Research Performance Progress Report (RPPR): Final Report

Project Title:	SAM Enhancements for CSP
Project Period:	10/01/12 – 9/30/15
Project Period Budget:	$781,675
Submission Date:	11/19/15
Recipient:	NREL
Recipient DUNS #:	805948051
Address:	15013 Denver West Parkway, MS RSF033 Golden, CO 80401-3305
Award Number:	Neises_A
Awarding Agency:	DOE EERE SETP CSP subprogram
Working Partners:	N/A
Cost-Sharing Partners:	N/A

Principal Investigator:	Ty Neises Engineer III Phone: 303-275-4537 Fax: 303-384-7495 Email: ty.neises@nrel.gov
Submitted by:	Ty Neises

DOE Technical Manager:	Andru Prescod

Contents

Introduction

Project Objectives (from FY15 SOPO)

1) Leverage recent SAM kernel improvements by developing a robust solver specifically designed for CSP systems. This work will allow for simulations of more complex plant control algorithms and receiver, field, and storage models optimization codes while also providing a standardized interface that will streamline modeling efforts.

2) Integrate SolarPILOT into SAM as a replacement for DELSOL3. SolarPILOT maintains the capabilities of DELSOL3 and add the ability to investigate complex solar field designs, analyze a greater variety of heliostat geometries, and characterize custom field layouts. Furthermore this work enables full integration and multi-threading within SAM.

3) Develop advanced power cycle and hybrid plant models that have been proposed for CSP systems. These innovative power cycles are potentially higher efficiency and lower cost alternatives to current CSP cycles.

Introduction

Previous work led by NREL has produced a suite of hourly CSP performance and annual financial models that are currently implemented in SAM. These analysis tools are used within the DOE national laboratory system and worldwide by researchers, industry members, academics, and policymakers to determine the potential performance and financial impact of renewable energy technologies.

SAM includes models for the following CSP solar-field technologies: parabolic troughs, both molten-salt and direct-steam central receivers, dish/Stirling, and direct-steam linear Fresnel systems. SAM also includes a generic solar system model that represents the solar field using a set of optical efficiency values for different sun angles and can be used for any solar technology that uses solar energy to generate steam for electric power generation. Each technology model requires information about the receiver and collector, power block, thermal storage, and operating conditions to predict component and overall system performance. The CSP models also include an optional auxiliary fossil backup system, as appropriate. This project aims to maintain SAM's unique ability to efficiently model pertinent CSP technologies by improving the functionality of SAM and improving current or creating new technology models.

Layout

This three year project is composed of six tasks. While there is some dependence between tasks, each task's objectives, approach, and conclusions are distinct enough to motivate separate sections for each task. Our goal is that each section serves as a stand-alone reference for its task. The sections are in rough chronological order. We included less detail for the first three tasks as we completed them in the first two years, and the respective annual reports contain extensive documentation. We completed the final three tasks in FY15. Consequently, these sections contain the most information as well as documentation regarding FY15 milestones. Some work has lent itself well to journal publication, and we have elected to provide documentation in journal format for several topics to facilitate the direct submission of work from this project to high-impact,

peer-reviewed journals. We provide relevant introductory material but refer the reader at the appropriate juncture to the appendices that contain the formatted papers.

Core Programming Enhancements

Motivation

CSP system models in SAM typically consist of independent models corresponding to the system components. For example, the molten salt parabolic trough model consists of components for the solar field, power block, and controller. This framework allows multiple systems to share component models (e.g. the trough, Fresnel, and power tower all use the same power cycle model). In a system simulation, the performance of these components is interdependent – i.e. the performance of any one component is a function of the performance of the other components. In turn, the system model is highly implicit and an iterative solver is required to find the correct solution. In order to solve the system model, as of 2012 the CSP system models were coded in TRNSYS [1], a general purpose transient modeling platform written in the Fortran programming language that uses a successive substitution solver.

As SAM became generally accepted as a standard for CSP system modeling, limitations of the TRNSYS software hindered the advancement of the SAM models on several fronts. First, the TRNSYS software is not thread-safe and therefore the CSP models cannot be run effectively on highly parallelized computer architectures. In addition, as is discussed in greater detail in the Dispatch Optimization section, the iteration structure imposed by TRNSYS, while generally applicable, is rigid and does not easily allow the solution method strategies to be modified for improved convergence, optimization, or system control. Third, the need to support SAM models across multiple operating systems (Windows, Linux, Mac OSX) requires that high quality compilers exist across all platforms and hardware architectures (x86-32,x64), which in our experience is problematic with Fortran – while excellent C++ compilers exist everywhere. Fourth, using an object oriented language like C++ allows the system models to be represented in reusable classes instead of unwieldy subroutines, making it easier and less error-prone to update and modify the software. Finally, TRNSYS is proprietary software. Although SAM was licensed to use it, users could easily use SAM's component models for customized studies without purchasing TRNSYS. Crucially, this also meant that SAM was reliant on TRNSYS for upgrades and improvements to the core solver.

Approach

We addressed the limitations described above by leveraging a new in-house modeling framework (this framework, TCS, was developed by Aron Dobos in the Energy Modeling and Forecasting Group at NREL). This reentrant, thread-safe framework is programmed in C++ (a modern object oriented language for which high quality free development tools are available) and built upon a custom successive substitution solver. The solver calls a class-based system of types. This approach compartmentalizes data and recipes, which creates a more intuitive code and a thread-safe environment without global variables. (Note that a thread-safe environment is achievable in Fortran – but less conveniently so, and it is very cumbersome to modify extensive Fortran codes to be thread-safe). Additionally, the new framework eliminates the parallel-processing-

bottleneck of repeatedly opening, reading, and closing input text files by reading the files, passing the information to C++, and then treating the information as constant data. The new framework integrates scripting for convenient testing of models across a variety of operating regimes, and includes built-in visualization of timeseries data to accelerate the code-debug-fix-validate model development process. Additional details describing the framework and the process we followed to reconstruct our CSP models in it can be found in the 2013 Continuation Report.

Results

The goals of this task were to accurately reconstruct the CSP models into the TCS framework while improving computational performance, especially when parallelized. The following is a brief discussion of task outcomes regarding the milestones. Detailed discussion can be found in the 2013 fourth quarter report. We also wrote a conference paper on this topic [2].

Figure 1 shows current parallelization performance on an Intel i7 mobile processor that has four physical cores and hyperthreading. Hyperthreading is a technology that can efficiently map two parallel threads onto one processor core with little overhead, so that the processor appears to have 8 virtual cores. Up to four cores, the overhead is at or below 10%. Above that, hyperthreading allows the TCS thread-based parallelization to more effectively utilize the processor than the TRNSYS simulation in which each simulation requires a separate operating system process to execute. Although we do not have access to a computer with 32 physical cores, we suspect the overhead shown for 2, 3, and 4 cores to be reduced and that the 15% overhead target on 32 core machine would be within reach. Note that cores 5 through 8 in the above figure are not representative with respect to the 15% 32 core metric, as the processor that generated these results is utilizing hyperthreading, not physical processor cores. Still, these results are a meaningful example of the performance gain for a typical PC.

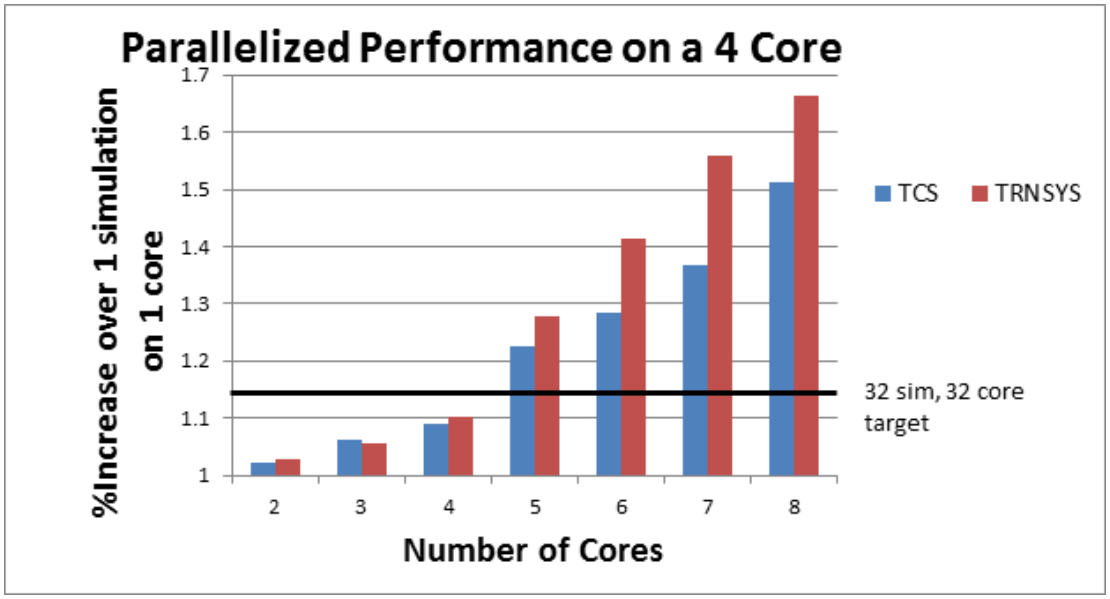


Figure 1: Parallelized performance of the TCS physical trough model on an Intel i7 mobile processor with four physical cores and hyperthreading.

Table 1 shows validation of the default case for each technology model that we converted to TCS. The results in the table are the root-mean-square deviation (RMSD) calculated from the hourly power generation results of an annual simulation and normalized by the design net power output. The results show that the validation was well within the 1% milestone target. Technology models requiring significant iteration within and between components trend towards high RMSDs. Because the RMSD considers the absolute difference each timestep, the annual cumulative energy difference balances positive and negative errors and is typically less than the RMSD.

Table 1: Validation of TCS against Fortran of the default case for each technology model. Results are the root-mean-square deviation calculated from the hourly power generation results of an annual simulation and normalized by the design net power output.

Physical Trough Model	Empirical Trough Model	Generic Solar System	Molten Salt Power Tower
0.78%	0.05%	0.02%	0.38%
Direct Steam Power Tower	**Direct Steam Linear Fresnel**	**Dish Stirling**	
0.69%	0.34%	0.05%	

It is worth reviewing the planned and recently added SAM features that this code restructuring enabled. Previously SAM called DELSOL to model the solar field performance and then passed those results via text file to the CSP system performance models. Now we have incorporated SolarPILOT and the system performance models into the same coding framework. This arrangement eliminates the major bottleneck of reading and writing text files and facilitates running batch simulations through scripting because only one main function call is required. The new framework also enables callbacks in the user interface to share information with performance models. We implemented this feature to add user interaction with solar field design and optimization and sCO2 cycle design. The ability to access thread-safe component-level performance models allows us to develop wrappers around the code that we can use for design and operation optimization. It also allows us to isolate various methods and functions to share with collaborators without overwhelming them with our entire code base and risking our IP. We will be using these capabilities on our ongoing CSP research projects. Moreover, the new framework reveals a potential pathway for users to simulate custom component models within SAM's system models by developing models that connect to SAM through API's. Finally, because this task allowed us to spend time improving existing code, we were able to find important bugs and improve its functionality and usability.

Integrated Solar Combined Cycle (ISCC) Modeling

Motivation

Solar hybridization with fossil-fuel power plants is one strategy to deploy CSP solar field technologies with a lower investment risk while maintain reliable, dispatchable generation. This configuration eliminates most of the power cycle cost associated with a solar-only plant. The thermal storage system may also be eliminated or significantly reduced because the fossil cycle provides guaranteed generation during changes in

solar irradiation. Furthermore, a solar hybrid plant may reduce the fossil-fuel consumption and emissions, even in comparison to an equivalent stand-alone solar plant. Due to these advantages, utilities have shown interest in solar integration into different types of fossil-fuel power plants [3], and a number of solar hybridization projects have been implemented or are under construction across the world.

