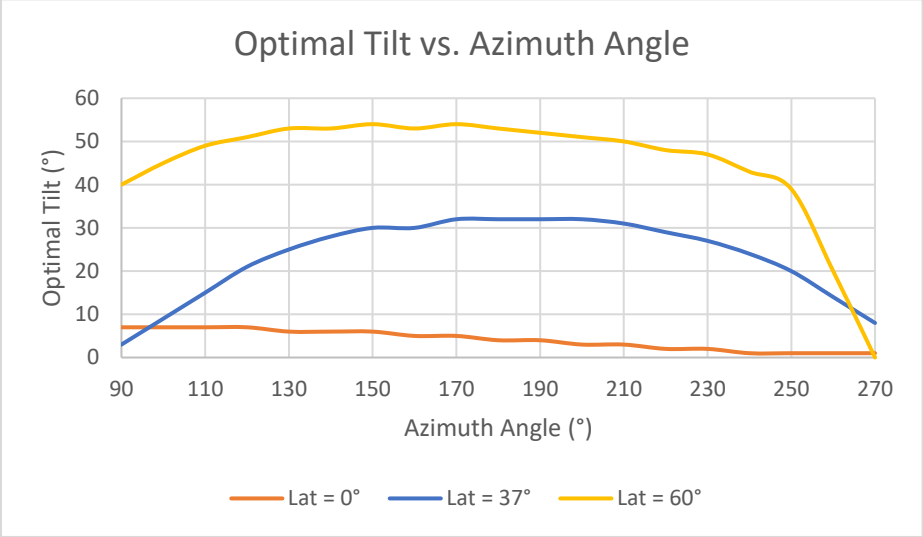


Figure 28—Optimal array tilt in function of azimuth angle and location.

The first observation is that the optimal tilt for south-facing modules tends to approximate the latitude of the location. For the 37° and 60° latitude plots, the optimal tilts were less than the latitude values to collect more irradiance in the summer months when the sun elevation angles are greater. The summer months are favored because they have longer days and clearer skies. The optimal tilts of the 0° latitude plot were

consistently greater than the latitude value to mitigate soiling losses associated with rainwater retention, as observed in [70].

The second observation is that the optimal tilts for east- and west-facing modules have no correlation with the latitude of the location. After examining the weather files, these optimal tilt values were observed to be dependent on the amount of DNI in the mornings and evenings. For example, the 0° latitude location had large DNI values in the mornings. This caused larger optimal tilts for east-facing modules to better orient them toward the rising sun.

Despite the general tendency toward the latitude value, it was concluded that the optimal tilt of the modules is strongly influenced by the local weather conditions. Thus, the canopy simulation and optimization algorithm will optimize the tilt of the planar array profile for the location and weather conditions of each route.

2.1.5.2 Tilted Curved Array Study

The tilted curved array is a variation of the tilted planar profile, designed to improve aesthetics. It is parameterized by two variables: the tilt increment and nominal tilt. The tilt increment ($\Delta\beta$) is the angular displacement between modules and is constant throughout the array. The nominal tilt (β_{nom}) is the tilt of the planar array from which the curved array deviates. Both angles are illustrated in Figure 29.

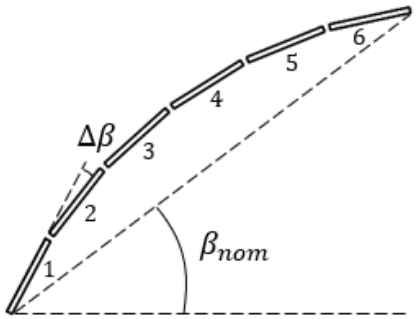


Figure 29—Tilt increment ($\Delta\beta$) and nominal tilt (β_{nom}) of the curved array profile.

The tilt of each module (β_i) in the curved array profile is determined in function of the nominal tilt, tilt increment, number of modules in the profile (N_{mod}), and array index (i). The numbering convention for the array index is represented in Figure 29.

$$\beta_i = \beta_{nom} + \frac{\Delta\beta}{2}(N_{mod} - 2i + 1) \tag{49}$$

As with the tilted planar array profile, the nominal tilt should be optimized for the location and weather conditions of each route. A series of studies were conducted to determine an appropriate tilt increment value. In the first study, the annual soiled POA irradiation was computed with the calcPOA() function for a six-module, south-facing array. The tilt increment was varied from 0° to 10° at 0.5° increments, and each computed irradiation value was normalized to the relative difference from the maximum value in the series. This was conducted for nominal tilt angles of 0° to 60° at 10° increments. The results indicated that the collected irradiation was most sensitive to the tilt increment when the nominal tilt was approximately equal to the latitude of the location. The study was repeated with the nominal tilt set to the latitude of the location and the azimuth angle was varied from 90° to 270° at 30° increments. The collected

irradiation was most sensitive to the tilt increment when the azimuth angle was 180°. Finally, the study was repeated for the most sensitive scenario of a south-facing array nominally tilted at the latitude angle. The number of modules was varied from 2 to 12 at 2-module increments and the results are presented in Figure 30.

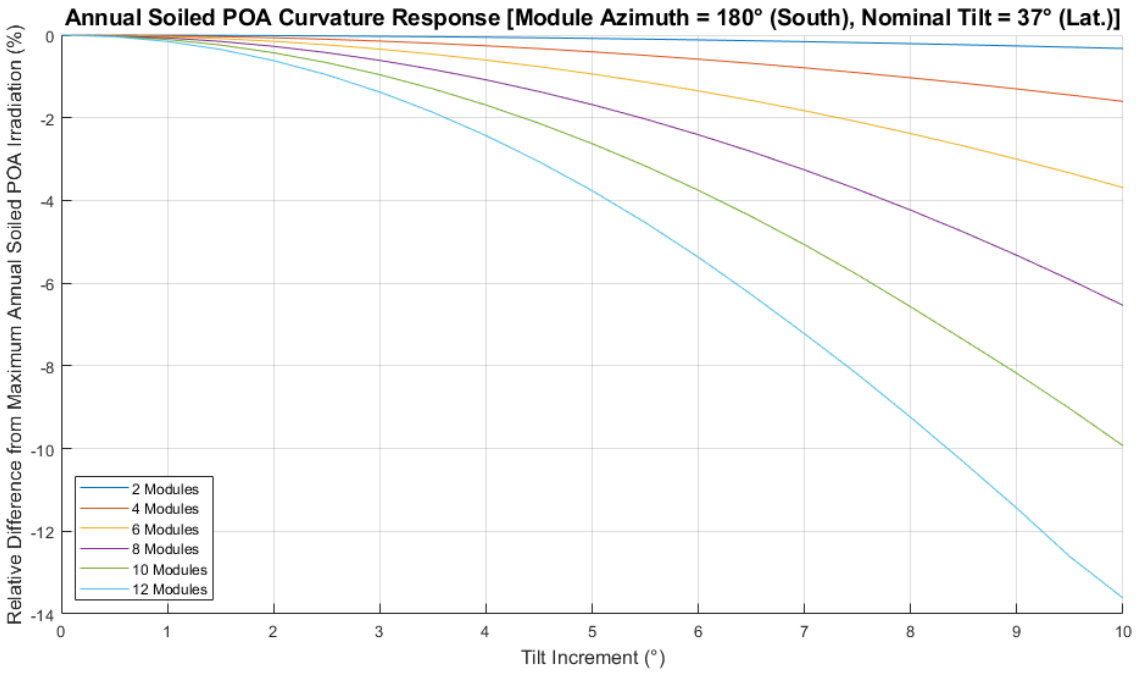


Figure 30—Soiled POA irradiation response to tilted array curvature and size.

The results indicate that the linear application of curvature to a planar array causes an exponential reduction of annual soiled POA irradiation. This was expected, as curvature causes the modules to deviate from their optimal orientation. It is also evident that a linear increase of the number of modules causes an exponential reduction of annual soiled POA irradiation. This is because each additional module has a tilt that is one tilt increment more or less than the tilt of the adjacent module. Thus, an array with many modules has extreme tilts at each end—orientations that are far from optimal.

A tilt increment of 4° reduces the annual soiled POA irradiation by a maximum of 2.5% for an array of 12 modules. It was thus selected as an acceptable compromise between aesthetics and system performance for arrays of most sizes. The 4° tilt increment is implemented as the default value for the tilted curved array definition in the canopy simulation and optimization algorithm.

2.1.5.3 Horizontal Planar Array Study

The horizontal planar array is an A-Frame design with sufficient tilt to mitigate soiling losses. The study that was conducted for the tilted curved array was repeated with a nominal tilt of 0° and an array size of two modules. The azimuth angles were varied from 90° to 270° at 30° increments and the results are shown in Figure 31.

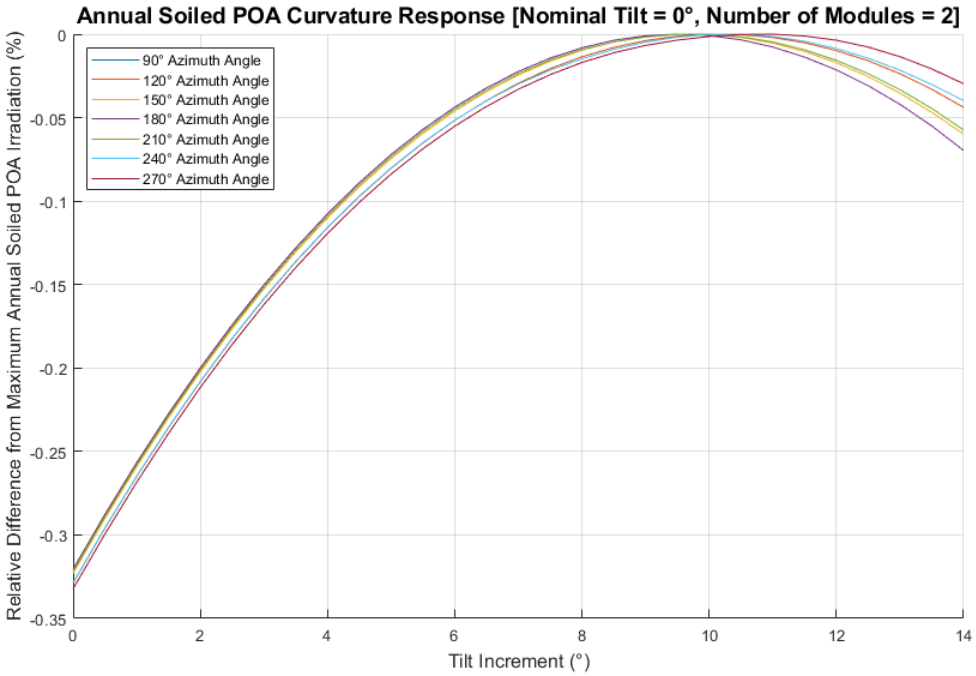


Figure 31— Soiled POA irradiation response to module tilt and azimuth angle.

The results reveal that the azimuth angle has a negligible effect on the annual soiled POA irradiation. The collected irradiation is maximized at a tilt increment of 10°, which is implemented as the default value for the horizontal planar array definition in the canopy simulation and optimization algorithm. This value balances the losses associated with soiling and orientation, as larger tilts reduce soiling loss through improved rainwater flow but also deviate from the optimal orientation for irradiance collection. Figure 32 shows an example of the horizontal planar array profile with six modules and the 10° tilt increment.

Figure 32—Horizontal planar array profile.

2.1.5.4 Horizontal Curved Array Study

The horizontal curved array is simply the tilted curved array with a nominal tilt of 0°. The same tilt increment study was conducted, this time with a module azimuth angle of 180°. The array size was varied from 2 to 12 modules at 2-module increments and the results are shown in Figure 33.