Solar thermal power can integrate with gas turbines, coal-fired Rankine cycles, or natural gas combined cycles, largely depending on its receiver outlet temperature. We modeled an Integrated Solar Combined-Cycle (ISCC) system for this task based on the discussion in the 2013 Continuation Report. An ISCC injects solar thermal power into heat recovery steam generators (HRSG) in a Natural Gas Combine Cycle (NGCC). ISCCs allow the potential for significant solar hybridization [4], [5] and have the potential to achieve substantial economic and environmental benefits, compared with solar integration into other types of power cycles. Previous work has shown that solar-to-electric conversion efficiency may reach around 31% with a solar contribution of 10% compared to the overall plant power generation [4]. McMahan and Zervos reported that retrofitting an NGCC through replacing the duct burner capacity with solar thermal power can be economical when compared to a standalone CSP plant [6]. According to Ojo et al. [7], higher solar fractions may be achieved through new ISCC plant design than through retrofitting an existing NGCC plant. Montes et al. [8] suggested that hybridization may be more beneficial in a hot, dry climate as it provides a natural offset to decreased gas turbine performance on hot days. When retrofitting an existing plant with solar, a thorough feasibility study is needed not only to evaluate the thermodynamic impact of solar integration to the plant, but also to examine realistic physical/operational limitations in the existing plant configuration. A series of reports led by the Electric Power Research Institute (EPRI) [9], [10], [11] showed that, even though a conceptual analysis may suggest a clear benefit from solar hybridization, the physical limitations on the size of solar field and solar injection points may add practical constraints to an otherwise feasible solar integration.

Approach

First we selected the specific ISCC configuration to model. We chose to integrate the molten salt power tower as a retrofit hybridization in parallel with the high pressure boiler. One reason for this selection is that an ISCC configuration with a high pressure boiler solar injection was previously modeled by EPRI. Their oversight on this project provided confidence that our configuration was practical and our modeling results were reasonable. Retrofitting the plant allowed us to use the design of an existing NGCC rather than spend considerable time designing a greenfield ISCC plant. Finally, we chose the molten salt power tower because it is an emerging CSP technology that can achieve hot outlet temperatures that integrate with the ISCC high pressure boiler. This combination allows for the highest solar efficiencies and solar fractions, and it allowed us to collaborate with a visiting research from GE who was studying this configuration.

We modeled the NGCC configuration in the IPSEpro software package. IPSEpro is a commercial code that contains first-principles models for common components in thermodynamics systems. Its graphical interface allows the user to define the connection between components and component operating parameters. It uses a free equation solver to simulate a system given fixed inputs. It’s a powerful tool for modeling

complex thermodynamic system behavior, but it requires a high level of user expertise and generally is not flexible or computationally efficient enough for the annual hourly simulations and high level design parametric simulations for which SAM is often used.

We used IPSEpro to study how different amounts of solar thermal input to the high pressure boiler affected the cycle performance. We discuss these results at length in the 2013 Continuation Report and the paper we published on the topic [12]. One interesting contribution to the literature from our work was showing the relationship between solar conversion efficiency and NGCC efficiency as a function of the solar thermal input. At small amounts of solar thermal input, the solar conversion efficiency is significantly higher than the NGCC efficiency and the expected thermodynamic efficiency of a stand-alone CSP plant operating at those temperatures. The reason for this behavior is that at the hot ambient temperature used for the analysis, the NGCC was operating at significant off-design. This is because the solar thermal input contributes to the NGCC output by both supplying thermal power *and* boosting overall conversion efficiency of the NGCC flue gas by driving the NGCC closer to its design operating conditions.

Then we ran the IPSEpro modeling over a range of possible operating conditions for the three independent operating inputs: ambient temperature, ambient pressure, and solar thermal input. We used the results from this parametric analysis to efficiently model the ISCC behavior of this configuration in SAM. We applied a three-dimensional interpolation routine to return performance results for a given set of inputs. Our IPSEpro analysis showed that ISCC behavior is strongly dependent on the design of the NGCC cycle. The design point behavior of the gas turbines, steam turbines, and HRSG all drive off-design performance. Consequently, we found that it is untenable to model modifiable NGCC and ISCC configurations using empirical routines – the off-design performance results between different designs can't be correlated.

Finally we integrated the ISCC power cycle model with SAM's existing molten salt power tower model. This work involved developing a design point model of the molten-salt-to-steam heat exchanger that is solved at the beginning of the simulation, and then developing an off-design model of the heat exchanger for off-design operation. With this first-principles heat exchanger model, we can capture the iterative interaction between the solar field and power cycle. We included the ISCC model as a CSP technology option in SAM. We also worked with GE to include an option in SAM to model their ISCC power cycle. We describe the SAM integration and present system level results in the 2013 fourth quarter report.

Results

This task advances the current work in two distinct ways. First, it adds an ISCC modeling option to the SAM CSP technology menu. This capability will allow users who do not have the time, budget, and/or expertise for custom or proprietary models to quickly study this technology. For example, a CSP researcher could analyze the impact of the solar field design on ISCC plants. Second, this task advances the current work from a technical standpoint by considering the power cycle interaction with the solar field. Modeling this interaction advances understanding of the integrated system design and operation. For example, as the thermal power from the solar field fluctuates, the

steam pressure and mass flow rate in the Rankine cycle also fluctuates. This behavior leads to varying inlet conditions to the solar field, which must be captured to fully understand the system behavior. As documented above, we met the task milestones, summarized as: 1) develop an ISCC model with consultation from industry experts and submit a journal paper documenting the results and 2) include the ISCC model in SAM.

Supercritical Carbon Dioxide Power Cycle Modeling

Motivation

Experimental and modeling research in the last decade has shown that supercritical CO2 (s-CO2) power cycles have the potential to improve thermal efficiency and economics for fossil, nuclear and concentrating solar power (CSP) plants, as well in waste heat recovery and marine systems. These closed-loop cycles take advantage of high density near the critical point to minimize compressor work and yield potentially higher cycle efficiency compared to superheated or supercritical steam cycles at hot temperatures likely for fossil, nuclear, and CSP. Additionally, these cycles are projected to have a smaller weight and volume, lower thermal mass, and less complex power block than Rankine cycles. The realization of these projections may also result in reduced installation, maintenance and operation cost of the system.

Theoretical design-point studies in the literature have reinforced the potential efficiency gains and established rough operating condition estimates for the s-CO2 power cycle at temperatures relevant to CSP applications. However, the off-design performance of these cycles is critical to understanding the system level design and operation and ultimately quantifying annual performance. To address this modeling gap NREL subcontracted, with funding from DOE, a doctoral research project by Dyreby [13] to develop a robust design and off-design model of the simple and recompression cycles. This model uses turbomachinery off-design models validated with experimental data and conductance design and off-design models for the recuperators. These components are then combined in a cycle controller that solves for cycle performance and optimizes free parameters to maximize efficiency. The goal of this task was to integrate Dyreby's models with SAM's existing molten salt power tower model and thermal storage models to enable annual system simulations.

Approach

The simple and recompression cycle models used in this task are described in detail both in Dyreby's thesis [13], a number of papers authored by Dyreby's group, and our 2014 continuation report. Dyreby coded these models in Fortran because SAM was in Fortran at the beginning of his project. First, we converted his models to C++ to integrate with SAM's new framework and validated our version against his. We structured the model into C++ classes that allow us to access the model independently of CSP system models. Consequently, we can easily simulate the cycle model through our in-house scripting and visualization tools.

Dyreby's model focused on cycle design and off-design *cycle* control. For example, his research explores compressor speed control to optimize off-design behavior. As such, his model applied simplified assumptions for the primary heat exchanger and air cooler, and we needed to build detailed models for these components. We developed a

counter-flow heat exchanger model to calculate the design and off-design behavior of the molten-salt-to-CO2 heat exchanger. The user can specify the design by setting the approach temperature input in SAM. Next, we developed a cross-flow, multi-pass, open air to tubular CO2 air cooler. The user can specify the design of the air cooler by setting the approach temperature and fan parasitic load. The off-design air cooler calculates the fan parasitic load required to reach the approach temperature.

Finally, we added the sCO2 cycle as a power cycle option for the molten salt power tower in SAM. One important difference between the sCO2 cycle and the typical CSP steam Rankine cycle is that the temperature gain in the sCO2 primary heat exchanger is significantly lower. Furthermore, because the sCO2 cycle model is a detailed first-principles model, setting the HTF hot side temperature and cycle efficiency constrains the HTF cold side return temperature. Consequently, users need to careful that the cold side temperature they enter in the user interface is consistent with temperature calculated by the cycle design point model at the input cycle efficiency. If the values are significantly different, the user may need to consider updating the receiver design to change the HTF flow path. To guide the user with this comparison, we implemented a cycle design point function in the user interface. The use can select a button that returns the solved cycle design point as a function of the current inputs in the interface. The user can then compare the calculated to the user-specified HTF cold side return temperature. We describe this in more detail in the 2014 fourth quarter and 2015 first quarter reports.

Results

This model allows users to quickly analyze the impact of the s-CO2 power cycle on system performance and financial metrics. For example, the model can be used to refine solar field/receiver designs for a s-CO2 power cycle or understand the effect of receiver outlet temperature and power cycle efficiency on the LCOE of a molten salt power tower with a s-CO2 power cycle. This task also advances the current work by explicitly modeling the iterative interaction between the power cycle, solar field/receiver, thermal storage, and air cooler. Modeling this interaction is critical to understanding the off-design performance of a CSP plan with a s-CO2 power cycle. For instance, the temperature difference between the inlet and outlet of the primary heat exchanger can vary significantly depending on both design point efficiency and off-design operation. This behavior, in turn, affects the hot and cold side molten salt temperatures in the receiver and the energy density of thermal storage.

The sCO2 power cycle continues to gain interest in the CSP, fossil, waste heat, and geothermal communities. This integrated cycle model is a powerful tool to begin understanding how this cycle affects system design and operation. We are involved in several ongoing projects that use and build upon this model, and we expect to incorporate improvements to the model in SAM.

We have received feedback from various SAM users who are exploring this model. One current limitation they highlight is that the system model is significantly slower than the other SAM models. The longer simulations time are due to the sCO2 model optimizing a first-principles model multiple times per timestep. In order to facilitate faster simulations, NREL has a SuNLaMP task in FY16 to implement a pre-processor that solves that cycle

model over a range of expected operating conditions, and then uses interpolation routines to return performance results during the annual simulation.

SolarPILOT Integration

SolarPILOT was integrated into SAM via an API during the final year of the project. Our measure of success is multifold. First, we have made a number of improvements to SolarPILOT's core functionality to enable rapid and flexible simulation capability to effectively replace the previous power tower design and characterization code (DELSOL3). Next, we have developed extensive documentation of SolarPILOT in an interface-linked help system that provides the user with clear descriptions of each input and configuration option. The documentation also provides usage guidance and formulation background.

The milestone relating to SolarPILOT work involves comparing its design and characterization capabilities to DELSOL3 for an analogous case. This comparison has been completed and shows that the number of heliostats for the sample case agrees within 0.022% (target 10%) and the power delivered to the receiver agrees within 0.192% (target 10%). Work in this task has resulted in the development of a journal submission, and the paper documenting the SolarPILOT software and progress made as part of the DOE-funded research in this project is discussed in **Appendix A: SolarPILOT: A Tool for Power Tower Solar Field Layout and Characterization**. The paper will be submitted to the high-impact peer-reviewed journal *Solar Energy*.