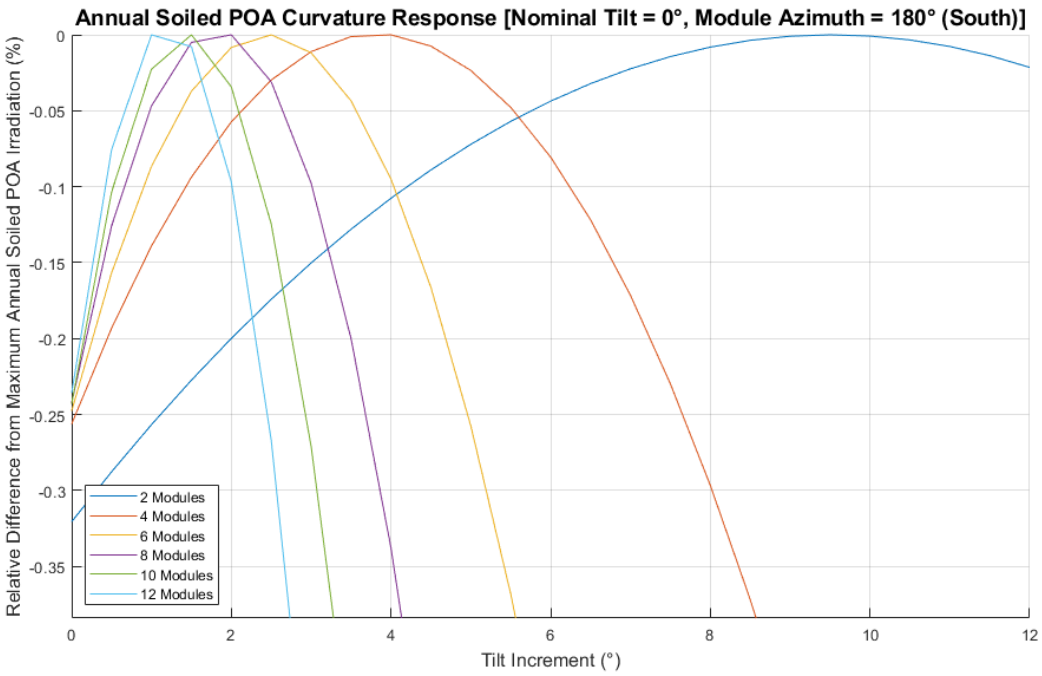


Figure 33—Soiled POA irradiation response to horizontal array curvature and size.

The results indicate that the maximum annual soiled POA irradiation is collected at a tilt increment associated with the number of modules in the array. As with the horizontal planar array study, these tilt increments balance the losses associated with soiling and orientation. The optimal tilt increments were plotted against the array size and a power function curve fit was applied, shown in Figure 34. The coefficient of determination was 0.99, indicating an acceptable fit.

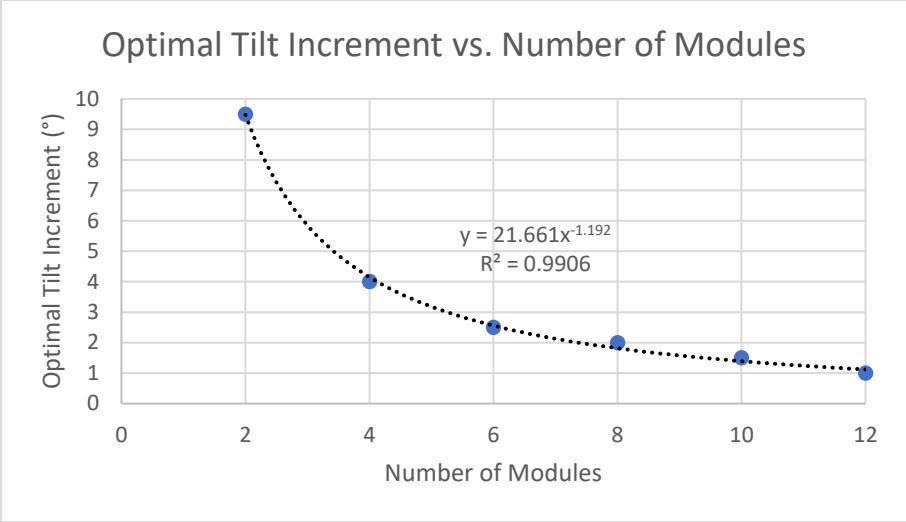


Figure 34—Curve fit correlation between array size and optimal tilt increment.

The tilt increment is rounded to the nearest degree to remove unnecessary resolution (i.e., it would be uneconomical to manufacture the canopies with sub-degree angular tolerances).

$$\Delta\beta = \text{round}(21.661\ N_{mod}^{-1.192}) \tag{50}$$

Equation (49) and Equation (50) were implemented in the canopy simulation and optimization algorithm to determine the tilt of each module in the horizontal curved array profile. A six-module example of this profile is depicted in Figure 35.

Figure 35—Horizontal curved array profile.

2.1.5.5 Dual Array Profile Study

The purpose of the dual array profile study is to quantify the gain in irradiance collection associated with the utilization of two different array profiles throughout the route. The appropriate profile is selected based on the bearing of the canopy. Tilted planar arrays are best suited for east-west canopies, while horizontal planar arrays are more productive on north-south canopies. The azimuth critical inset angle ($A_{inset,cr}$) is a symmetric angle from the east and west direction lines that increases positively toward the south direction line. It determines the array profile of the canopy, based on the location of the module normal ($\mathbf{n}_M$) within the regions marked in Figure 36. As an example, the module normal of the canopy in Figure 36 is beyond the azimuth critical inset angle, indicating the array profile is tilted.

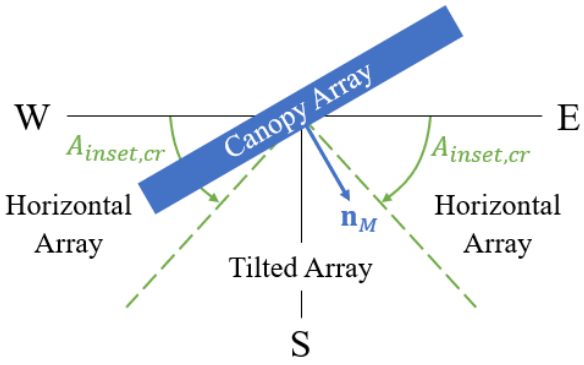
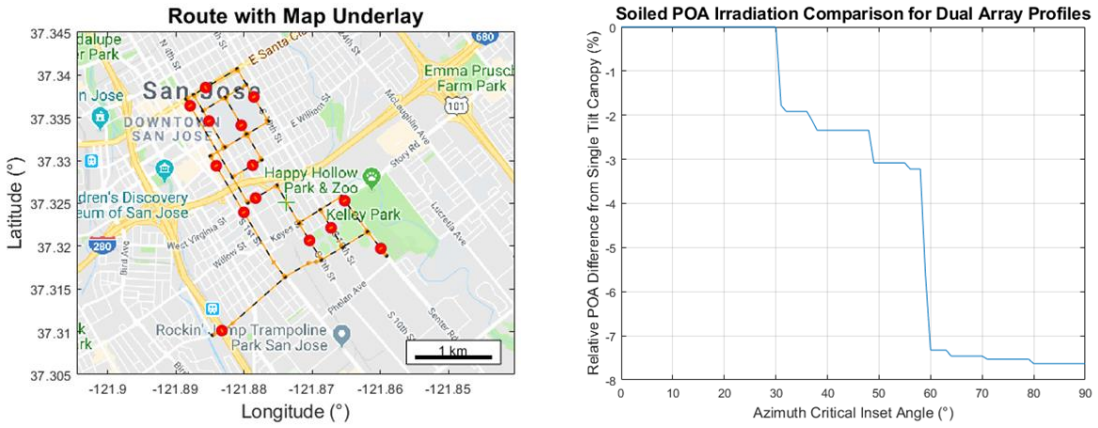


Figure 36—Depiction of azimuth critical inset angle ($A_{inset,cr}$).

The simulation study was configured to vary the azimuth critical inset angle from 0° to 90° in 1° increments for four hypothetical transit routes. For each increment, the annual soiled POA irradiation was computed with the calcPOA() function. Each soiled POA irradiation value was then normalized as the relative difference from the value

generated for a single tilted array profile. This relative difference in soiled POA irradiation represents the gain or loss associated with using a dual array profile configuration in place of a single tilted array profile.

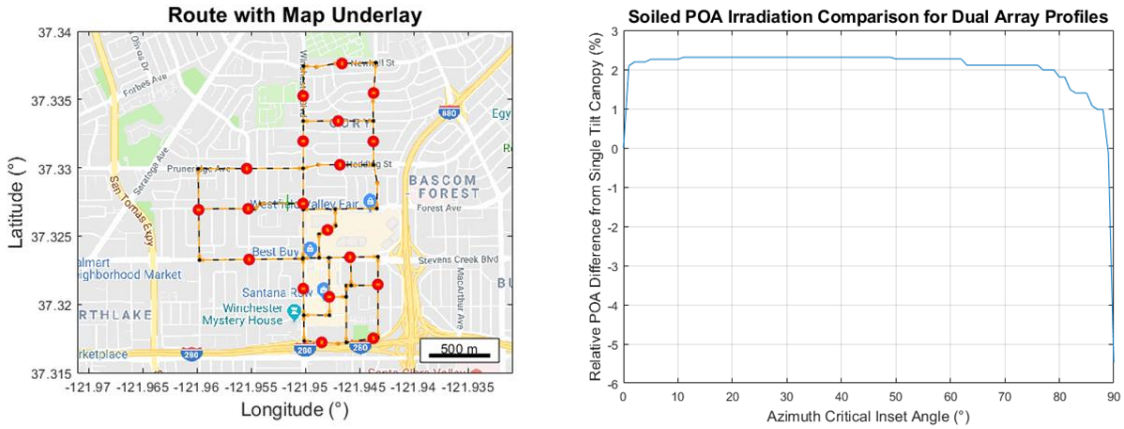
The first two routes were created to investigate the effect of city grid orientation on the irradiation collected by the dual array profile configuration. City grids are typically oriented in one of two configurations: Spanish or Jeffersonian [80]. The Spanish grid has roads oriented along the northwest-southeast and northeast-southwest direction lines, while the roads in the Jeffersonian grid are aligned with the cardinal directions [80]. The first route was designed for a region of San Jose, CA that utilizes the Spanish grid. Figure 37 shows the plot of the route (*a*) and the results from the simulation (*b*).



(*a*) (*b*)

Figure 37—Plot (*a*) and simulation results (*b*) of the Spanish San Jose route.

Figure 38 shows the plot (*a*) and results (*b*) for the second route, which was designed for a region of San Jose with a Jeffersonian grid.

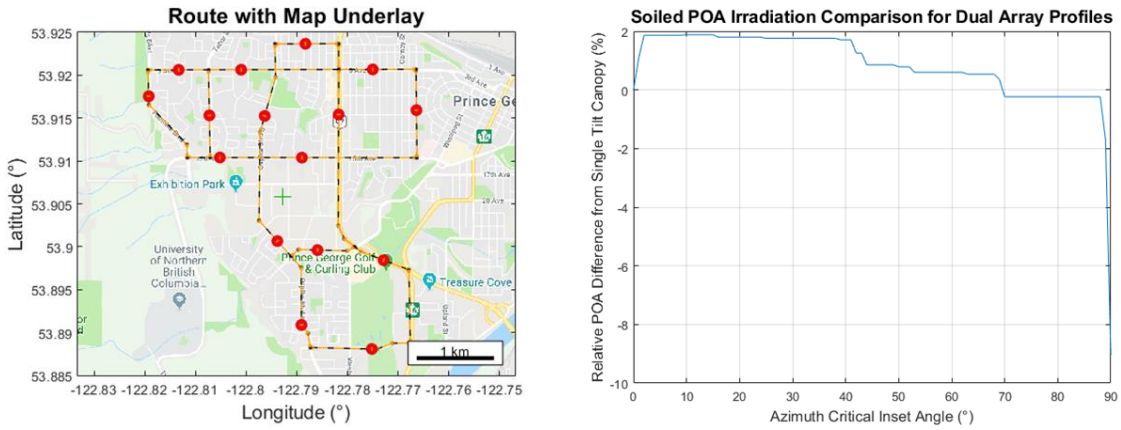


(*a*) (*b*)

Figure 38—Plot (*a*) and simulation results (*b*) of the Jeffersonian San Jose route.

The results from the Spanish route reveal that the dual array profile configuration does not collect more irradiation than the single tilted array configuration. This is attributed to the lack of north-south canopies on the route, which would benefit from the dual array profile configuration the most. The results from the Jeffersonian route indicate that the dual array profile configuration is beneficial for azimuth critical inset angles less than 77°, with a maximum gain in collected POA irradiation of 2.3% over the single tilted array configuration. This is attributed to the large quantity of north-south canopies on the route. Thus, city grid orientation significantly influences the performance of dual array profile configurations, with Jeffersonian grids achieving the largest gains in collected irradiation.

Two more routes were created to investigate the effects of location on the collected irradiation of the dual array profile configuration. Both routes were plotted on Jeffersonian grids to maximize the performance of the dual array profiles. The high-latitude route was designed for Prince George, Canada and Figure 39 shows the plot (*a*) and simulation results (*b*). The central route latitude was 53.9°.

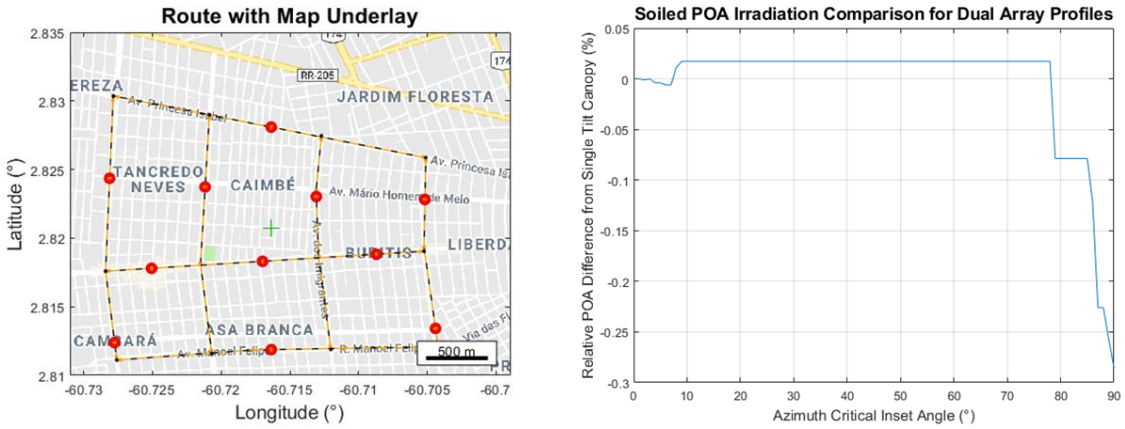


(*a*) (*b*)

Figure 39—Plot (*a*) and simulation results (*b*) of the high-latitude Prince George route.

The high-latitude route was observed to experience a maximum gain in collected irradiation of 1.9%. It was expected that the high-latitude route would benefit the greatest from a dual array profile configuration because the tilted arrays have large tilt angles to account for the small sun elevation angles. The large angular difference between the steep tilted arrays of the east-west canopies and horizontal arrays of the north-south canopies provides the greatest opportunity for irradiance gains from the implementation of the dual array profile configuration. However, the reduction in relative irradiation collection from the Jeffersonian San Jose route indicates that the performance of the dual array configuration is largely dependent on weather. Higher latitudes typically have larger amounts of diffuse irradiance, which has little dependence on module orientation when collected. Thus, selectively using different array profiles has less benefit in these conditions. Analysis of the weather data confirmed these conditions, revealing the ratio of DHI to DNI of the high-latitude Prince George route to be 2.7 times that of the San Jose route.

The final route was designed for the Jeffersonian low-latitude city of Boa Vista, Brazil. The mean route latitude was 2.8°, and Figure 40 shows the plot (*a*) and simulation results (*b*).



(*a*) (*b*)

Figure 40—Plot (*a*) and simulation results (*b*) of the low-latitude Boa Vista route.

The low-latitude results indicate a maximum gain in collected irradiance of 0.02%. This was expected because the optimal array configuration for low-latitudes is typically horizontal, yielding negligible gains if a dual array profile configuration is implemented.

In summary, the use of the dual array profile configuration is most beneficial for Jeffersonian routes in locations with low amounts of diffuse irradiance. The maximum gain in collected irradiation over that of a single tilted array profile was 2.3% for the Jeffersonian San Jose route. The mean of the maximum irradiation gains for the four routes was 1.05%. While the dual array profile configuration does improve the amount of collected irradiation, the benefit was determined to be too small to justify the additional engineering and manufacturing expenses. For this reason, the implementation

of the dual array profile configuration is omitted from the canopy simulation and optimization algorithm.

2.1.6 Module Power Output Model

The purpose of the module model is to calculate the DC power output of a single user-specified PV module, based on the soiled POA irradiance, angle of incidence, and module characteristics. The moduleModel() function is featured in Appendix J and is executed in the following four steps:

1. Irradiance Transmittance Correction
2. Irradiance Air Mass Correction
3. Cell Temperature Modeling
4. Computation of DC Power Output

2.1.6.1 Irradiance Transmittance Correction

The soiled POA irradiance on the module must be corrected to account for reflectance and absorptance losses caused by the interaction with the glass cover. Incident irradiance reflects off the cover in function of the angle of incidence, resulting in reflectance losses. A study by King, Boyson, and Kratochvil determined that these losses reduced the POA irradiance by 5% on a vertical, south-facing PV module [81]. Though reflectance losses are small, they can be easily determined and should be incorporated to improve model accuracy. The angle of refraction (γ') is first determined with Snell's law, in function of the angle of incidence (γ). The first medium is air with an index of

refraction (n_1) of unity. The second medium is glass with an index of refraction (n_2) of 1.526 [66].

$$\gamma' = \sin^{-1}\left(\frac{n_1}{n_2} sin\,\gamma\right) \;\rightarrow\; \gamma' = \sin^{-1}\left(\frac{1}{1.526}\sin\gamma\right) \tag{51}$$

The transmittance-absorptance product ($\tau\alpha$) is the proportion of soiled POA irradiance that is transmitted through the glass cover to the surface of the cell. It is calculated with the equation developed by Duffie and Beckman in [66], presented as Equation (52). This is a combination of Fresnel's and Bouguer's laws to incorporate the losses associated with the reflectance and absorptance of the glass cover, respectively. Typical extinction coefficient (K) and thickness (t) values for PV module glass covers are 4 m^{-1} and 2 mm [66].

$$\tau\alpha = e^{-\left(\frac{Kt}{\cos\gamma'}\right)}\left[1 - \frac{1}{2}\left(\frac{\sin^2(\gamma'-\gamma)}{\sin^2(\gamma'+\gamma)} + \frac{\tan^2(\gamma'-\gamma)}{\tan^2(\gamma'+\gamma)}\right)\right] \tag{52}$$

The transmittance-absorptance product is calculated for the four components of irradiance required to compute the incidence-angle correction factor. These components are surface normal ($\tau\alpha_\perp$), direct ($\tau\alpha_{dir}$), diffuse ($\tau\alpha_{dif}$), and ground-reflected ($\tau\alpha_{ref}$), which are differentiated by their associated angles of incidence. The angle of incidence for the surface normal component ($\gamma_\perp$) is, by definition, 0°. However, to avoid divide-by-zero errors, it is acceptable to use a small angle of 1° [41]. The angles of incidence for diffuse (γ_{dif}) and ground-reflected (γ_{ref}) irradiance are determined from the curve fit relationships in [82]. These relationships are functions of the module tilt, presented as Equation (53) and Equation (54).

$$\gamma_{dif} = 59.7 - 0.1388\beta + 0.001497\beta^2 \quad (53)$$

$$\gamma_{ref} = 90 - 0.5788\beta + 0.002693\beta^2 \quad (54)$$

The transmittance-absorptance product is computed for each of the four components using the associated angle of incidence. The incidence-angle correction factors ($k_{\tau\alpha}$) are then determined for each component of irradiance, as the ratio of the associated transmittance-absorptance product to that of the surface normal [66].

$$k_{\tau\alpha,dir} = \frac{\tau\alpha_{dir}}{\tau\alpha_{\perp}}, \qquad k_{\tau\alpha,dif} = \frac{\tau\alpha_{dif}}{\tau\alpha_{\perp}}, \qquad k_{\tau\alpha,ref} = \frac{\tau\alpha_{ref}}{\tau\alpha_{\perp}} \quad (55)$$

The irradiance transmitted through the glass cover ($G_{\tau\alpha}$) is the sum of the product of each component of soiled POA irradiance and the associated incidence-angle correction factor [66].

$$G_{\tau\alpha} = k_{\tau\alpha,dir}G_{dir,A} + k_{\tau\alpha,dif}G_{dif,A} + k_{\tau\alpha,ref}G_{ref,A} \quad (56)$$

2.1.6.2 Irradiance Air Mass Correction

The spectral distribution is the irradiance associated with each wavelength of the incoming photons. It has been observed to vary in function of the air mass ratio, which often alters the efficiency of receptive PV cells [83]. The cumulative effect of these spectral fluctuations on annual electricity generation is generally less than 3% [81]. Equation (57) is an empirical relationship developed by King et al. to calculate the air mass correction factor (k_{AM}) in function of the absolute air mass ratio (AM_{abs}) [84], [85]. The absolute air mass ratio is different from the air mass ratio in Equation (26)

because it incorporates the elevation of the modules (h) [83]. The zenith angle (Z) is represented in degrees.

$$k_{AM} = 0.918093 + 0.086257AM_{abs} - 0.024459AM_{abs}^2 + 0.002816AM_{abs}^3 - 0.000126AM_{abs}^4 \quad (57)$$

$$AM_{abs} = e^{-0.0001184h}\left[\cos\left(\frac{\pi}{180°}Z\right) + 0.5057(96.08° - Z)^{-1.634}\right]^{-1} \quad (58)$$

The irradiance incident on the cells (G_{cell}) is the product of the air mass correction factor and the irradiance transmitted through the glass cover.

$$G_{cell} = k_{AM}G_{\tau\alpha} \quad (59)$$

2.1.6.3 Cell Temperature Modeling

Since PV conversion is dependent on temperature, the temperature of the cells must be modeled. The nominal operating cell temperature (NOCT) method from [66] is implemented, which requires module test data, such as that of the California Energy Commission (CEC) module data base [86]. This data base features standard test condition (STC) reference values for 21,187 modules [86]. The reference efficiency (η_{ref}) is determined first, using the module area (A_m), STC irradiance (G_{STC}), and reference maximum power point current $(I_{mp,ref})$ and voltage $(V_{mp,ref})$ [41]. STC irradiance is 1000 W/m^2 [66].

$$\eta_{ref} = \frac{I_{mp,ref}V_{mp,ref}}{G_{STC}A_m} = \frac{I_{mp,ref}V_{mp,ref}}{\left(1000\frac{W}{m^2}\right)A_m} \quad (60)$$

The transmittance-absorptance product for the module ($\tau\alpha_m$) is computed as 90% of the irradiance transmitted through the glass cover divided by the soiled POA irradiance, according to [41].

$$\tau\alpha_m = 0.9\frac{G_{\tau\alpha}}{G_A} \tag{61}$$

The wind speed (v_w) is adjusted by multiplying by 0.61, as recommended in [41] for arrays that are two-or-more stories tall.

$$v_{w,adj} = 0.61 v_w \tag{62}$$

The cell temperature (T_{cell}) is then calculated according to the method in [66], in function of the ambient temperature (T_a), soiled POA irradiance, NOCT temperature (T_{NOCT}), module reference efficiency, module transmittance-absorptance product, and adjusted wind speed.

$$T_{cell} = T_a + \frac{G_A}{800\frac{W}{m^2}}(T_{NOCT} - 20\,°C)\left(1 - \frac{\eta_{ref}}{\tau\alpha_m}\right)\left(\frac{9.5\frac{°C\,m}{s}}{5.7\frac{m}{s} + 3.8 v_{w,adj}}\right) \tag{63}$$

The cell temperature is then converted to the absolute value by adding 273.15 K.

$$T_{cell,abs} = T_{cell} + 273.15\,K \tag{64}$$

2.1.6.4 Computation of DC Power Output

To compute the DC power output of the module, the current and voltage must be determined. This is done with the equivalent circuit shown in Figure 41, from [66].

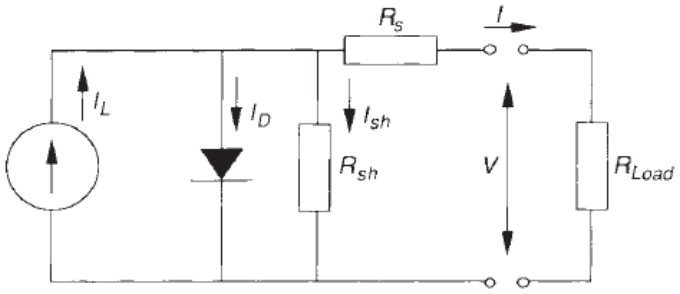


Figure 41—Equivalent circuit for PV conversion. The light current (I_L), diode current (I_D), shunt resistance (R_{sh}), shunt current (I_{sh}), series resistance (R_s), load resistance (R_{Load}), output current (I), and output voltage (V) are shown. Source: Duffie and Beckman, 2013 [66]

The current-voltage relationship is derived with Kirchhoff's current law and is comprised of five parameters: light current (I_L), reverse saturation current (I_o), series resistance (R_s), shunt resistance (R_{sh}), and ideality factor (a) [66].

$$I = I_L - I_D - I_{sh} \quad \rightarrow \quad I = I_L - I_o\left(e^{\frac{V+IR_s}{a}} - 1\right) - \frac{V + IR_s}{R_{sh}} \tag{65}$$

Each parameter is a characteristic of the module and can be determined by adjusting measured reference values in accordance with the modeled cell temperature and incident irradiance. STC reference values are loaded from the CEC module data base for these computations [86].