User-defined Power Cycle Performance

Motivation

SAM models parabolic trough, linear Fresnel, and molten salt power tower configurations that employ a heat transfer fluid (HTF) to absorb solar irradiance and deliver it as thermal power to a thermodynamic power cycle that utilizes steam as the working fluid. This type of configuration is known as an "indirect HTF" configuration, as opposed to "direct HTF" configurations wherein the power cycle working fluid also passes through the receiver (e.g. direct steam power tower). SAM's default indirect HTF power cycle model is a regression model developed from a detailed first-principles basis Rankine cycle model. This basis model calculates cycle performance over the expected cycle operating range by modeling each cycle component at off-design conditions. The model assumes that deviation in cycle performance at off-design conditions is independent of cycle design and only a function of deviation from the user specified design point [14]. This model generally has been a fast, flexible, and accurate tool for most conventional CSP power cycles. However, some users have requested the capability to model their own Rankine cycle design or to model newer concepts that pursue the aggressive SunShot targets.

Approach

Overview

NREL has developed a user-defined power cycle option for SAM's indirect HTF technology models to meet this growing demand to model diverse and custom cycles. This option presupposes that the user has a custom power cycle model that can be

used to generate cycle performance results over expected operating conditions. The methodology uses a structured design-of-experiments approach to guide and limit the number of custom power cycle simulations required. SAM provides data tables in its User Interface to store the user's performance data. SAM uses this tabular data to build a power cycle regression model that considers single variable effects and two variable interactions. The following sections explain the user-defined power cycle option in more detail.

Custom Power Cycle Model Requirements

SAM's indirect HTF component models use first-principles relationships to model the interaction between physical component design (e.g. receiver tube diameter, absorptivity, etc.), ambient conditions, and plant performance. Consequently, SAM's component models for the receiver, storage, and power cycle must conserve mass and energy as well as track the HTF temperature as the HTF passes between components. In order to integrate custom power cycle data into the existing indirect HTF technology models, the custom power cycle model must accept as inputs the HTF temperature ($T_{HTF,hot}$) and normalized mass flow rate ($\dot{m}$). Ambient temperature (T_{amb}) also influences the performance of thermodynamics power cycles and is the third independent input required of the custom model. Conceptually, the custom model calculates the outputs in the form of Equation (1), where Y represents any model output (e.g. cycle electric power generated). Because the technology models depend on the relationship between temperatures, mass flow rate, and thermal power, it is crucial that the custom cycle model is assuming HTF properties corresponding to the HTF selected in the SAM user interface (UI).

$$Y = f(\dot{\boldsymbol{m}}, \boldsymbol{T_{HTF}}, \boldsymbol{T_{amb}}) \tag{1}$$

The custom model must return calculated metrics that define the cycle's performance; at a minimum, SAM requires the thermal power delivered to the cycle from the HTF ($\dot{q}_{HTF}$) and cycle electric power generated ($\dot{W}_{cycle}$). Given these values, SAM applies Equation (2) to calculate the HTF cold temperature returning to the receiver and/or thermal energy storage ($T_{HTF,cold}$), where c_p is the HTF specific heat at the average of the hot and cold temperatures. SAM also allows the user to optionally report calculated cooling parasitic load ($\dot{W}_{cooling}$) and cycle water use ($\dot{m}_{water}$). Because the cooling parasitic load is optional, the user must be sure that it is consistent with the reported cycle electric power generated. Equation (3) shows the relationship between the cycle net power calculated in SAM's regression model and the values reported from the user's custom model.

$$T_{HTF,cold} = T_{HTF,hot} - \frac{\dot{q}_{HTF}}{\dot{m} * c_p} \tag{2}$$

$$\dot{W}_{net,calculated} = \dot{W}_{cycle,custom} - \dot{W}_{cooling,custom} \tag{3}$$

Custom Cycle Design Point Performance

Because the custom power cycle model must interface with the CSP system defined in SAM, it is important to maintain consistency between the design points in SAM and the

inputs and response of the custom cycle model. The SAM user interface contains user-specified and calculated inputs that define the custom cycle inputs and outputs at design. That is, two of the cycle inputs at design, HTF hot temperature and ambient temperature, are defined on the user interface, while the normalized mass flow rate is defined as 1.0 by convention. Similarly, the design point cycle efficiency, electric power generation, cooling parasitic load, and cycle water use are also defined in the user interface. Consequently, when solved with the design inputs, the custom power cycle outputs should match the corresponding values in SAM.

SAM's regression model requires that the user report the custom cycle model outputs normalized relative to their design values. As such, the normalized outputs will equal 1.0 at the design case.

Sampling the Custom Power Cycle Model

With a custom cycle model meeting the above requirements, the user must populate SAM's data tables with cycle outputs. The goal of the data tables is to accurately capture the custom cycle performance over practical ranges for each of the three independent inputs (for example, the normalized HTF mass flow rate may float between 0.3 and 1.1 during an annual simulation). One way to ensure that the tables represent the custom cycle over its expected operating conditions is to require the user to sample a dense mesh of input combinations. For example, if the user determines that 20 values accurately represent the range of possible values for each input, then the user would need to complete 8000 (i.e. 20^3) custom power cycle simulations. For many detailed process simulation software packages, this is a significant computational burden. Moreover, SAM would need to import all of the calculated data, and the regression model would need to expansively search through the data to find the correct interpolation region at any given set of inputs.

To reduce the computational requirements, SAM uses a multi-level design-of-experiments approach to limit the number of simulations required to represent the custom power cycle model by modeling single variable effects and two variable interactions. This approach requires that the user define *low* and *high* level values for each input, designated in Table 1 by $-$ and $+$ superscripts, respectively. The *low* level value should be less than the input's design value (designated by the $*$ superscript) and greater than or equal to the lowest value of the input's practical range. For example, if the practical range of the normalized HTF mass flow rate is from 0.3 and 1.1 and the design value is 1.0, then the *low* level of HTF mass flow rate could be 0.5 or 0.3, but not 0.2. Similarly, the *high* level value should be greater than the input's design value and less than or equal to the highest value of the input's practical range.

This approach requires nine parametric simulations of the custom cycle model: three for each input. First, the single variable (or *main*) effects are captured by a parametric analysis of the custom power cycle model over the practical range of the respective *main* input with the remaining two inputs at their design values, as shown by Parametric Analyses 2, 5, and 8 in Table 2. Next the *interaction* input for each *main* input is set to its *low* level, and the parametric analyses are rerun, as shown by Parametric Analyses 1, 4, and 7. Finally, the *interaction* inputs are set to their *high* levels, and the process repeated, as shown by Parametric Analyses 3, 6, and 9. In this way, the interaction is

captured for each of the three possible combinations of two independent inputs. If the user selects 20 values to cover the practical range for each independent input, for example, then the approach outlined in Table 2 requires only 180 (i.e. 20*9) custom power cycle simulations.

Table 2: Custom Power Cycle Simulations Required to Populate SAM's Data Tables

SAM table	Parametric Analysis #	Number of Simulations	Custom Model Inputs		
			HTF Hot Temp	HTF Mass Flow Rate	Ambient Temperature
Table 1	1	$N_{T_{HTF,hot}}$	$T^i_{HTF,hot}$ $for\ i = 1..N_{T_{HTF,hot}}$	$\dot{m}^-$	T^*_{amb}
	2			$\dot{m}^*$	
	3			$\dot{m}^+$	
Table 2	4	$N_{\dot{m}}$	$T^*_{HTF,hot}$	$\dot{m}^i$ $for\ i = 1..N_{\dot{m}}$	T^-_{amb}
	5				T^*_{amb}
	6				T^+_{amb}
Table 3	7	$N_{T_{HTF,amb}}$	$T^-_{HTF,hot}$	$\dot{m}^*$	$T^i_{HTF,hot}$ $for\ i = 1..N_{T_{HTF,hot}}$
	8		$T^*_{HTF,hot}$		
	9		$T^+_{HTF,hot}$		

Populating Data Tables in SAM

The user can simulate a custom power cycle model at the conditions in Table 2 to create data that SAM uses in its power cycle regression model. Next, the user must import that data to the data tables in the SAM User Interface. Rather than a unique table for each of the nine parametric analyses listed in Table 2, SAM groups into tables the parametric analyses that were calculated with the same *main* inputs. For example, analyses 1-3 were all calculated over the practical range of HTF hot temperatures. Consequently, the first column in the corresponding data table in SAM contains the HTF hot temperature, with the number of rows matching the number of values in the HTF hot temperature range. Then, the table provides *three consecutive columns for each calculated output: one column for each level of the interaction input.*

Figure 2 shows an example of how data is applied from the parametric analyses in Table 2 to the SAM data table containing results from parametric analyses of the practical range for the HTF hot temperature. Note that the *low* and *high* levels for the *interaction* input, $\dot{m}$, are user inputs above the table. The data table containing the HTF mass flow rate results uses parametric analyses 4-6, and the table containing the ambient temperature results uses parametric analyses 7-9.

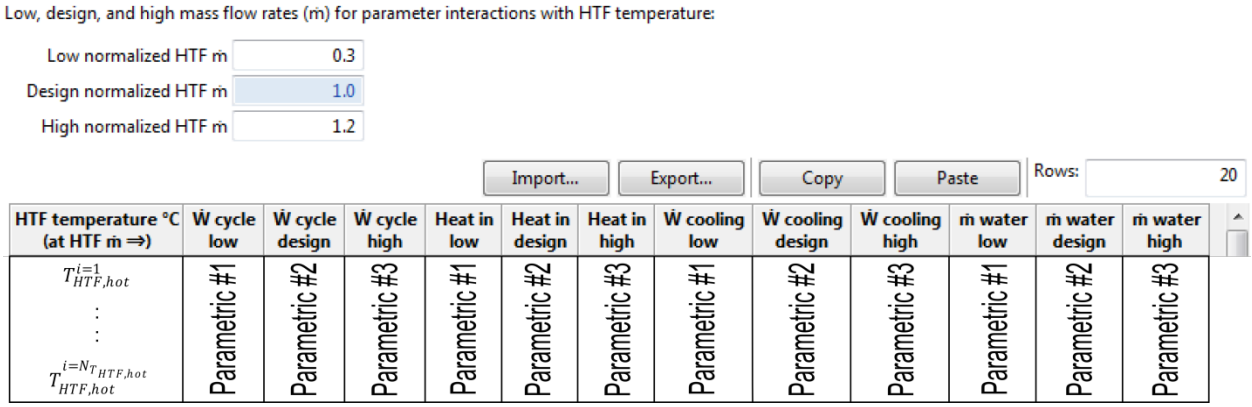


Figure 2: Populating the HTF Temperature data table

SAM's Regression Model

SAM uses the normalized performance data that the use enters in the data tables to calculate cycle performance by fitting the data to the regression model in Equation (4). The three inputs, represented here by $\boldsymbol{T_{HTF,hot}}$, $\dot{\boldsymbol{m}}$, and $\boldsymbol{T_{amb}}$, are passed to the regression model from the other CSP component models, and as such should fall within but not directly coincide with the *main* inputs in the respective data tables. Equation (4) is solved for each of the four outputs, and its normalized output is multiplied to the corresponding design value to calculate the output's absolute value.