The light current is determined by summing the reference value $\left(I_{L,ref}\right)$ with the associated temperature correction factor and scaling according to the cell irradiance [66]. The temperature correction factor is comprised of the reference short-circuit current

temperature coefficient $(\alpha_{sc,ref})$, CEC adjustment factor $(adjust)$, and absolute cell temperature [41].

$$I_L = \frac{G_{cell}}{1000\,\frac{W}{m^2}}\left[I_{L,ref} + \alpha_{sc,ref}\left(1 - \frac{adjust}{100}\right)\left(T_{cell,abs} - 298.15\ K\right)\right] \tag{66}$$

The reverse saturation current is calculated for silicon cells according to [66]. The bandgap energy (E_{bg}) must first be determined according to the cell temperature and reference bandgap energy of 1.12 eV [66].

$$E_{bg} = (1.12\ eV)\left[1 - 0.0002677\left(T_{cell,abs} - 298.15\ K\right)\right] \tag{67}$$

The reverse saturation current is then computed in function of the reference reverse saturation current $(I_{o,ref})$, absolute cell temperature, bandgap energy, and Boltzmann constant $\left(k = 8.618 \times 10^{-5}\,\frac{eV}{K}\right)$ [66].

$$I_o = I_{o,ref}\left(\frac{T_{cell,abs}}{298.15\ K}\right)^3 e^{\left[\frac{1}{k}\left(\frac{1.12\ eV}{298.15\ K} - \frac{E_{bg}}{T_{cell,abs}}\right)\right]} \tag{68}$$

The ideality factor is determined from the reference value (a_{ref}) and absolute cell temperature [66].

$$a = a_{ref}\,\frac{T_{cell,abs}}{298.15\ K} \tag{69}$$

The shunt resistance is calculated from the reference value $(R_{sh,ref})$ and cell irradiance [66].

$$R_{sh} = R_{sh,ref} \frac{1000 \frac{W}{m^2}}{G_{cell}} \tag{70}$$

The series resistance is set to the reference value $(R_{s,ref})$ because it is generally not affected by the cell temperature or irradiance [66].

$$R_s = R_{s,ref} \tag{71}$$

With the five parameters determined for the modeled cell temperature and irradiance, the open-circuit condition of $I = 0$ and $V = V_{oc}$ is applied to Equation (65). This yields Equation (72), which is implicitly solved for the open-circuit voltage using the STC reference value as an initial guess.

$$0 = I_L - I_o \left(e^{\frac{V_{oc}}{a}} - 1 \right) - \frac{V_{oc}}{R_{sh}} \tag{72}$$

Finally, the module DC power curve is generated by sweeping through voltage values and implicitly solving Equation (65) for the associated current. The upper voltage limit is the open-circuit voltage. For each voltage step, the current and voltage are multiplied together to generate the DC power curve. Since essentially all PV systems utilize a maximum power point tracker, the maximum value of the power curve is returned as the module DC power output [12], [41].

2.1.7 PV Canopy Simulation and Optimization

The purpose of the PV canopy simulation and optimization algorithm is to generate the most efficient leg and station array profiles that will satisfy the energy

demand. This is conducted iteratively through a series of simulations that utilize the calcPOA() and moduleModel() functions. The optimized array geometries and simulated hourly power output are returned. The canopySimOpt() function is presented in Appendix K and is executed in the following six steps:

1. Array Definition
2. Array Tilt Optimization
3. Computation of Subarray Tilts
4. Orientation Correction
5. Power Output Simulation
6. Iterative Computation of Outputs

2.1.7.1 Array Definition

The leg and station arrays must be fully defined for the simulation and optimization to be performed. This entails selecting a PV module, station array scale factor, and canopy profile. The PV module is selected from the CEC module data base [86] by inputting the model name as a character vector. The data base is searched and the associated module characteristics are loaded.

The station array scale factor is the ratio of the number of modules across the width of the station array to that of the leg array. For example, a leg array profile of three modules with a station array scale factor of two will result in a station array profile of six modules. This value is specified by the user to maintain the correct array proportions during the iterative sizing algorithm.

The canopy profile is the cross-sectional profile of the array. This is selected through a dialog box that prompts the user with five options. These are illustrated in Figure 42 and subsequently detailed:

1. Tilted Planar Array: The array is planar with an optimized tilt angle.
2. Tilted Curved Array: The array is curved with 4° between each module and an optimized nominal tilt. The tilt increment of 4° was determined to be an acceptable compromise between aesthetics and performance in the tilted curved array study (Section 2.1.5.2).
3. Horizontal Planar Array: The array is nominally horizontal but features a 5° tilt on each side. This tilt was the optimal value that balanced soiling losses with orientation losses in the horizontal planar array study (Section 2.1.5.3).
4. Horizontal Curved Array: The array is nominally horizontal with the optimal curvature that balances soiling losses with orientation losses. The optimal tilt increment is determined with Equation (50) from the horizontal curved array study (Section 2.1.5.4).
5. User-Defined Array: The leg and station array profiles are each defined by specifying the number of modules and associated tilt angles. This array profile is simulated as defined, without optimization or sizing iterations.

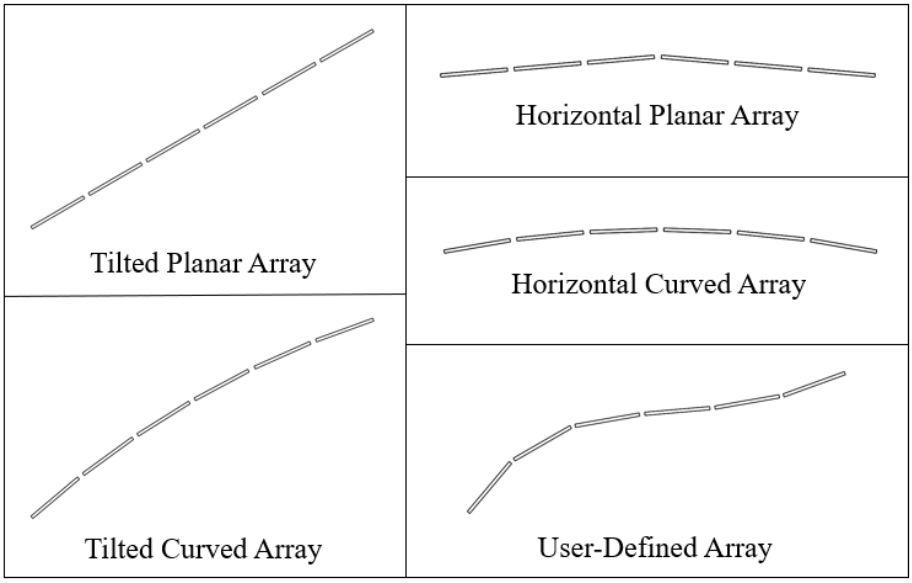


Figure 42—Illustrative summary of the five canopy profile options.

2.1.7.2 Array Tilt Optimization

The optimal tilt is the angle that causes the array to collect the maximum annual irradiation. It is dependent on the route layout, location, weather conditions, soiling losses, and shading losses. An optimization table is constructed for the computation, comprised of canopy columns and tilt angle rows. Each column represents the canopy over a leg of the route, which includes the associated station canopies. Each row represents a different tilt angle, ranging from 0° to the absolute value of the route latitude. If the latitude is within the range of ±20°, the upper tilt angle limit is set to 20°. This is a conservative value that was selected to ensure the optimal tilt angle is included in the optimization table for low-latitude routes. It was derived from the simulation results in Figure 28, which indicate that a low-latitude route can have an optimal tilt angle greater than the latitude value. The optimization table is then populated with the weighted annual soiled POA irradiation values that correspond to each canopy and tilt angle

combination. The soiled shaded and unshaded POA irradiation values are computed with the calcPOA() function. They are weighted according to the shaded and unshaded lengths of each canopy, which are determined from the leg and station shaded ratios.

Once the optimization table is populated, the weighted annual soiled POA irradiation values are summed across each row to represent the weighted irradiation collected by the entire route for each tilt angle. The maximum of these values is determined, and the associated tilt angle is returned as the optimal tilt. The tilt optimization algorithm is only performed for the tilted planar and tilted curved canopy profiles.

2.1.7.3 Computation of Subarray Tilts

Each canopy array is divided into subarrays that represent the different module orientations. Using the six-module canopy profiles illustrated in Figure 42 as an example, the tilted planar array is comprised of a single subarray, the horizontal planar array is comprised of two subarrays, and the other profiles are comprised of six subarrays. Each subarray shares the same azimuth and incline angles, which are determined for each leg and station with the loadRoute() function. The computation of subarray tilt angles is dependent on the canopy profile:

1. Tilted Planar Array: The single subarray tilt is the calculated optimal tilt.
2. Tilted Curved Array: There are the same number of subarrays as number of modules across the width of the array. Each subarray tilt is determined with Equation (49), using the specified 4° tilt increment and optimized nominal tilt.

3. Horizontal Planar Array: If there are an even number of modules across the width of the array, two subarrays of equal size are created. One has a tilt of 5° and the other has a tilt of -5°. If there are an odd number of modules across the width of the array, three subarrays are created. The central subarray has a width of one module and a tilt of 0°. The other two are of equal size, tilted at 5° and -5°.
4. Horizontal Curved Array: There are the same number of subarrays as number of modules across the width of the array. Each subarray tilt is calculated with Equation (49), using the tilt increment determined from Equation (50) and a nominal tilt of 0°.
5. User-Defined Array: There are the same number of subarrays as number of modules across the width of the array. The tilt of each is specified by the user.

2.1.7.4 Orientation Correction

At this point, the orientation of each subarray is defined by a tilt and azimuth angle, both derived from the two-dimensional representation of the route. The purpose of the orientation correction algorithm is to apply the incline angle to account for route topography. The incline angle is a counterclockwise rotation of the module normal vector about the projection of the normal vector onto the horizontal plane. The equivalent tilt and azimuth angles are returned for use in the power output simulation. Appendix L features the orientationCorrection() function, which is executed in the following five steps:

1. Convert spherical coordinate system to Cartesian coordinate system: The tilt (β) and azimuth (A) angles describe the module normal vector ($\mathbf{n}_M$), which must be

represented in Cartesian coordinates (x, y, z) for subsequent calculations. To be consistent with the sign convention of the spherical coordinate system, the longitudinal angle (ψ) is equivalent to the negative azimuth angle. The conversion is executed with Equation (73) [87], and is graphically represented in Figure 43.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos\psi \sin\beta \\ \sin\psi \sin\beta \\ \cos\beta \end{bmatrix} \tag{73}$$

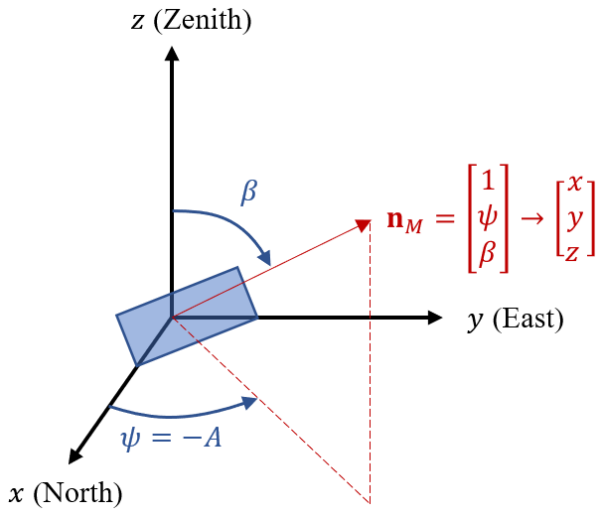


Figure 43—Spherical to Cartesian coordinate system conversion.

2. Transform Cartesian coordinate system to align with array bearing: Since the incline angle describes a rotation about the projection of the module normal onto the horizontal plane, the coordinate system must be transformed to align with this projection. A transformation angle (ψ_T) is applied, equivalent to the azimuth angle minus 90°. The resulting x'-axis aligns with the array bearing and the y'-axis aligns with the negative projection of the module normal onto the horizontal

plane. This transformation is performed with Equation (74) and Equation (75) [88], illustrated in Figure 44.

$$\psi_T = A - 90° \tag{74}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\psi_T & \sin\psi_T & 0 \\ -\sin\psi_T & \cos\psi_T & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{75}$$

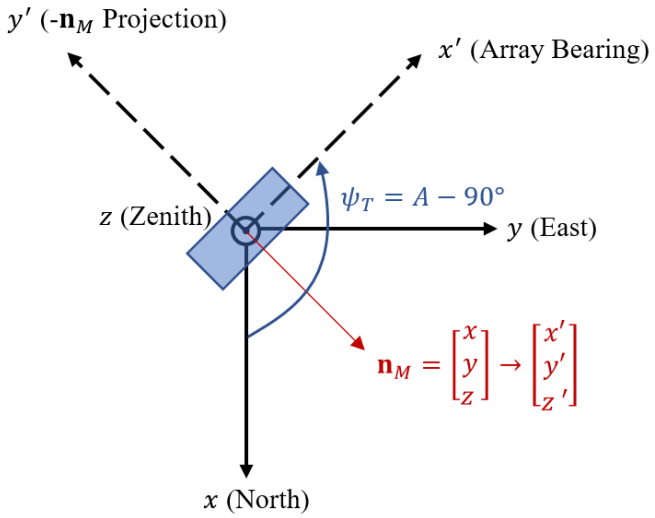


Figure 44—Coordinate system transformation to align with array bearing.

3. Rotate module normal about -y'-axis to apply incline: The incline angle (Φ) is applied as a counterclockwise rotation of the module normal about the -y'-axis. The inclined transformed Cartesian coordinates ($x'_{inc}, y'_{inc}, z'_{inc}$) are determined with Equation (76) [89], as illustrated in Figure 45.

$$\begin{bmatrix} x'_{inc} \\ y'_{inc} \\ z'_{inc} \end{bmatrix} = \begin{bmatrix} \cos\Phi & 0 & -\sin\Phi \\ 0 & 1 & 0 \\ \sin\Phi & 0 & \cos\Phi \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \tag{76}$$

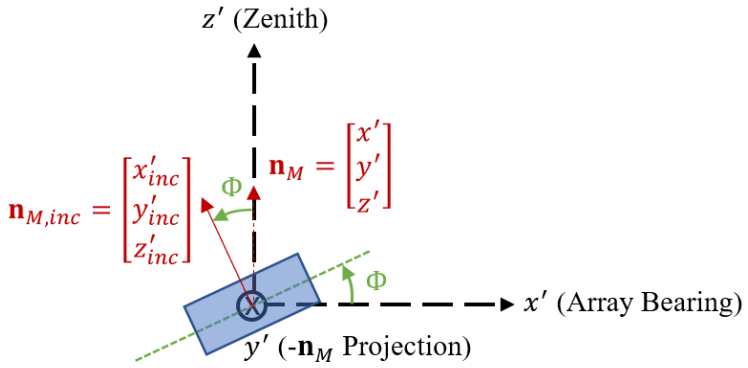


Figure 45—Incline angle (Φ) rotation of module normal vector ($\mathbf{n}_M$).

4. Transform inclined module normal vector to global Cartesian coordinates: The inclined module normal in the transformed Cartesian coordinate system must be represented in the original global Cartesian coordinate system from Step 1. This is executed in function of the transformation angle determined with Equation (74).

$$\begin{bmatrix} x_{inc} \\ y_{inc} \\ z_{inc} \end{bmatrix} = \begin{bmatrix} \cos(-\psi_T) & \sin(-\psi_T) & 0 \\ -\sin(-\psi_T) & \cos(-\psi_T) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x'_{inc} \\ y'_{inc} \\ z'_{inc} \end{bmatrix} \tag{77}$$

5. Convert Cartesian coordinate system to spherical coordinate system: The final step is to convert the inclined module normal coordinates from the Cartesian system to the spherical representation $(\beta_{inc}, \psi_{inc})$. This is conducted with Equation (78) and Equation (79), both derived from Equation (73).

$$\beta_{inc} = \cos^{-1} z_{inc} \tag{78}$$

$$\psi'_{inc} = \sin^{-1}\left(\frac{y_{inc}}{\sin \beta_{inc}}\right) \tag{79}$$

Since the range of inverse sine is [-90°, 90°], the result from Equation (79) must be adjusted to conform to the longitudinal domain of [0°, 360°]. This correction is dependent on the quadrant of the projection of the module normal onto the Cartesian xy-plane, as follows:

- Quadrant I: $\psi_{inc} = \psi'_{inc}$
- Quadrant II: $\psi_{inc} = 180° - \psi'_{inc}$
- Quadrant III: $\psi_{inc} = 180° - \psi'_{inc}$
- Quadrant IV: $\psi_{inc} = 360° + \psi'_{inc}$

Finally, the inclined azimuth angle (A_{inc}) is computed as the negative longitudinal angle. The inclined tilt and azimuth angles describe the true orientation of the subarray and are returned for use in the power output simulation.

$$A_{inc} = -\psi_{inc} \tag{80}$$

2.1.7.5 Power Output Simulation

The power output simulation computes the hourly electrical power supplied by the PV system. The first step is to calculate the number of modules along the length of each subarray. This is conducted by dividing the subarray length by the module length, orienting the long edge of each module along the direction of the route. The true tilt and azimuth angles of the subarray are computed with the orientationCorrection() function for use in the calcPOA() function, yielding the hourly soiled POA irradiance values. Next, for each hour, the number of shaded modules is determined by multiplying the number of

modules in the subarray by the shaded ratio. The unshaded DC power output of the subarray is the product of the number of unshaded modules and the single-module DC power output, returned from the moduleModel() function. The shaded DC output is computed in the same way, using the number of shaded modules and a value of zero for the DNI input to the moduleModel() function. This is because shading blocks all direct irradiance. The shaded and unshaded hourly DC power outputs are summed to yield the hourly DC power output of the subarray. The hourly outputs of each subarray are then summed and multiplied by the electrical efficiency to return the hourly power output of the canopy array. This algorithm is represented by Equation (81), where the hourly array power output $(P_{A,h})$ is determined in function of the electrical efficiency (η_{elec}), number of subarrays in the array (N_{sub}), unshaded single-module hourly DC power output $(P_{mod,un,h})$, shaded single-module hourly DC power output $(P_{mod,sh,h})$, hourly number of unshaded modules in the subarray $(N_{un,h})$, and hourly number of shaded modules in the subarray $(N_{sh,h})$.

$$P_{A,h} = \eta_{elec} \sum_{N_{sub}} \left(P_{mod,un,h} N_{un,h} + P_{mod,sh,h} N_{sh,h}\right) \tag{81}$$

The electrical efficiency term is used to incorporate the losses of the electrical components connected to the PV modules. It is comprised of four user-defined efficiencies:

1. Inverter Efficiency (η_{inv}): The average efficiency of the inverters if AC power is supplied by the system. The default value is 0.96, which is the average value listed in [12].

2. Wiring Efficiency (η_{wire}): Accounts for the transmission losses in the DC wiring. The default value is 0.98, listed in [90].
3. Connection Efficiency (η_{con}): Accounts for voltage drops across electrical connections. The default value is 0.995, listed in [90].
4. Module Mismatch Efficiency (η_{mis}): Incorporates the differences in module performance throughout the array. The default value for string inverter systems is 0.98, from [12], [90]. The use of microinverters eliminates the influence of one module output on another, yielding a mismatch efficiency of 1.00 [90].

The net electrical efficiency is the product of the four components.

$$\eta_{elec} = \eta_{inv}\eta_{wire}\eta_{con}\eta_{mis} \tag{82}$$

Finally, the hourly power outputs of the leg and station canopies are summed to yield the hourly power output of the system $(P_{sys,h})$. The number of canopies is represented as N_A.

$$P_{sys,h} = \sum_{N_A} P_{A,h} \tag{83}$$

2.1.7.6 Iterative Computation of Outputs

After the power output simulation has been completed, the following 15 outputs are displayed:

1. Rated System Generation Capacity $\left(P_{sys,rated}\right)$: The maximum DC power output of the PV system, computed as the product of the number of modules in the system $\left(N_{sys,mod}\right)$ and the rated power output of a single module $\left(P_{mod,rated}\right)$.

$$P_{sys,rated} = N_{sys,mod} P_{mod,rated} \tag{84}$$

2. Average System Power Output $\left(P_{sys,avg}\right)$: The average power output of the system, computed as the mean of the 8,760 hourly system power output values.

$$P_{sys,avg} = \frac{\sum_{i=1}^{8760} P_{sys,h,i}}{8760} \tag{85}$$

3. Annual System Energy Output $\left(E_{sys}\right)$: The total energy supplied by the PV system, calculated as the sum of the hourly energy outputs. Each hourly energy output is the product of the hourly power output and the hourly time step.

$$E_{sys} = \sum_{i=1}^{8760} E_{sys,h,i} = \sum_{i=1}^{8760} \left[\left(P_{sys,h,i}\right)(1\ hr)\right] \tag{86}$$

4. Energy Factor of Safety (FOS_E): The energy factor of safety is the simulated annual system energy output divided by the annual system energy demand (E_{dem}), from the energyDemand() function.

$$FOS_E = \frac{E_{sys}}{E_{dem}} \tag{87}$$

5. Number of Modules across Leg Canopy $\left(N_{LC,mod}\right)$: The number of modules across the width of the leg canopy profile.

6. Module Tilts on Leg Canopy $(\beta_{LC,i})$: Tilt angle of each module across the width of the leg canopy profile.
7. Total Length of Leg Canopy $(L_{LC,net})$: Net length of leg canopy to be manufactured for the route, computed as the sum of the individual leg canopy lengths $(L_{LC,i})$. The number of leg canopies is represented as N_{LC}.

$$L_{LC,net} = \sum_{i=1}^{N_{LC}} L_{LC,i} \tag{88}$$

8. Width of Leg Canopy $(L_{LC,w})$: Cross-sectional width of the leg canopy profile, determined in function of the module width $(L_{mod,w})$, module tilt angle, and number of modules across the leg canopy profile.

$$L_{LC,w} = \sum_{i=1}^{N_{LC,mod}} L_{mod,w} \cos \beta_{LC,i} \tag{89}$$

9. Number of Modules across Station Canopy $(N_{SC,mod})$: The number of modules across the width of the station canopy profile.
10. Module Tilts on Station Canopy $(\beta_{SC,i})$: Tilt angle of each module across the width of the station canopy profile.
11. Total Length of Station Canopy $(L_{SC,net})$: Net length of station canopy to be manufactured for the route, computed as the sum of the individual station canopy lengths $(L_{SC,i})$. The number of station canopies is represented as N_{SC}.

$$L_{SC,net} = \sum_{i=1}^{N_{SC}} L_{SC,i} \tag{90}$$

12. Width of Station Canopy $\left(L_{SC,w}\right)$: Cross-sectional width of the station canopy profile, determined in function of the module width, module tilt angle, and number of modules across the station canopy profile.

$$L_{SC,w} = \sum_{i=1}^{N_{SC,mod}} L_{mod,w} \cos \beta_{SC,i} \tag{91}$$

13. Total Number of Modules $\left(N_{sys,mod}\right)$: The total number of modules throughout the entire PV system.
14. Capacity Factor: The ratio of the simulated annual system energy output $\left(E_{sys}\right)$ to the rated annual system energy output (i.e., ratio of actual output to the output if all modules constantly operated at maximum rated power) [91].

$$Capacity\ Factor = \frac{E_{sys}}{\left(P_{sys,rated}\right)\left(8760 \frac{hr}{yr}\right)} \tag{92}$$

15. Estimated Cost of PV System: The estimated cost of the PV system, determined as the product of the rated system generation capacity and cost benchmark from [92]. The cost benchmark is the cost of the entire PV system per Watt of rated generation capacity. It incorporates the costs associated with the modules, racking, wiring, labor, permitting, and remaining balance of system [92]. The

2018 cost benchmarks published by NREL are presented in Table 3 and implemented in Equation (93).

$$Cost = \left(P_{sys,rated}\right)(Cost\ Benchmark) \tag{93}$$

Table 3—Residential, commercial, and utility PV cost benchmarks.