$$Y = 1 + f_{ME,T_{HTF,hot}}(\boldsymbol{T_{HTF,hot}}) + f_{ME,\dot{m}}(\dot{\boldsymbol{m}}) + f_{ME,T_{amb}}(\boldsymbol{T_{amb}}) +$$
$$f^{\pm}_{INT,\dot{m}\rightarrow T_{HTF,hot}}(\boldsymbol{T_{HTF,hot}}) * \frac{(\dot{\boldsymbol{m}} - \dot{m}^{*})}{(\dot{m}^{*} - \dot{m}^{\pm})} + f^{\pm}_{INT,T_{amb}\rightarrow \dot{m}}(\dot{\boldsymbol{m}}) * \frac{(\boldsymbol{T_{amb}} - T^{*}_{amb})}{(T^{*}_{amb} - T^{\pm}_{amb})} \quad (4)$$
$$+ f^{\pm}_{INT,T_{HTF,hot}\rightarrow T_{amb}}(\boldsymbol{T_{amb}}) * \frac{(\boldsymbol{T_{HTF,hot}} - T^{*}_{HTF,hot})}{(T^{*}_{HTF,hot} - T^{\pm}_{HTF,hot})}$$

where:

- the $f_{ME,i}(\boldsymbol{i})$ terms represent the main effect of input i, linearly interpolated from the corresponding lookup table at $\boldsymbol{i}$ and the design value of the *interaction* input.
- The superscript $\pm$ refers to either the lower or upper level of the *interaction* input, depending on whether the *interaction* input is less or greater than its design value, respectively.
- the $f^{\pm}_{INT,j\rightarrow i}(\boldsymbol{i})$ terms represent the interaction effect of input j on input i and are calculated two times for each input (one for the upper and one for the lower level of the *interaction* input) from the data tables at the beginning of the a simulation for each value in the practical range of i using Equation (5). When Equation (4) is applied during the annual CSP system simulation, these terms are calculated by linearly interpolating at $\boldsymbol{i}$.

$$f^{\pm}_{INT,j\rightarrow i}(\boldsymbol{i}) = -\left(Table_i(\boldsymbol{i}, \boldsymbol{j}^{\pm}) - 1.0 - f_{ME,j}(\boldsymbol{j}^{\pm}) - f_{ME,i}(\boldsymbol{i})\right) \quad (5)$$

Summary

The following steps define the high-level process to successfully run the user-defined power cycle option in SAM.

1. Develop a custom power cycle model that accepts as inputs the HTF hot temperature, the normalized HTF mass flow rate (with respect to the design point mass flow rate), and the ambient temperature. Ensure that when applying the design point inputs, the calculated outputs match the corresponding values in SAM.
2. For each of the three model inputs:
    a. Select a practical range covering expected cycle operating conditions over the course of the annual simulation. Create a sample of values within this range to accurately capture the cycle response over the operating range (i.e. select the number of values in the range).
    b. Select *low* and *high* levels required when the input is the *interaction* input.
3. Complete the parametric analyses outlined in Table 2.
4. Using Figure 2 as a guide, populate the data tables in SAM using the normalized custom cycle results from the parametric analyses.
5. Run the SAM simulation.
6. Repeat these steps if you modify SAM inputs that affect the custom model results (e.g. the HTF temperature at design is increased).

FY15 Task Milestone

We have completed the milestone to *"include the user-defined power cycle option and documentation in an upcoming SAM release."* The user-defined power cycle option is available for the molten salt power tower, physical trough, and molten salt linear Fresnel models in SAM. A new SAM release containing the new models, improvements, and bug fixes for CSP, as well as updates from other technologies funded outside of DOE-CSP is tentatively scheduled for January 2016. We also wrote documentation that includes sample *Python* code that generates tables that the user can import to the data tables in SAM. We will include this documentation in SAM as a reference for users.

Storage Dispatch Optimization

Motivation

SAM's valuation of energy from CSP plants with storage was inaccurate because it applied a simplistic dispatch model that did not reflect the actual operation strategies adopted by industry since the model was first developed. Instead SAM's dispatch model relied on simple, static heuristics to generate electricity during high value periods. The dispatch optimization model improves the financial performance of the molten salt power tower model (e.g. PPA price, LCOE) by providing the flexibility to operate the plant, especially thermal storage and power cycle behavior, based on locational marginal price. Additionally, the new plant controller and numerical solver models that provide the underlying framework for the dispatch optimization routines improve the stability, predictability, and converge of the system level model. Finally, the model also

provides industry with a broadly-applicable mathematical methodology for optimizing the dispatch of plants with thermal storage.

Modeling Framework Overview

Figure 3 shows a flow diagram of the new modeling framework we've designed to optimize thermal energy storage dispatch and plant operation to maximize revenue. Conceptually, the dispatch optimization model is the highest level, or outer nest, of the framework, and its main objective is to specify to the controller the target turbine output, and component operation Booleans (i.e. is power cycle generation allowed) for a given timestep. The controller is the next level down, and it decides *how* to operate the system components in order to achieve the turbine output and component signals it received from the dispatch optimization model. After the controller choses an operating mode, it passes control to the numerical solver. The solver contains tailored algorithms for each operating mode that allow it to exhaustively search the solution space to either find close convergence of energy and mass balances or report to the controller that the operating mode is unfeasible. The controller then uses its hierarchy to select the next best operating mode.

The division between the dispatch optimization and controller models is perhaps subtle and worth a brief discussion. The former *could* be designed to also select the operating mode, while the latter *could* be designed to also determine the component Booleans and target turbine output. The distinction is that the dispatch optimization model is effective precisely because it is adaptive and only constrained by the bounds of the solution space; it can report infinite combinations of signals to the controller. These calculations require significant computation, and therefore the model necessarily uses simplified proxy models. While the proxy models are generally accurate, they don't approach the convergence tolerances required by the controller. As such, under certain forcing conditions, the dispatch model will predict an infeasible operating mode.

The controller, on the other hand, is a static, heuristic model; it can only search for solutions over a fixed hierarchy. Because of these limited options, the computational effort required to find a solution is negligible. As such, its role is finding the optimal feasible operating conditions after the dispatch optimization model selects its signals. Consequently, it is most useful when the proxy models error such that the first selected operating mode is unfeasible and the hierarchy is required to find a viable solution.

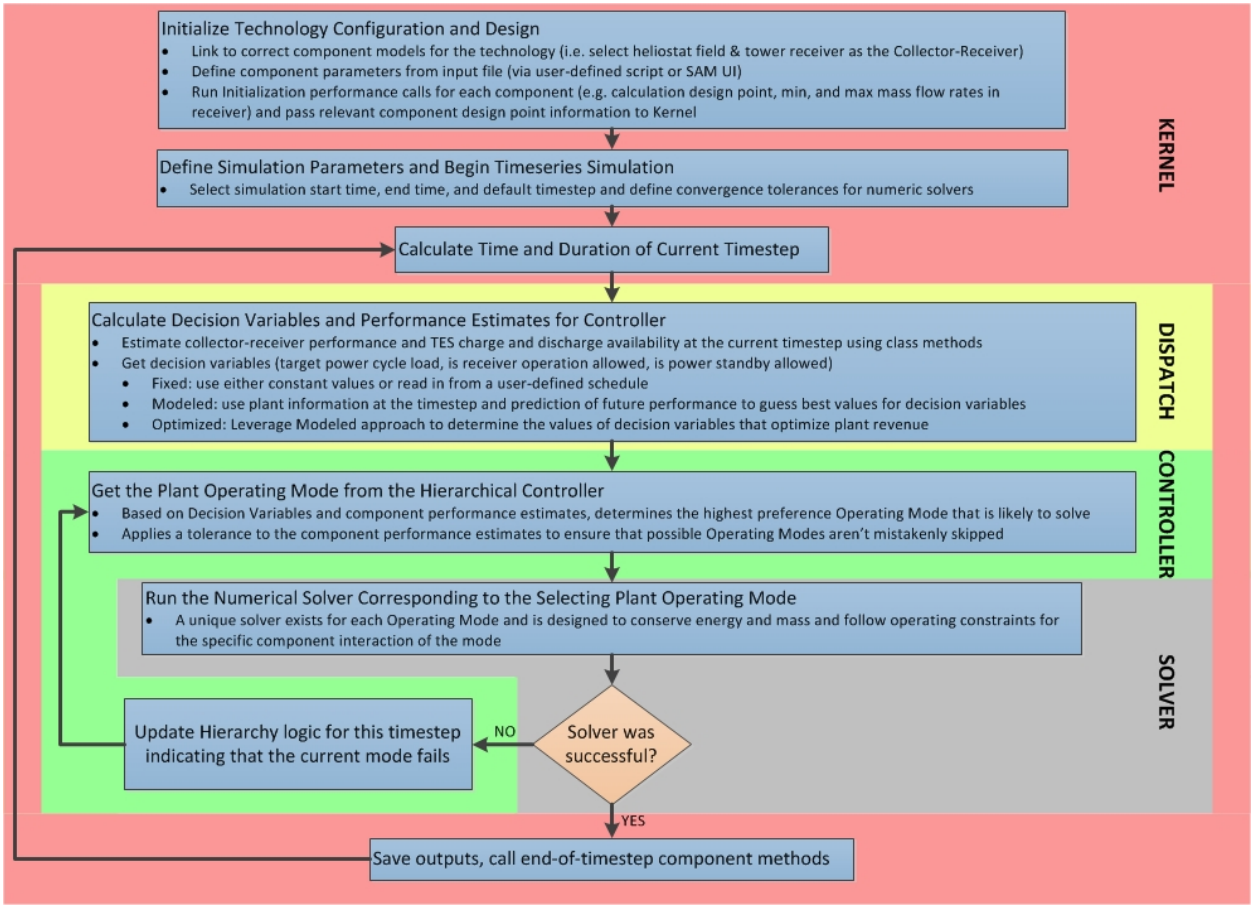


Figure 3: Flow diagram showing the modeling framework that simulates annual plant performance with optimized dispatch.

The Hierarchical Controller and Numerical Solver

Overview

It is worth revisiting benefits of an improved controller and solver. Although the primary objective of including it in this task is to enable the dispatch optimization model, we initially planned it with the goal to improve the accuracy and convergence of SAM's CSP indirect HTF technology models. Although in general the most important outcome of poor convergence is suboptimal representation of system performance, it's helpful to group the sources of convergence failure into two broad categories: Operational and Numerical.

Operational failure occurs when the controller cannot decide on the model's final operating mode. For example, under certain conditions, the molten salt power tower model may be on the threshold of operating at any of the following modes: 1) charging storage and operating the power cycle at full power, 2) operating the power cycle at full power with idle storage, or 3) dispatching storage to achieve full power cycle output. Each mode can result in component performance that is different enough from the previous mode such that the controller logic selects another mode for the next iteration. The controller can become stuck in this pattern of constantly switching modes without iterating over the solution space of a single mode. The end result is that the

convergence at the end of that timestep is often poor and above typical limits. One potential mitigation strategy is a “mode sticking” approach which forces the controller to select a final mode after a fixed number of mode switching iterations. However, oftentimes this approach results in selection of a mode that cannot solve within the physical bounds of the components and results in Numerical convergence failures.

Numerical failures are the failure to find the correct values of component inputs that result in a technology model solution that is both stable (component inputs that are dependent on outputs of other components are equal to those outputs) and obeys mass and energy conservation. For example, consider an operating mode wherein the receiver is charging thermal storage. The receiver inlet temperature is dependent on the storage performance, while the storage performance is dependent on the receiver performance. One solution strategy is to guess the receiver inlet temperature until the calculated receiver inlet temperature (by way of the receiver performance model and storage performance model) is within a close tolerance of the guess. The successive substitution solution strategy in TCS (similar to the solver in TRNSYS) can fail to reach convergence in cases where the component input fluctuates significantly depending on the performance of other components. This occurs most often when the receiver, storage, and power cycle are operating concurrently. When the receiver output mass flow rate is split between storage and the power cycle, the return temperature to the receiver can vary significantly between the storage exit temperature and the power cycle exit temperature. CSP systems employing the steam Rankine power cycle reach a reasonable convergence for most simulations, but are aided by a relatively constant power cycle HTF return temperature at most off-design conditions. The sCO2 cycle, as discussed in previous reports, *may* operate most efficiently at varying HTF return temperatures. However, the TCS solver fails if this temperature is allowed to float. Similarly, it is likely that some user-defined power cycle models also will exhibit this HTF return temperature behavior.

We focused our controller/solver development on solving the convergence challenges described above. Operational convergence issues are mitigated through the implementation of a hierarchical controller. We have implemented the hierarchy such that it first selects the most preferred mode that is likely possible under the given conditions. Then the solver tries to reach convergence at that mode. If the solver does not convergence, it then sends the controller information that it uses to determine the next operating mode in the hierarchy to try. If the solver fails at an operating mode, only a less preferable operating mode can be selected. This strategy guarantees that an optimal operating mode is selected while simultaneously ensuring that mode switching is avoided. The final, least desirable (but always available) operating mode is shutting off all of the component models.

In addition to the significant improvements to model convergence, we designed the controller to efficiently interface with the dispatch optimization model. To this end, we moved all component performance models to stand-alone C++ class based models. This framework allows the dispatch model to interact directly with each component model. For example, the dispatch model can call the molten salt power tower receiver model with guessed inputs for a few hours ahead of its present time step. The dispatch model could then use the results to make decisions about whether to dispatch or charge

storage at its present time step. This component model framework allows the dispatch model to request each component's present operating state. (Component models exist in class-based structures in TCS, but those classes are closely integrated with the TCS framework. We *could* have achieved most of the functionality described in this paragraph within the TCS framework, but the result would have been suboptimal and complicated code that would have restricted or eliminated the other benefits of moving to the hierarchal controller/solver.)

The hierarchal controller also realizes advantages over the current CSP modeling framework relative to simulation resolution. The controller calculates the exact time step required to achieve the receiver and then the power cycle startup. TCS, on the other hand, requires a fixed simulation time step. In the typical case of hourly time steps, the receiver would startup in TCS in less than one hour. Then, the receiver generates hot HTF the remaining portion of the hour. However, the receiver communicates with the power cycle and controller in *power* not *energy* units. Therefore, in order to balance *energy* over the entire time step, the output mass flow *rate* and corresponding thermal *power* is derated in TCS to account for the startup energy during the time step. The result is that the power cycle receives a small receiver *power* output over the entire time step and decides that it can't operate at that level of part-load. In the flexible time step scenario, the power cycle receives the instantaneous *power* output, but only over the second portion of the hourly time step. The result is that, for a subhourly time step, the power cycle receives a *power* output that exceeds its minimum operation (part-load) fraction, and it can startup and generate electricity.

Finally, we developed the component framework for the solver using pure-virtual parent classes for each component type. Essentially, the parent class serves as an API that allows child classes to be built for different variations of the component, given that each defined child class supplies methods for the virtual methods defined in the parent class. For example a collector-receiver parent class would contain a method that takes as inputs HTF temperature and defocus and returns as solved outputs the hot HTF temperature and mass flow rate. One child class may define a heliostat field and molten salt power tower, while another child class may define a parabolic trough collector-receiver field. Analogously, there is a parent class for the power cycle that could define methods for child classes for the steam-Rankine and sCO2 power cycles.

The Controller Hierarchy and Selecting an Initial Operating Mode

The controller receives control signals from the dispatch optimization model or static inputs (Table 3), uses the current component operating states (Table 4-Table 6), and predicts component performance (Table 7). It then applies this information to its hierarchy to select the most preferred operating mode as the initial operating mode. The next section discusses the iteration process if the initial operating mode fails. The innovation of the hierarchy is that it captures the complex relationship between these inputs and the preferred operating mode, as well as the relationship between operating modes.

Table 3: Plant control signals passed to the controller from the Dispatch model or static user inputs.

is_rec_su_ok	Is the receiver allowed to start-up or stay on?
is_pc_su_ok	Is the power cycle allowed to start-up or stay on?
is_pc_SB_ok	Is the power cycle allowed to enter Standby mode?
q_PC_target	What is the preferred thermal power input to the power cycle?

Table 4: Collector-receiver operating states

_OFF	Off: the HTF is not absorbing energy. The receiver is either drained or recirculating HTF, depending on the technology.
_SU	Starting-Up: the receiver is recirculating mass flow to safely heat its material to operating conditions. Start-up is complete when user-defined absorbed energy and start-up time requirements are met.
_ON	On: the receiver is producing hot HTF that must be used by thermal storage or the power cycle.
_DF	Defocus: the receiver is producing more hot HTF than the thermal storage and power cycle can accept. This mode defocuses the collectors until the receiver mass flow decreases to an acceptable rate.

Table 5: Thermal energy storage operating states

_OFF	Off: thermal energy storage is idle. Thermal losses to the environment are still calculated and freeze protection is applied if necessary.
_DC	Discharge: thermal energy storage is discharging thermal power.
_CH	Charge: the receiver HTF output is charging thermal energy storage.
_EMPTY	Empty storage: thermal energy storage is completely discharged over the timestep.
_FULL	Fill storage: thermal energy storage is fully charged over the timestep.

Table 6: Power cycle operating states and user-defined operational limits

_OFF	Off: the power cycle is not receiving mass flow from thermal storage or the receiver and requires startup energy before it can produce power.
_SU	Starting-Up: the power cycle is receiving mass flow from thermal storage or the receiver to safely heat its components to operating temperature. Start-up is complete when user-defined absorbed energy and start-up time requirements are met.
_SB	Standby: the power cycle is receiving enough mass flow from thermal storage or the receiver to stay at its operating temperature, but not enough to produce power. If during the next timestep the power cycle receives more mass flow, it can produce power without meeting cold start-up requirements.
_TARGET	Target thermal input: the power cycle requests the target (defined by dispatch model for each timestep) thermal input from the receiver and

	thermal storage.
_MAX	Maximum thermal input: the power cycle is operating at its user-defined maximum thermal input.
_MIN	Minimum thermal input: the power cycle is operating at its user-defined minimum thermal input.
_RM_HI	Resource match – high: the power cycle allows flexible thermal input between its target and maximum values, such that its thermal input matches the thermal power available from the receiver and thermal storage.
_RM_LO	Resource match – low: the power cycle allows flexible thermal input between its target and minimum values, such that its thermal input matches the thermal power available from the receiver and thermal storage.
q_PC_min	Minimum allowable thermal power input to the power cycle to produce power
q_PC_max	Maximum allowable thermal power input to the power cycle
q_PC_sb	Minimum allowable thermal power input to the power cycle to maintain standby operation

Table 7: Component performance estimates, completed at the beginning of each timestep

q_CR_on	Estimate of receiver thermal power output assuming steady state operation
q_CR_su	Estimate of receiver thermal power available for receiver startup
q_TES_ch	Maximum available thermal power thermal energy storage can accept
q_TES_dc	Maximum available thermal power thermal energy storage can discharge
tol	Relative value applied to estimated performance values that account for the uncertainty of the estimate. Applying the tolerance in decision point logic inequalities ensures that higher preference operating modes are not missed due to low performance estimates.

The following four figures (Figure 4-Figure 7) show hierarchical controller flow diagrams. The following information explains important features of the diagrams:

- Each timestep, the controller uses one of the hierarchies depending on operating states of the collector-receiver and power cycle. These entry criteria are described in the green shapes.
- The controller navigates the hierarchy using decision point logic in the tan diamonds and the black arrows corresponding to whether the logic is true (YES) or false (NO).
- Operating modes are listed in the blue rectangles. Red outlines of the boxes indicate that the timestep of the operating mode is calculated by the solver and

may be shorter than the baseline timestep set by the simulation kernel (e.g. collector-receiver startup).

- Finally, the red dotted arrows show the few occasions when iteration on the operating model requires moving to a different branch on the hierarchy. This is discussed in more detail in the following section.

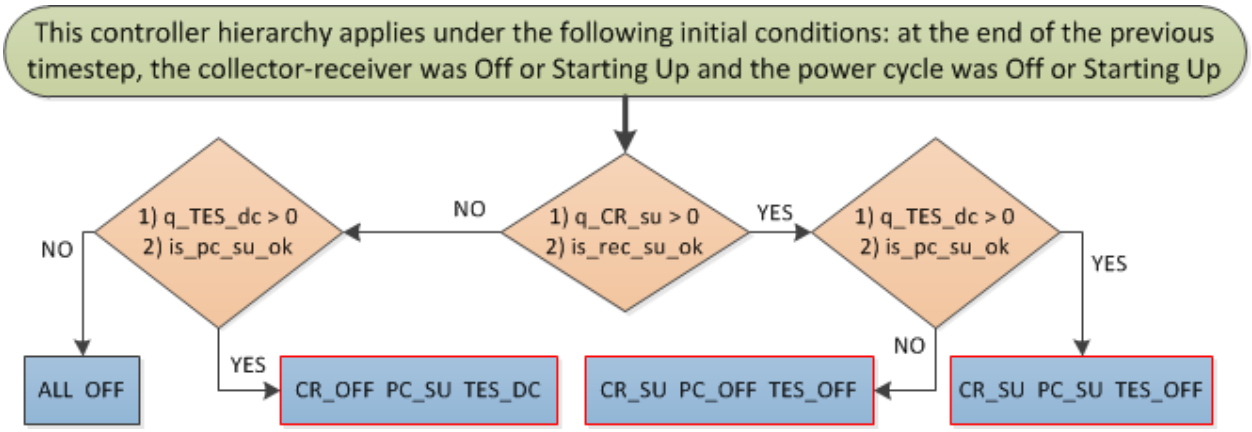


Figure 4: Hierarchy for initial condition of the collector-receiver and power cycle both off or starting up.