PV Sector	Rated System Generation Capacity	Cost Benchmark ($USD/$W_{DC,rated}$)
Residential	< 10 kW	$2.70
Commercial	10 kW - 2 MW	$1.83
Utility	> 2 MW	$1.06

Source: R. Fu et al., 2018 [92]

The computation of the preceding outputs concludes the first sizing iteration. At this point, the computed energy factor of safety is compared with the user-specified target value. If the computed value is greater than or equal to the target value, the simulation and optimization is complete. Otherwise, the number of modules across the width of the leg canopy is incremented. Consequently, the number of modules across the width of the station canopy is updated according to the user-specified station array scale factor. Then the subarray tilts are calculated (Section 2.1.7.3), hourly power output is simulated (Section 2.1.7.5), and system outputs are computed for the new canopies. This cycle is repeated until the simulated energy factor of safety satisfies the target value. By default, the number of modules across the leg canopy is set to two for the first iteration.

2.1.8 Algorithm Integration

The algorithms developed in the previous sections must be integrated into a single user interface. MATLAB Live Script was selected as the environment because of the

capability to embed sections of code into formatted text. This feature was utilized to list descriptions and instructions before each section of code, enabling the computational aid to be used without knowledge of programing or MATLAB syntax. The Live Script environment has additional features that include a live message display, embedded figure generation, and the capability to export results to a PDF file. The computational aid is comprised of the subsequent five operational steps, and a sample simulation report is presented in Appendix M:

1. Route Input
2. Irradiance Model
3. Energy Demand Model
4. Shading Analysis
5. Canopy Simulation and Optimization

2.1.8.1 Route Input

The route input step calls the loadRoute() function to load transit route information and represent the PV canopy. The route KML file, average corner radius, and station length must be specified by the user. The route representation data is returned in leg and station table variables that are further developed in subsequent steps. Additionally, the central coordinates of the route, length of the route, average elevation of the route, and average distance between stations are returned. Two figures are generated: Route Representation and Route with Map Underlay. The Route Representation plot illustrates the leg lengths, leg PV canopy lengths, station center points, and station PV canopy lengths on a GCS. The center of the route is marked with a green plus-sign,

while orange arrows indicate the module azimuth angle for each canopy. All legs and stations are labeled, and an example is presented in Figure 46.

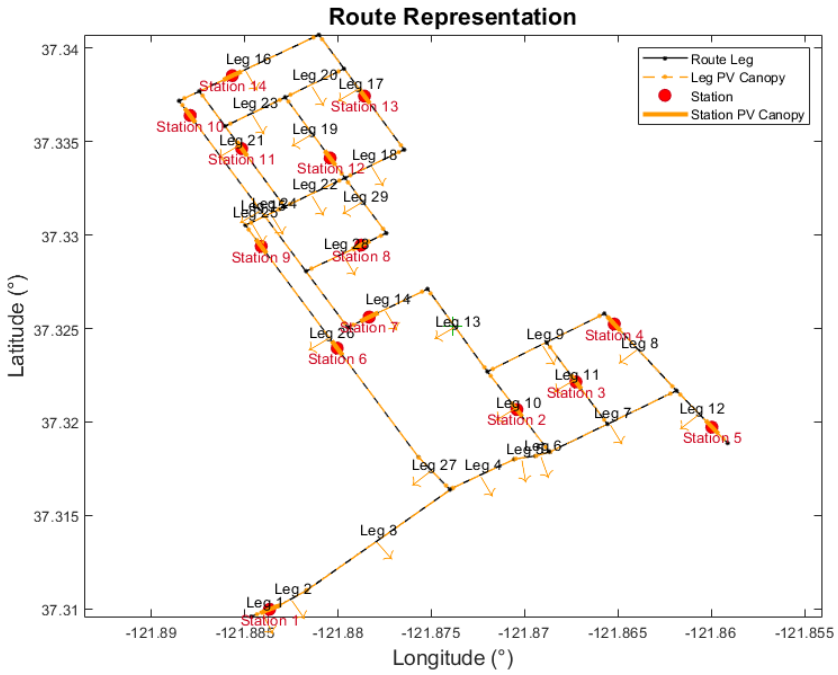


Figure 46—Example of Route Representation plot.

The Route with Map Underlay plot illustrates a context view of the route by underlaying a map from Google Maps [44], exemplified in Figure 47. This utilizes the plot_google_map() function created by Zohar Bar-Yehuda in 2018, available via the MATLAB File Exchange data base [93].

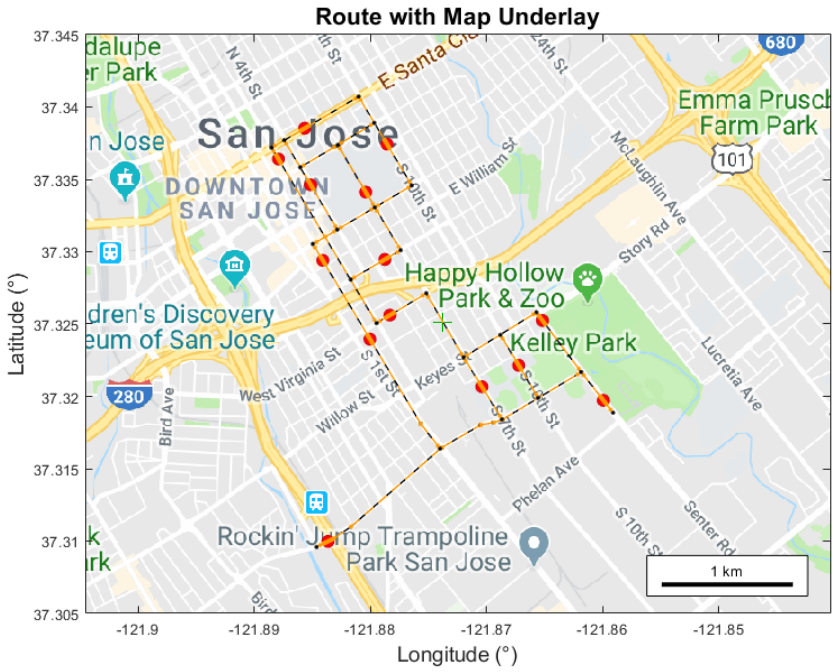


Figure 47—Example of Route with Map Underlay plot.

2.1.8.2 Irradiance Model

The second step is the generation of the irradiance model through the irradianceModel() function. The time zone of the route must be specified by the user, formatted as the number of hours ahead-of or behind Coordinated Universal Time. Additionally, a weather file may be specified that adheres to the SAM weather file format detailed in [94]. If no weather file is specified, one will be downloaded for the location through the NSRDB API [60]. The irradiance model table and mean square wind speed are returned. The Monthly DHI and DNI figure is generated, which illustrates the accumulated DHI and DNI for each month of the year. This serves as a visualization of the available solar resource, exemplified in Figure 48.

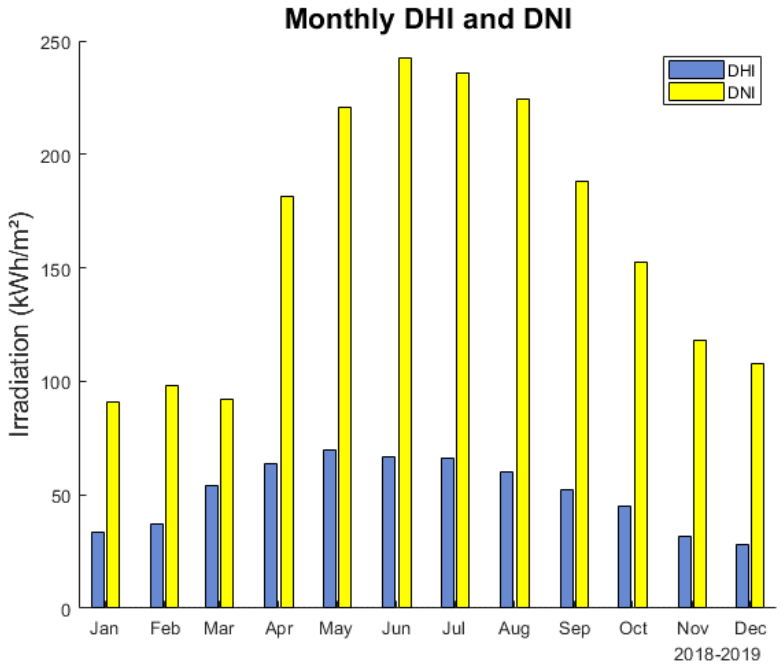


Figure 48—Example of Monthly DHI and DNI plot.

2.1.8.3 Energy Demand Model

The energy demand of the transit route is modeled with the energyDemand() function. The user must specify the vehicle mass, frontal area, wheel radius, static rolling resistance coefficient, aerodynamic drag coefficient, average motor efficiency, regenerative braking efficiency, and auxiliary power. In addition, the route line speed, headway, dwell time, and average trip elevation change must be inputted. The function returns the number of vehicles on the route, average trip duration, average system power demand, and annual system energy demand. Figure 49 shows an example of the Energy Demand Breakdown plot that is generated, which illustrates the proportion of each component of the net energy demand.

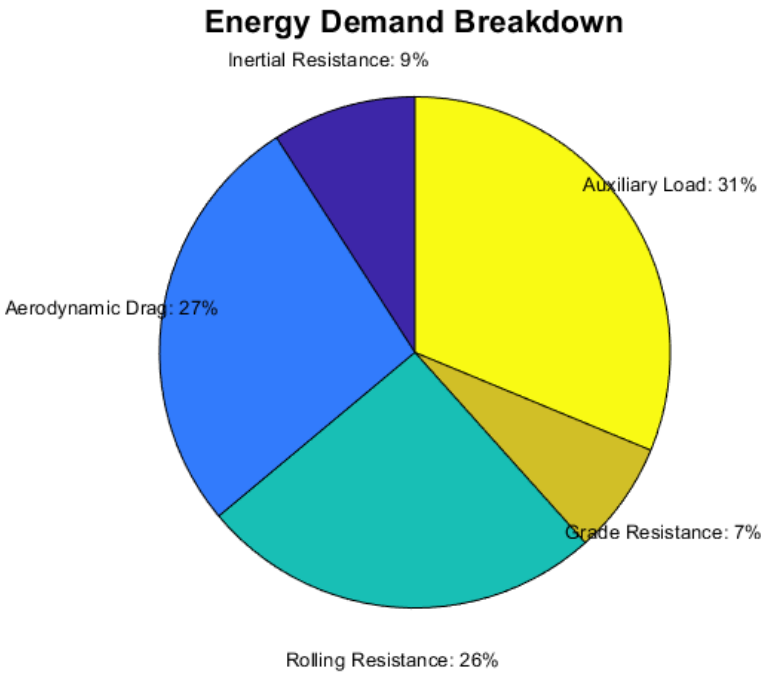


Figure 49—Example of Energy Demand Breakdown plot.

2.1.8.4 Shading Analysis

The optional shading analysis step calls the shadingAnalysis() function to determine the hourly proportion of each leg and station canopy that is shaded. A shapefile of building footprints and heights must be specified by the user, along with the height of the PV canopy. The hourly and average shaded ratios are returned for each leg and station in the associated tables. The Annual Average Shading figure is generated, which illustrates the footprints of the surrounding buildings that are taller than the PV canopy. The transit route is underlaid in black, with leg and station canopies plotted in colors that correspond to the associated annual average shaded ratios. The axes are defined by the PCS of the shapefile and an example is presented in Figure 50.

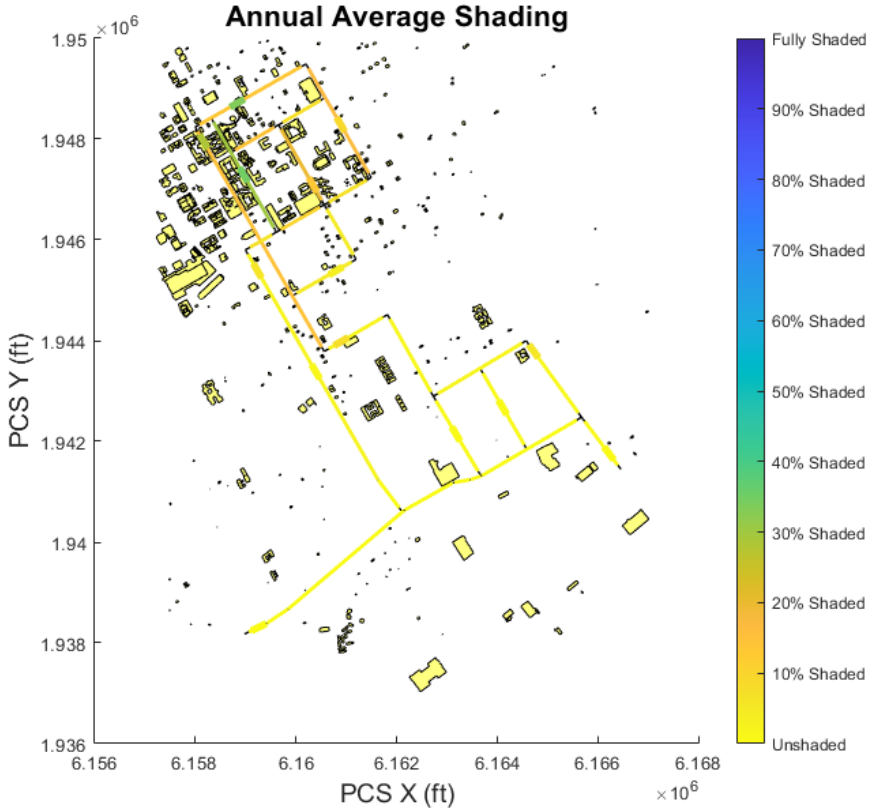


Figure 50—Example of Annual Average Shading plot.

2.1.8.5 Canopy Simulation and Optimization

The final operational step of the computational aid is the canopy simulation and optimization, performed with the canopySimOpt() function. The user must specify the station array scale factor, target energy factor of safety, and a PV module from the CEC module data base [86]. The inverter, wiring, connection, and module mismatch efficiencies are also required. The 15 outputs from Section 2.1.7.6 are printed for each sizing iteration, and the final system hourly power output results are stored in an array. Additionally, the hourly power output results for each leg and station canopy are stored in the associated tables.

Four figures are generated: System Power Output, Monthly Energy Output, Leg Array Profile, and Station Array Profile. The System Power Output plot illustrates the PV system power output at each hour of the year, as exemplified in Figure 51. The zoom feature of the MATLAB figure window can be used to inspect the power profiles of individual months and days.

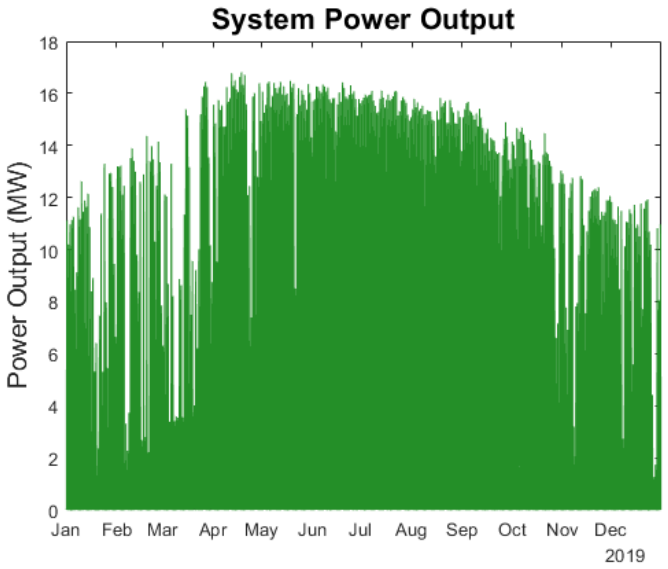


Figure 51—Example of System Power Output plot.

The Monthly Energy Output plot reveals the system energy output for each month of the year. An example is shown in Figure 52.

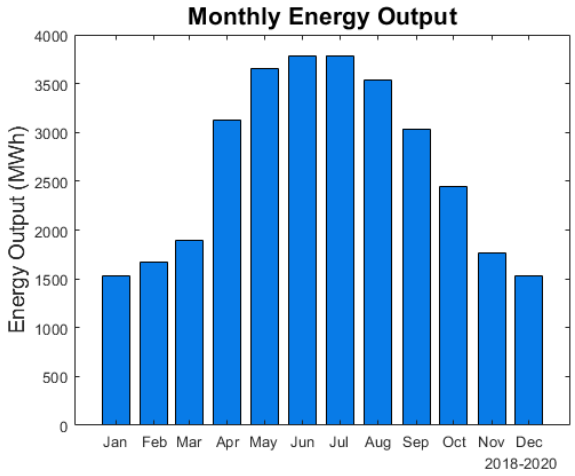
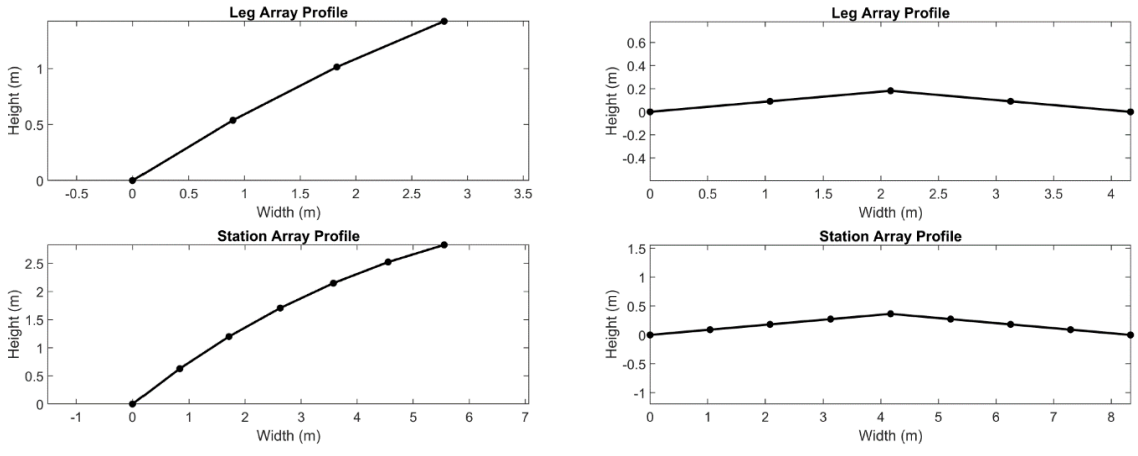


Figure 52—Example of Monthly Energy Output plot.

The Leg Array Profile and Station Array Profile plots illustrate the cross-sectional array profiles for the leg and station canopies. Each line represents a module, and the data cursor may be used to make measurements. Figure 53 shows examples of these plots for a tilted curved array profile (*a*) and a horizontal planar array profile (*b*), each with a station array scale factor of two.



(*a*) (*b*)

Figure 53—Examples of Leg Array Profile and Station Array Profile plots. (*a*) is a tilted curved array profile; (*b*) is a horizontal planar array profile.

2.2 Experimental Work

An experimental fixture was designed and fabricated to measure the effects of PV array curvature on collected irradiance. It is depicted in Figure 54 and contains three arrays, each of six cells. The first array is planar, the second has a 4° tilt increment, and the third has an 8° tilt increment. Each array shares the same nominal tilt of 37°, which is the latitude of the test location.

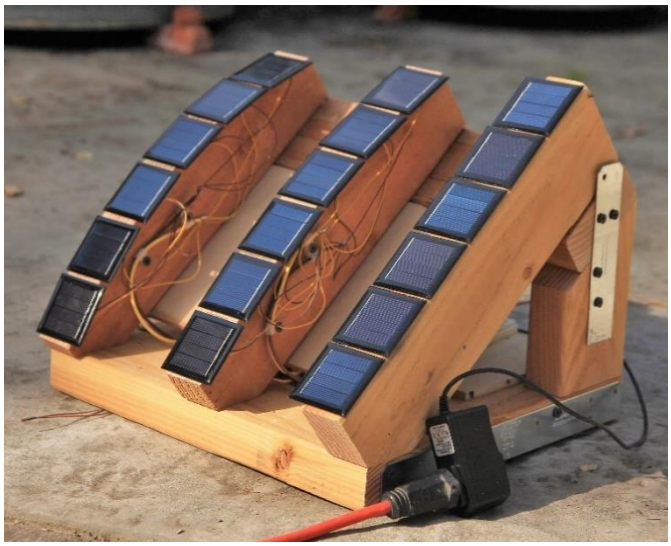


Figure 54—Apparatus for comparing irradiance collection of curved arrays.

The PV cells are manufactured by Aoshike and are specified to have a 2 V open-circuit voltage, 130 mA short-circuit current, and 29.2 cm^2 area [95]. The cells in each array are wired in parallel to prevent series mismatch losses and maintain a voltage within the range of the data acquisition system. A data acquisition system is housed within the fixture, comprised of a DS3231 real time clock, microSD card writer, 8 Ω power resistor, two relays, and an Arduino UNO microcontroller. The arrays are

connected to the power resistor through the relays so that the load can be efficiently applied to one array at a time. The microcontroller is programed to apply the load to the planar array for 1.5 seconds while sampling the voltage across the resistor. After 50 samples are acquired, the mean voltage value is squared and divided by the resistance to yield the power supplied by the array. This process is subsequently executed for the 4° and 8° curved arrays, and the cycle is repeated at the beginning of every minute. The date, time, and power from each measurement are recorded to the microSD card. Figure 55 shows the data acquisition system, and the Arduino program is listed in Appendix N.

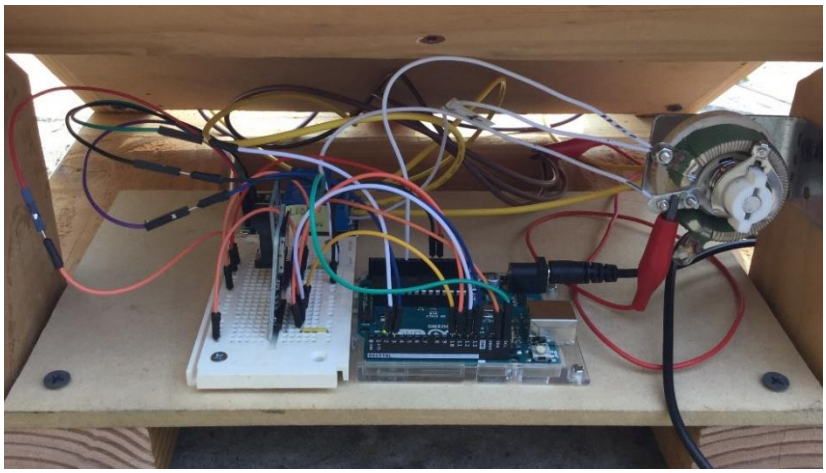


Figure 55—Arduino UNO configured to measure and record array power output.

2.3 Computer Simulation

To ensure the performance of the computational aid aligns with industry expectations, comparative simulations were conducted with two validated PV modeling programs. The first was NREL SAM, a free performance and financial analysis program for modeling PV and other renewable energy systems [22]. Freeman et al. conducted a validation study that used SAM to simulate the annual energy output of seven installed

PV systems [96]. The simulated results were within ±3% of the measured values. Another study compiled the measured annual energy outputs from 100 PV systems installed around the United States [97]. These systems were modeled with SAM and the results were compared, yielding a mean bias error of -0.8% [97]. Furthermore, since the minimum uncertainty of irradiance data is approximately 2% to 8%, the discrepancies between the SAM results and the installed-system measurements may be within the uncertainty of the input data [98]. Thus, SAM has been validated with experimental measurements and is suitable for comparison with the computational aid.

The second program was Folsom Labs HelioScope, a commercial cloud-based PV modeling software [42]. HelioScope was created to accelerate PV system design for installers, with features that include proposal generation, electrical schematic generation, and a 3D design environment [42]. In a comparison study with the measured annual energy outputs of seven installed PV systems, HelioScope had a mean bias error of -2% [99]. Another study concluded that HelioScope simulation results are within 1% of that of PVsyst, an industry-standard commercial PV modeling software [100]. HelioScope is thus a validated PV modeling program appropriate for comparison with the computational aid.

SAM, HelioScope, and the computational aid were compared through the simulation of the PV canopy on a simple transit route. The route encircles the San José State University campus and is comprised of four legs and two stations, as shown in Figure 56. It was designed to be simple to conform to the simulation capabilities of SAM and HelioScope, which require each subarray to be manually defined. The annual AC electricity output was used as the metric for comparison.