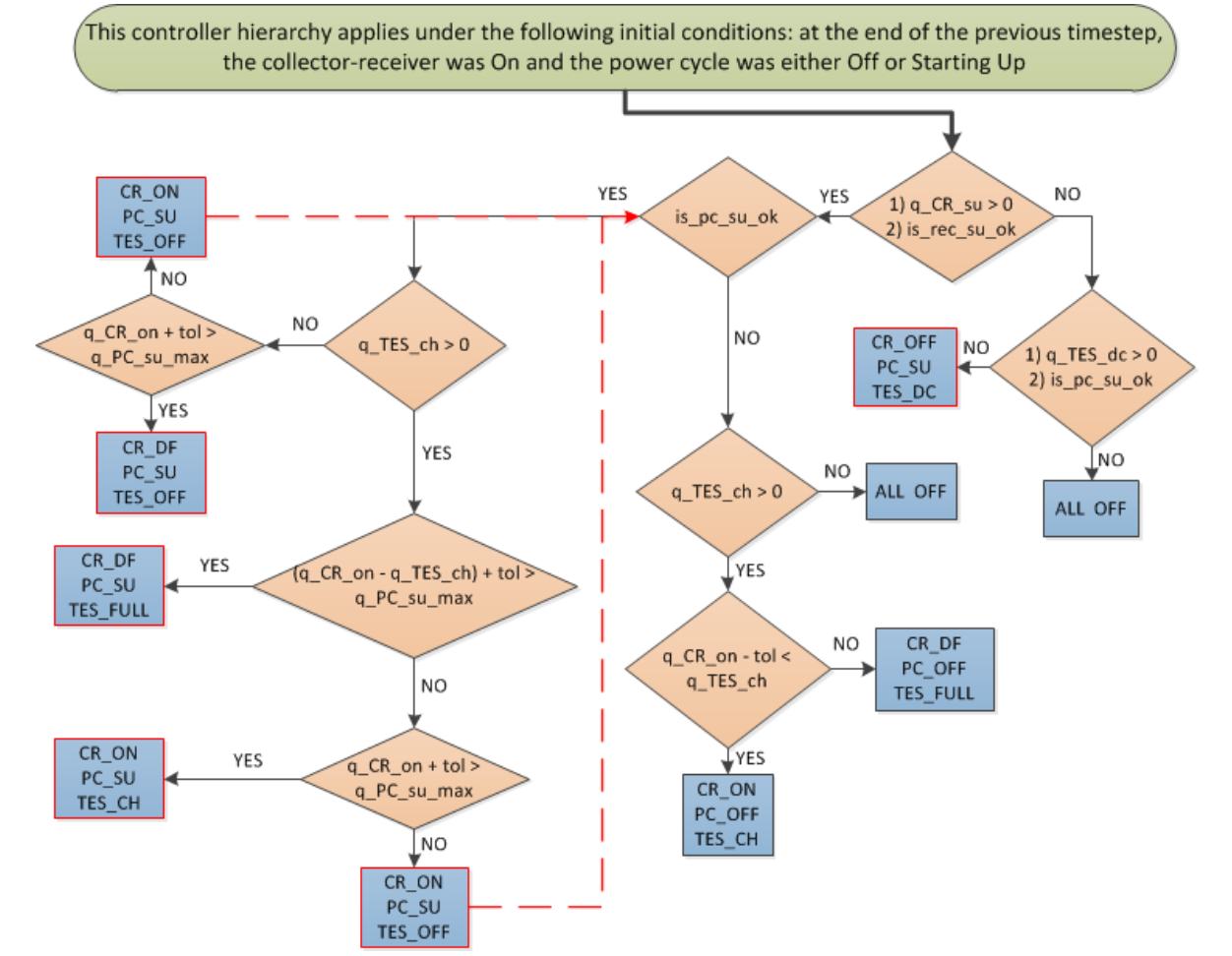


Figure 5: Hierarchy for initial condition of the collector-receiver on and power cycle off or starting up.

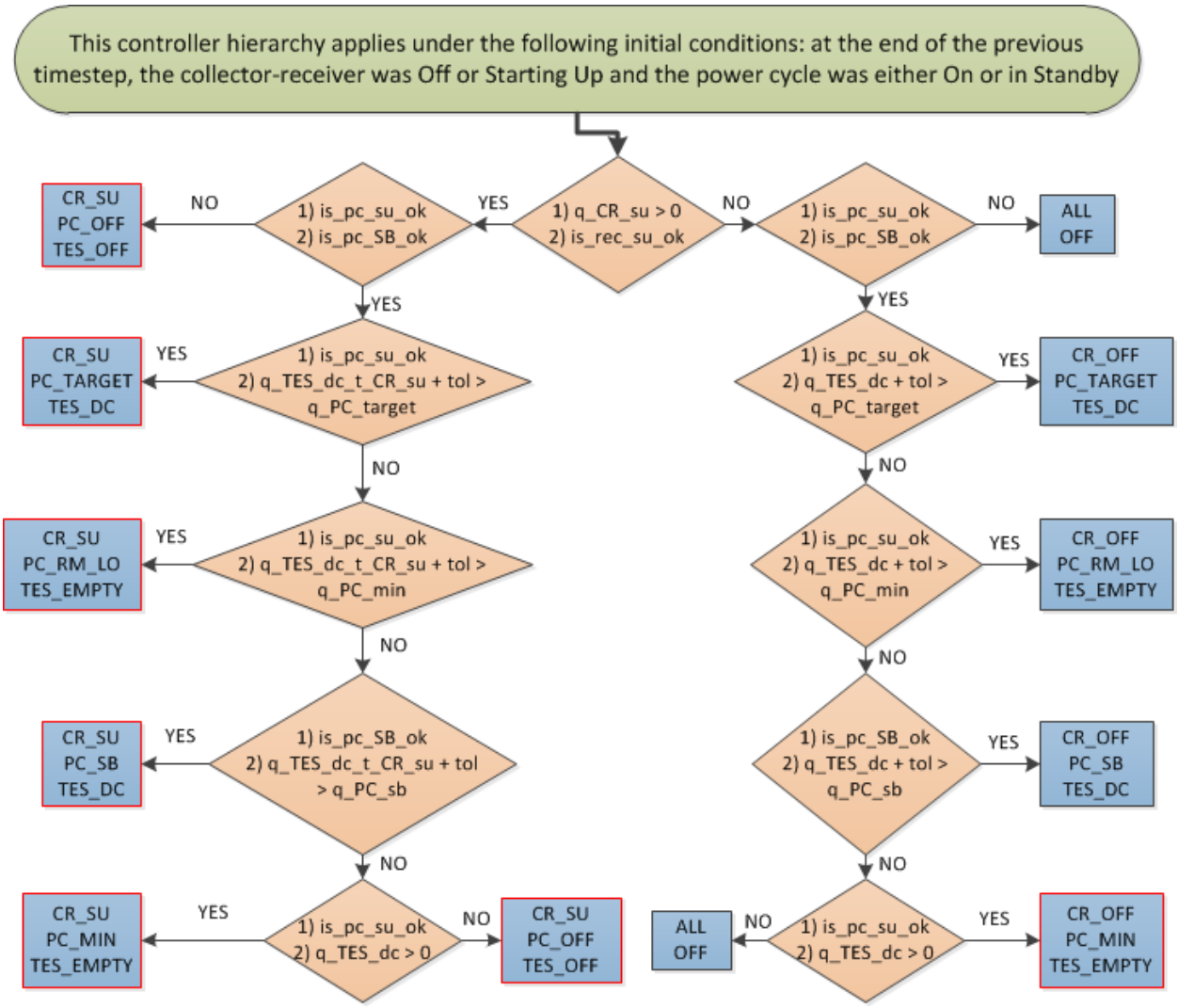


Figure 6: Hierarchy for initial conditions of the collector-receiver off or starting up and power cycle on or in standby.

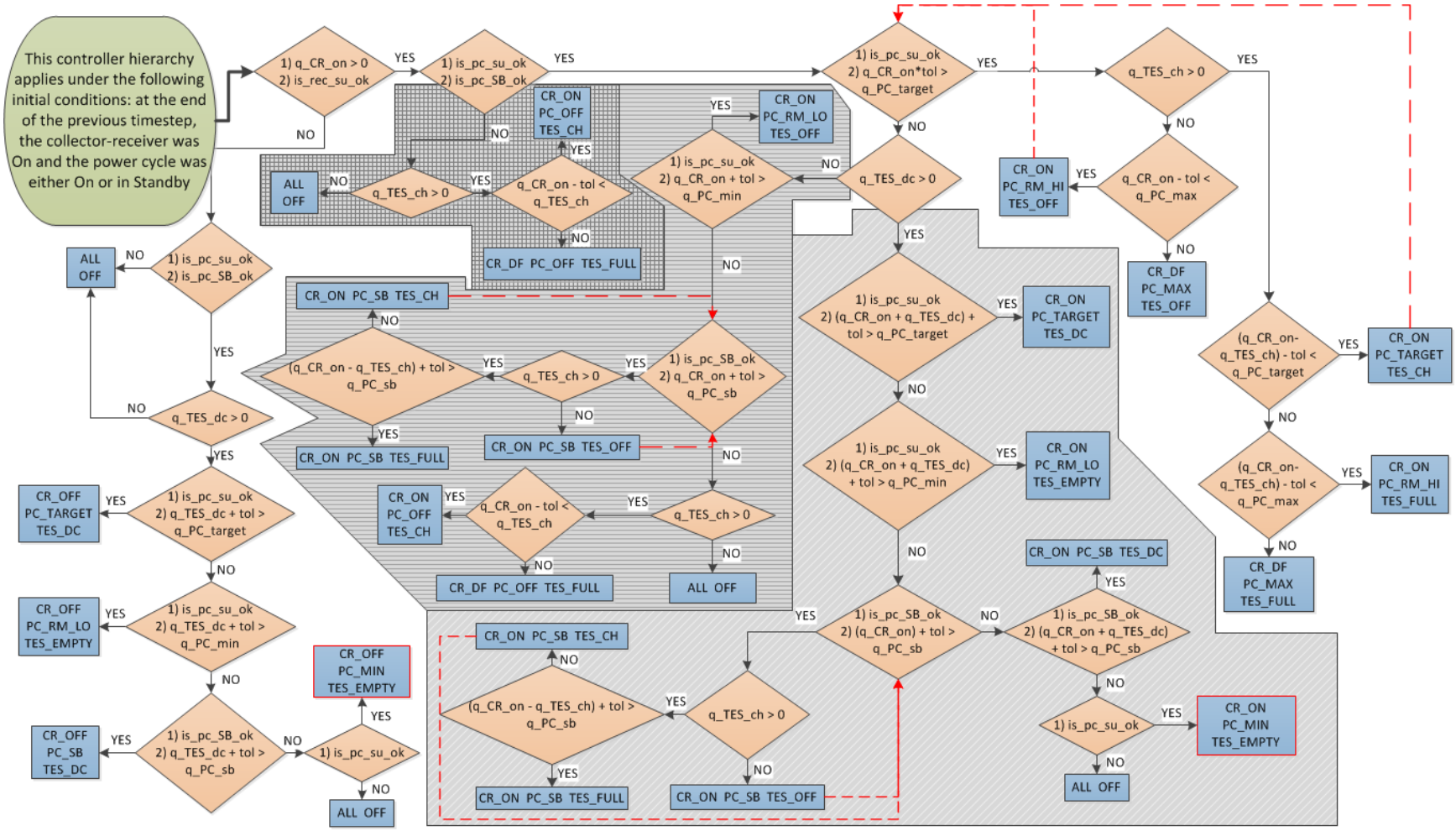


Figure 7: Hierarchy for initial condition of the collector-receiver on and power cycle on or in standby. Shading is intended to help differentiate difference branches.

Operating Mode Iteration

Figure 4 through Figure 7 shows hierarchies for different component operating state initial conditions. As shown in the figures, we've included a tolerance in the hierarchy decision point logic inequalities to ensure that higher preference operating modes are always tried in the case that performance is underestimated. For example, the logic may compare the estimated receiver output to the minimum power cycle operation fraction. The best outcome for this decision point is that the receiver can supply at least enough thermal power to meet this minimum fraction. Therefore, we add the tolerance to the receiver output and then compare that summation to the minimum fraction. If the summation is still less than the minimum fraction, we are comfortable that that preferred branch is not possible under those conditions.

On the other hand, the summation including the tolerance in this example may be only marginally greater than the minimum fraction. In that case, when the preferred operating mode is selected, the numerical solver will determine that the mode cannot be solved within the constraints (in this example, the power cycle will not be able to operate at its minimum operation fraction). As a result, the solver needs to communicate to the controller that it failed to solve the selected operating mode, and then the controller needs to find the next best operating mode to try.

Figure 8 shows an example of the process to find the optimal operating mode for two iterations. First, the controller selects the highest preference operating mode given the initial conditions and performance estimates: the collector receiver is on and the power cycle is operating at a level between its minimum and target operating fractions to match the collector receiver output. However, the solver determines that when the detailed collector receiver and power cycle performance models are evaluated as a system, the collector receiver cannot supply enough thermal power to the power cycle to operate at its minimum fraction. Then the solver communicates to the controller that this operating mode is not feasible. The controller hierarchy is updated such that the decision point that led to the failed operating mode is now hard-coded to avoid that operating mode. Therefore, the second iteration through the controller hierarchy bypasses the failed operating mode and selects the next highest preference operating mode – in this example the collector receiver is on while the power cycle operates in standby. If this mode fails, the solver will instruct the controller to avoid this operating mode as well, and the third iteration would advance to the "additional decision points…" where a new, lower preference operating mode would be selected. The final, lowest preference, operating mode is always shutting off each component.