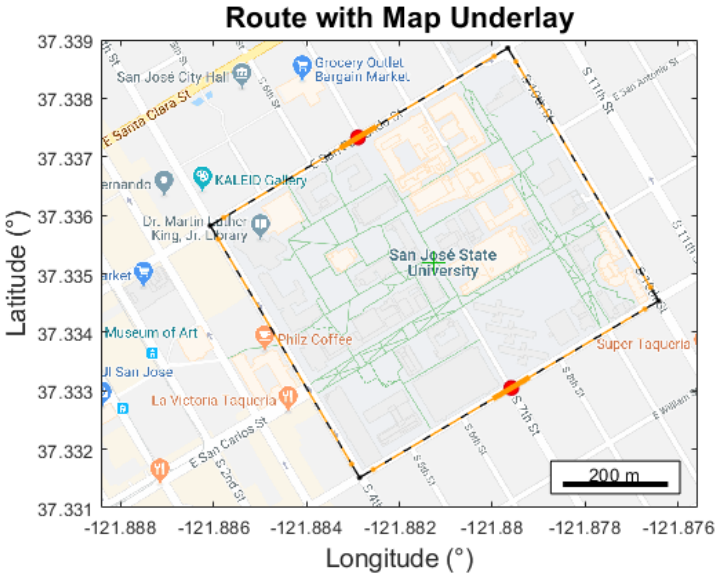


Figure 56—Simple route for performance comparison with SAM and HelioScope.

3.0 RESULTS & DISCUSSION

The analytical, experimental, and computer simulation results are presented and interpreted through the lens of the project objectives.

3.1 Analytical Results

As an extension of the 2016 case study by Branco et al. [35], the computational aid was used to model the proposed ATN route that connects the north and south San José State University campuses. This entailed modeling the energy demand of the route and sizing the PV canopy accordingly. The results are subsequently discussed, and the complete simulation report is presented in Appendix M.

3.1.1 Energy Demand Results

The annual energy demand was computed with Equation (39) in both the case study and computational aid simulation, yielding 10.4 GWh and 14.1 GWh, respectively. The large discrepancy of 3.7 GWh was attributed to five causes:

1. Inconsistent Route Lengths: The route length listed in the case study was 14.4 km, which was 3% less than the 14.9 km calculated by the computational aid. This error may have been caused by inconsistent route definitions or different length calculation methods. The longer length in the computational aid simulation caused the number of operating vehicles to be 155, instead of the 150 modeled in the case study. The resulting energy demand was approximately 0.3 GWh greater than that of the case study.

2. Erroneous Mean Square Wind Speed: The mean square wind speed was computed from the hourly weather data in the computational aid, returning a value of 14.5 m^2/s^2. A value of 9 m^2/s^2 was erroneously selected in the case study, which caused the associated energy demand to be underrepresented by 0.1 GWh.
3. Inconsistent Average Trip Distance: The case study used a value of 3,500 meters—the distance between the north and south campuses—for the average trip distance. The computational aid used a more conservative value of 1,490 meters, which was determined from Equation (15) as the average distance between two stations. This value represents the scenario of maximum energy demand, in which all vehicles stop at each station to load and unload passengers. The resulting energy demand of the simulation was 0.8 GWh larger than that of the case study.
4. Erroneous Static Rolling Resistance: The static rolling resistance was erroneously calculated in the case study with the dimensionless coefficient of rolling resistance (value of 0.0095). The static rolling resistance per unit mass should have been used instead (value of 0.093 N/kg), causing the energy demand to be underrepresented by approximately 3.5 GWh.
5. Erroneous Average Trip Elevation Change: The case study used a value of 10 meters for the average trip elevation change, which was erroneous in that it implied each vehicle ascended 10 meters during every trip without descending. Since the route is a closed loop, all elevation gained by a vehicle is lost each time it passes through the same point. Thus, the correct average trip elevation change

for the route is zero, causing the case study energy demand to be overrepresented by 1.0 GWh.

3.1.2 PV Canopy Sizing Results

The case study concluded that a PV canopy of 19,600 south-facing modules, tilted at 30°, would satisfy the energy demand. This is invalid for four reasons. The first is that the energy demand was underrepresented due to the erroneous mean square wind speed, average trip elevation change, and static rolling resistance terms. The second reason is that the orientation of each module is dependent on the bearing of the associated leg. This causes modules to have orientations that are less productive than the uniform south-facing 30° tilt specified in the case study. The third reason is that no shading analysis was conducted in the case study, causing the energy supplied by the PV system to be overrepresented by 1.6% for this building-route configuration. Finally, the case study uses an energy safety factor of unity, in which the canopy is sized to supply to exact amount of energy demanded by the transit system. This is unrealistic because the power output of the PV system is unevenly distributed throughout the year. The system generates more electricity during the summer months, which cannot be stored for use in the less-producing winter months. A safety factor of unity also requires the storage system to operate with zero losses—another impossibility.

The inaccuracies associated with the case study resulted in a largely undersized PV canopy for the transit system. The computational aid is suitable for sizing the canopy because it utilizes the correct energy demand parameters, models different module orientations, computes and applies shading losses, and incorporates a user-defined energy

safety factor. The simulation was conducted with the same SunEdison SE-R360EzC-4y modules and 96% inverter efficiency that were used in the case study. The results indicate that a planar, seven-module PV array profile at an optimal tilt of 27° will yield a conservative energy safety factor of 1.97. The associated width of the array is 6.2 meters, which is less than the width of a two-lane road with standard 3.6-meter lanes [101]. The entire system contains 47,943 modules and has a projected installed cost of $18.3 million, which is only 5% more than the $17.4 million estimated for the case study canopy. This is because the case study used an older commercial cost benchmark of $2.46/W, rather than the current utility cost benchmark of $1.06/W [92]. Thus, the simulation results reveal that it is spatially and economically feasible to conservatively power a transit system with PV modules.

3.2 Experimental Results

The results from the experimental fixture and computational aid simulation are presented individually and subsequently compared.

3.2.1 Experimental Fixture Results

The experiment was conducted in clear-sky conditions on November 4, 2018 from 10:26 AM to 5:44 PM. The fixture was positioned so that the arrays were facing south, and Figure 57 shows the resulting power output of each array at every minute of the data collection period. The 0°, 4°, and 8° arrays are represented in blue, orange, and yellow, respectively.

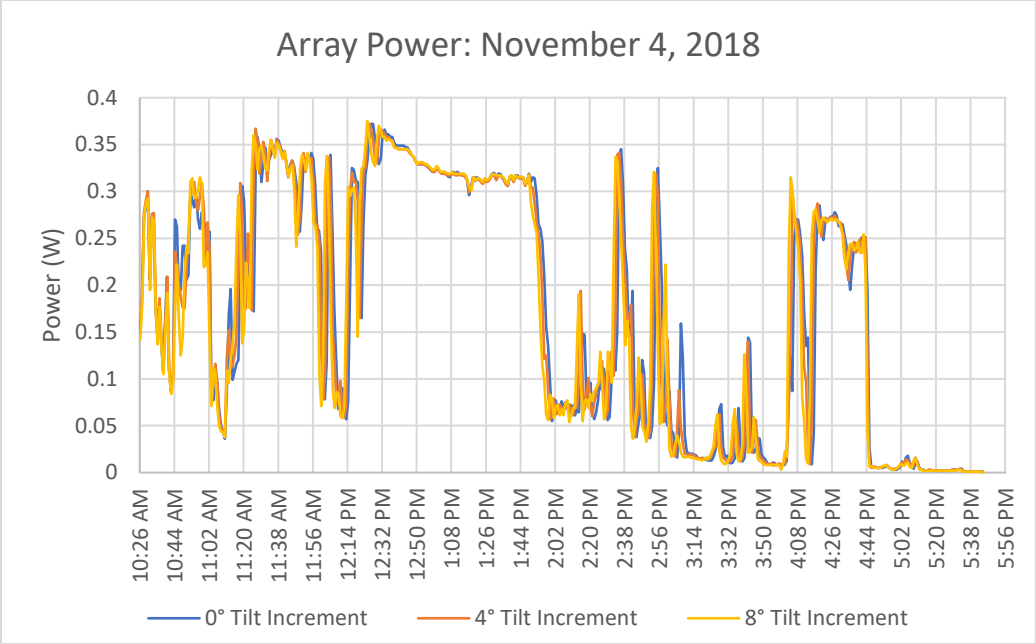


Figure 57—Power measurements from the 0°, 4°, and 8° arrays on Nov. 4, 2018.

The location had many surrounding trees that caused frequent shading throughout the test, with the exception of the period from 12:45 to 1:45 PM. This is clearly represented in Figure 57 by the steady plateau in power output. The rest of the plot features sharp transitions, which are indicative of shading because shadows rapidly block most of the incident irradiance on the cells. The results also reveal that all three arrays had nearly identical power outputs throughout the day. To determine the losses associated with the application of curvature, the total energy output of each array was calculated and compared. These values are summarized in Table 4 and indicate that the application of curvature reduces the generated electricity. This loss appears to grow exponentially as the tilt increment is increased.

Table 4—Total energy output and associated losses of the experimental arrays.

Experimental Results	0° Array	4° Array	8° Array
Measured Energy Output (Wh)	1.264	1.261	1.245
Loss (%)	-	0.2%	1.5%

3.2.2 Simulation Results

The results from the tilted curved array study (Section 2.1.5.2) were used for comparison with the experimental data. The six-module simulation in this study replicated the experimental configuration of south-facing, six-module arrays of 37° nominal tilt and 0°, 4°, and 8° tilt increments. Table 5 summarizes the simulated annual POA irradiation collected by each array, as well as the associated losses.

Table 5—Annual POA irradiation and associated losses of the simulated arrays.

Simulation Results	0° Array	4° Array	8° Array
Annual POA Irradiation (MWh/m^2)	2.054	2.042	2.006
Loss (%)	-	0.6%	2.4%

3.2.3 Comparison of Results

Similar to the experimental results, the simulation results reveal an exponential growth in collection loss as the tilt increment is increased. The experimental losses are lower than those of the simulation, which was attributed to the large amount of shading. Shading blocks DNI, causing the incident irradiance to be predominately diffuse. Since diffuse POA irradiance is much less sensitive to module orientation than direct POA irradiance, the relative loss caused by curvature is lower. Figure 58 illustrates this

through the comparison of simulated hourly POA irradiance across three consecutive days in February. The vertical axis indicates the relative difference in POA irradiance from that of the planar configuration (i.e., irradiance collection loss caused by curvature).

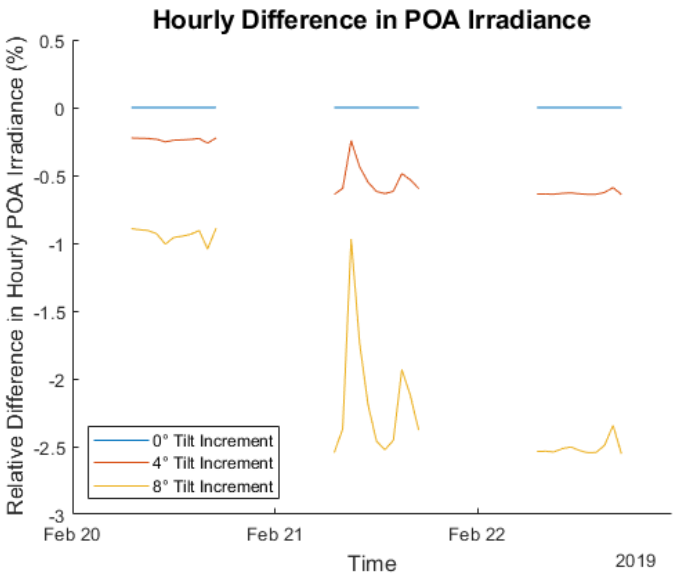


Figure 58—Simulated relative difference in hourly POA irradiance.

The first day, February 20, is overcast with no DNI. February 21 is partly cloudy with moments of low DNI, and February 22 is mostly clear with high DNI. The associated loss of collected irradiance ranges from 0.2% to 0.6% for the 4° array, and 1.0% to 2.5% for the 8° array, with the lowest losses encountered when DNI is zero. The experimental losses of 0.2% and 1.5% for the 4° and 8° arrays are consistent with these ranges, indicating acceptable agreeance between the simulation model and experimental data.

One consideration about the results from the experimental fixture is the quality of the PV cells. The cells were inexpensive and have physical dissimilarities when inspected closely. Two cells were randomly selected and characterized by measuring the