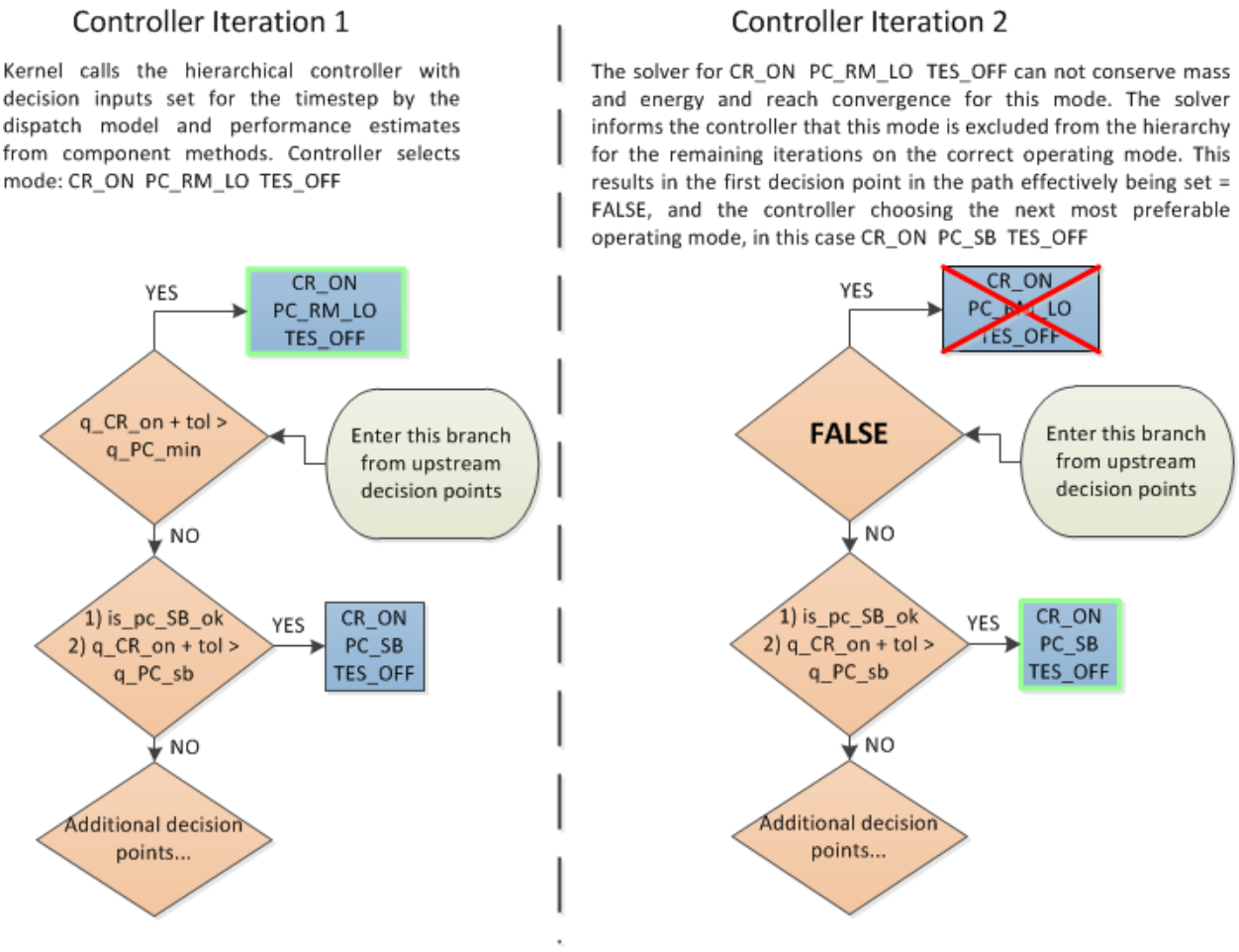


Figure 8: Iterative interaction between the hierarchical controller and numerical solvers to find the correct operating mode.

Finally, there are a few operating modes wherein a solver failure necessitates switching branches in the hierarchy. Figure 9 shows an example of the hierarchy where this switching could occur. Some set of inputs may result in the first decision point logic being marginally true when the tolerance is applied. In that case, assuming no thermal storage charging capacity and a maximum turbine fraction greater than the target, the controller will select mode CR_ON PC_RM_HI TES_OFF from the hierarchy. However, in this case when the solver attempts to solve this mode, it will not be able to match or exceed the target turbine fraction. This is because the logic in the *first* decision point in the diagram is incorrect, not due to the logic in the decision point directly upstream. Consequently, the first decision point is hardcoded false, and the next iteration branches off of the example shown. The red dotted line indicates that the operating mode interacts with the first decision point.

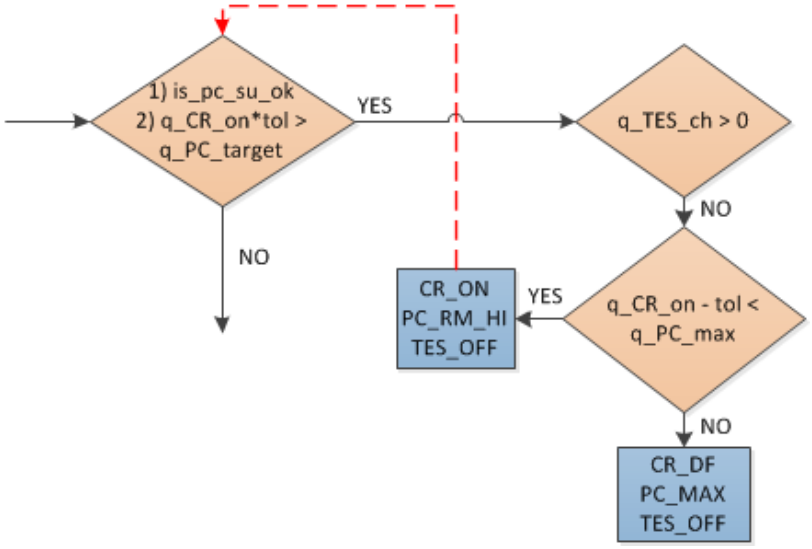


Figure 9: Example section from the hierarchy for the initial conditions of the collector-receiver on and the power cycle highlighting operating mode iteration that requires switch hierarchy branches.

Time Series Results

Annual results are discussed later in this report. Because one of the main goals of the controller and solver was to improve convergence, it is important to confirm that we achieved that outcome. Figure 10 shows the relative error of mass and energy balances for a plant configuration with four hours of thermal energy storage, a solar multiple of 1.5, and a constant turbine fraction of 1.0. The energy balance in plot e) shows excellent energy conservation in both the TCS and new modeling framework. However, the mass conservation in plot a) shows that TCS tends to achieve poor mass balance convergence when the receiver, thermal storage, and power are all active. Inversely, TCS does not experience significant mass conservation errors for plant configurations with no storage.

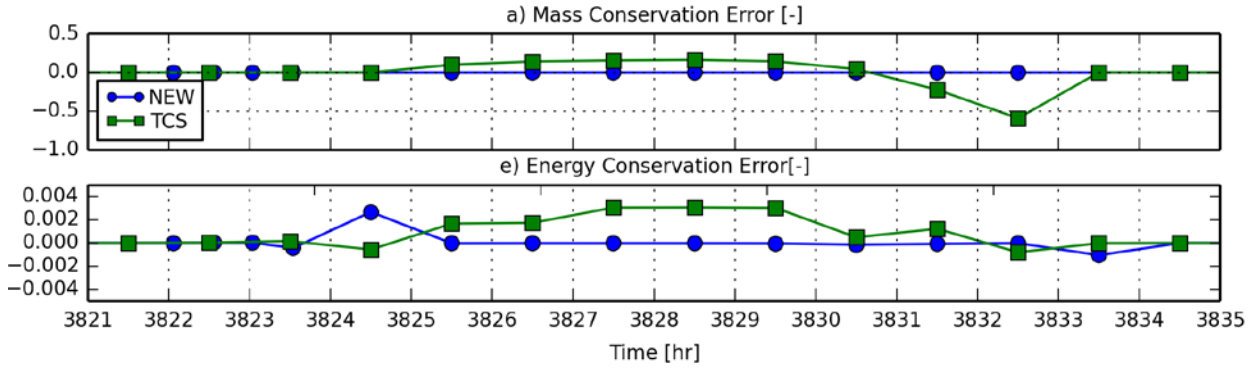


Figure 10: Sunny day comparison of mass and energy conservation relative errors for the plant configuration with four hours of thermal energy storage.

Dispatch Optimization

We have successfully implemented and demonstrated a dispatch optimization methodology in SAM. The new capability meets the targeted milestone by demonstrating improved revenue results (via PPA price reduction) for an optimized case versus the previous “block dispatch” case. The results of this work are presented in

detail in **Appendix B: Concentrating Solar Power Energy Dispatch Optimization in a Detailed Annual Simulation Model**. This paper will be submitted to the journal Applied Energy.

Dispatch optimization relies on the premises that (a) future thermal energy generation from the solar field can be predicted with some accuracy, (b) a CSP plant can be operated according to a pre-specified production schedule, and (c) the behavior of a CSP plant over time can be approximated using a model comprised exclusively of constant, linear, and binary variables (a “mixed integer linear program” or MIP). In this project, we developed a simplified forecast model and a MIP that is evaluated by a solver to determine the optimal production schedule. The MIP takes as input the thermal energy production forecast from the solar field, the expected electricity sales price or tariff multiplier schedule, and plant sizing parameters. It simultaneously solves for the optimal electricity generation profile while treating as variables the charge state of thermal storage over time, the energy use by the power cycle, and energy use for system startup. The objective of the solver is to maximize expected revenue from the plant over the optimization time horizon, and revenue is modeled as the product of electricity production and a price multiplier for each hour in the horizon.

Once the optimal electricity generation schedule is identified using the MIP solver, the corresponding target power cycle thermal consumption and operational states of the power cycle and receiver are passed to the simulation controller. The controller replicates the optimal profile as best as possible using the methodology described in previous sections. The dispatch optimization process determines the optimal schedule every 24 hours of the simulation and uses a look-ahead window of 48 hours (by default) to allow for day-to-day carryover of thermal storage.

The effectiveness of dispatch optimization is demonstrated by considering four tariff schedules with differing daily price multiplier profiles. Each schedule seeks to emphasize a different aspect of CSP plant operation. We first considered the impact of dispatch optimization in comparison to the previous heuristic dispatch model with respect to power purchase agreement (PPA) price. This financial metric is a good indicator of overall plant profitability, whereas LCOE does not capture the impact of hourly revenue multipliers. The first analysis investigates the optimal sizing of thermal storage and solar multiple for a plant with optimized dispatch and a plant with heuristic dispatch. The results of this study show that dispatch optimization substantially reduces PPA price in all tariff scenarios. The reduction in PPA price ranges from 2.9% to 14.2%. The results also show that dispatch optimization can change the optimal sizing of the solar multiple and thermal storage systems. For example, one tariff schedule indicated a minimal PPA price at a solar multiple of 2.2 and TES size of 5 hours for the heuristic dispatch case and a solar multiple of 2.0 and TES size of 7 hours for the optimized dispatch case. This illustrates a counter-intuitive design relationship where solar multiple is reduced and TES sizing is enhanced because of dispatch optimization. Notably, this case corresponded to a reduction in PPA price of 14.2%.

Finally, the tariff case studies illustrate the ability of dispatch optimization to reduce electricity generation variability during peak pricing hours. Heuristic dispatch algorithms create repetitive profiles based on simple rules, and the electricity generation may not correspond well with peak pricing. Optimized dispatch can reserve energy in storage

until the peak pricing periods – whether that peak occurs in the early morning, late afternoon, or another time of day – and generate electricity to maximize revenue. The analysis shows that peak period variability is small over the course of the year, and this implies the viability of CSP for serving peaking or ancillary services markets. We refer the reader to Appendix B for detailed discussion and presentation of the results.

SAM User Interface Implementation and Default Case Comparison

The dispatch optimization model requires new user interface inputs for the user to enable and configure the model. We also realized from developing the controller and solver that we could more clearly differentiate on the interface inputs to the system design, system components, and system control. This led to an overhaul of the molten salt power tower layout, including the following changes:

- We created a System Design page that contains high level system inputs like hot and cold HTF design temperatures, solar multiple, power cycle output, and hours of thermal storage. These values are then displayed on the component pages to help inform component design.
- Similarly, we created a System Control page that contains system level operation and control inputs like balance-of-plant parasitics, TOU schedules, turbine outlet fractions, and dispatch optimization configuration.
- We removed the parasitics page and relocated those inputs to their respective component pages.

Finally, we made a number of other changes to the molten salt power tower component performance models during the development of the controller and solver. Some of these changes were bug fixes; others were made to achieve better consistency with user expectations and input definitions. The annual net electric output for the default case, without dispatch optimization, is now roughly 3.5% greater than in the previous release. This increase is due to the following changes:

- We moved the tower piping heat losses from a parasitic applied to the electric output to calculated thermal losses in the receiver energy balance model. This resulted in about 1.0% increase for the default case.
- We are allowing the receiver and power cycle to startup in subhourly timesteps (using hourly weather data), as discussed in detail above. The result is about 0.5% increase for the default case.
- Finally, the largest difference results from a change in how the TES volume is calculated. We are now assuming that 2-tank TES system can deliver at design the rated hours of storage after the tank fluid minimum height is considered. Consequently, when this input is increased, the tank volume increases. Our old approach sized the tanks without considering this input. So, for the default case, our new approach results in a larger tank that effectively adds more hours of storage relative to the previous version.

This model is now available in an NREL-internal release of SAM, with new Help features for the dispatch optimization features. NREL has distributed this version to Beta testers and is expecting feedback in December. Additionally, the paper describing the dispatch optimization methodology attached in the Appendix will be available to users upon publication. A new SAM release containing the new models, improvements, and bug fixes for CSP, as well as updates from other technologies funded outside of DOE-CSP is tentatively scheduled for January 2016. Given these results, NREL has completed Milestone 1 for this task: "*A publicly available dispatch optimization model in SAM with corresponding documentation and a user's manual detailing the methodology and sample results.*"

References

[1] "TRNSYS 16 a TRaNsient SYstem Simulation program," University of Wisconsin - Madison Solar Energy Laboratory, 2007.

[2] A. Dobos, T. Neises, and M. Wagner, "Advances in CSP simulation technology in the System Advisor Model," *Proc. 2013 SolarPACES Int. Symp.*, 2013.

[3] E. J. Sheu, A. Mitsos, A. a. Eter, E. M. a. Mokheimer, M. a. Habib, and A. Al-Qutub, "A Review of Hybrid Solar–Fossil Fuel Power Generation Systems and Performance Metrics," *J. Sol. Energy Eng.*, vol. 134, no. 4, p. 041006, 2012.

[4] J. Dersch, M. Geyer, U. Herrmann, S. a. Jones, B. Kelly, R. Kistner, W. Ortmanns, R. Pitz-Paal, and H. Price, "Trough integration into power plants—a study on the performance and economy of integrated solar combined cycle systems," *Energy*, vol. 29, no. 5–6, pp. 947–959, Apr. 2004.

[5] B. Kelly, U. Hermann, and M. J. Hale, "Optimization Studies for Integrated Solar Combined Cycles," in *Proceedings of Solar Forum 2001*, 2001.

[6] A. C. McMahan and N. Zervos, "Integrating Steam Generation from Concentrating Solar Thermal Collectors to Displace Duct Burner Fuel in Combined Cycle Power Plants."

[7] C. O. Ojo, D. Pont, E. Conte, and R. Carroni, "Performance Evaluation of an Integrated Solar Combined Cycle," in *ASME TurboExpo*, 2012, pp. 1–8.

[8] M. J. Montes, A. Rovira, M. Muñoz, and J. M. Martínez-Val, "Performance analysis of an Integrated Solar Combined Cycle using Direct Steam Generation in parabolic trough collectors," *Appl. Energy*, vol. 88, no. 9, pp. 3228–3238, Sep. 2011.

[9] C. Libby, "Solar Augmented Steam Cycles for Natural Gas Plants: Conceptual Design Study, Report 1018645," Electric Power Research Institute, Palo Alto, CA.

[10] "Solar Augmented Steam Cycles for Natural Gas Plants: Chuck Lenzie Generating Station Development Guideline Manual," Electric Power Research Institute, Palo Alto, CA.

[11] “Solar Augmented Steam Cycles for Natural Gas Plants: Griffith Energy Facility Development Guideline Manual.”

[12] G. Zhu, T. Neises, C. Turchi, and R. Bedilion, “Thermodynamic evaluation of solar integration into a natural gas combined cycle power plant,” *Renew. Energy*, vol. 74, no. 2015, pp. 815–824.

[13] J. J. Dyreby, “Modeling the Supercritical Carbon Dioxide Brayton Cycle with Recompression,” University of Wisconsin-Madison, 2014.

[14] M. J. Wagner and P. Gilman, “Technical Manual for the SAM Physical Trough Model.” NREL, Golden, CO, 2011.

SolarPILOT: A Tool for Power Tower Solar Field Layout and Characterization

Michael J. Wagner[c,a,d,*], Tim Wendelin[c,b]

[a]*Mechanical Engineer*
[b]*Senior Optical Engineer*
[c]*National Renewable Energy Laboratory, Thermal Systems Group, 15013 Denver West Parkway, Golden, CO 80401*
[d]*Colorado School of Mines, 1500 Illinois St., Golden, Colorado 80401*

Abstract

A new Solar Power tower Integrated Layout and Optimization Tool (SolarPILOT) is developed and demonstrated. The tool advances the analytical flux image Hermite series approximation used in DELSOL3 by applying it to individual heliostat images rather than "zones", allowing characterization of a wide variety of heliostat field layouts. A number of improvements to the analytical approximation method have been developed and implemented in SolarPILOT to improve model accuracy and computational efficiency. Several of these methods are discussed in this paper, including a method for dynamic heliostat grouping to reduce the expense of intercept factor evaluation, methods for approximating annual productivity with a subset of time steps throughout the year, a polygon clipping method to accurately calculate inter-heliostat shadowing and blocking, methods for receiver and tower geometry optimization, and an trigonometric image transform algorithm that ensures analytical equation accuracy for small heliostats. SolarPILOT also integrates the SolTrace Monte-Carlo ray tracing engine, providing improved receiver optical modeling capability, a user-friendly front end for geometry definition, and side-by-side validation of the analytical algorithms.

Keywords: SolarPILOT, Power Tower, Heliostat Layout, Optical Characterization, Simulation Optimization

1. Introduction

Power tower systems (also known as "central receiver systems") are optically complex, using thousands of individually-tracking heliostats to reflect sunlight onto a stationary receiver throughout the day and the year. The angular acceptance window for the reflected image from a heliostat is typically very small, requiring tracking precision with an error distribution standard deviation on the order of 1 mrad or less. In addition, receiver operation typically requires that the incident flux density be maintained below a maximum value, and heliostat images must be strategically placed on the receiver to achieve a practicable distribution [1, 2, 3] that extends the receiver material lifetime and minimizes optical interception losses. The redirection of sunlight by the heliostat field is also subject to a series of losses that depend on the heliostat's position relative to the receiver, the position and orientation of neighboring heliostats, the position and apparent shape of the solar disc, the particulate content in the atmosphere, the geometry of the heliostat, optical errors in the heliostat, and the heliostat field operation strategy. Many of these losses are dynamic in time and must be modeled over a range of conditions in order to adequately characterize the likely performance of a plant, as shown previously by the authors and others [4]. Consequently, computer software has been used to generate solar field geometry and characterize its performance since the late 1970's [5, 6, 7, 8].

The history of available codes extending from first-generation tools through current solutions is well-documented [9, 10]. A number of power tower tools have been developed to support the various stages of analysis necessary to characterize system performance. Codes such as the University of Houston Codes (UHC - also known as the RCELL suite) [8], DELSOL3 [11], TieSOL [12], and HFLCAL [13] can be used to generate solar field geometry programmatically. Other codes such as MIRVAL [6], HELIOS [14], STRAL [15], Tonatiuh [16], and SolTrace [17] are capable of detailed field characterization but are not designed to quickly generate and optimize solar field geometry[1]. Finally, given a particular geometry, several codes are capable of characterizing the annual performance of tower systems, including Solergy [18], System Advisor Model (SAM) [19], and the TRNSYS STEC library [20]. Because these various tools emphasize different aspects of power tower solar field design or characterization, each must be used deliberately within the scope of the problem that it addresses.

1.1. Modeling approaches

The aforementioned tools characterize optical performance using one of two general approaches: analytical (or semi-analytical) approximation or Monte-Carlo Ray-Tracing (MCRT). The basis for analytical methods lies in modeling a

*Corresponding author
Email addresses: michael.wagner@nrel.gov (Michael J. Wagner), tim.wendelin@nrel.gov (Tim Wendelin)
URL: www.nrel.gov/csp (Michael J. Wagner)

[1] DLR has developed an extension for MIRVAL that facilitates automated field layout [10]

reflected image with a closed-form density function. Most simply, an image can be approximated using a Gauss-normal distribution with standard error deviation defined in one or two dimensions.

Multiple physical effects – each with their own error distribution – can impact the overall image error distribution. One approach for modeling multiple error factors is to simply convolve the various error sources as independent normal distributions into a single normal distribution described by a standard deviation in each dimension. This approach limits the shapes of the reflected images that can be modeled, but may be appropriate for heliostats with certain optical properties. A more nuanced approach utilizes the truncated Hermite polynomial series to describe the image shape in two dimensions [7, 5]. This method accommodates non-normal sun shape distributions and can accurately represent flux patterns for flat, focused, or canted heliostats at a variety of tracking angles. This method is the basis for DELSOL3 and UHC/RCELL. The primary advantages of this approach are its computational efficiency in comparison with ray-tracing methods, flexibility in describing complex flux shapes with continuous functions using relatively few expansion coefficients, and its corresponding ability to accurately determine intercepted power on the receiver using integration by quadrature.

One limitation of the Hermite method is that directional information is not preserved in the analytical approach. This makes analysis of multiple reflections or beam-spread within a cavity receiver non-trivial [21, 22]. Furthermore, unlike MCRT, shadowing and blocking must be handled independently of flux image calculations, and accounting for partial shadowing/blocking exclusions in the final image shape is not straightforward.

The MCRT approach is widely used in optical analysis as it offers easy implementation, flexibility in the geometry that can be modeled, preservation of directional information through multiple reflections, and a clear physical analog. Codes such as SolTrace, MIRVAL, and Tonatiuh offer solutions for power tower modeling that can account for the various error sources and shapes, and can characterize non-ideal reflector surfaces as obtained by high-resolution surface slope measurements (e.g. VSHOT [23]). The primary disadvantage of MCRT approaches is their relatively long run times. This is especially true for power tower heliostat fields where ray intersections are possible over a large number of geometrical entities and many rays are required to obtain convergence. However, Izygon (2011) and others have developed an innovative hardware solution using graphical processing units (GPUs) that enables massive parallelization, greatly improving run time [12] but often requiring additional graphics processing hardware.

With these considerations in mind, the Hermite analytical approach has traditionally been used in optimization tools where many simulations are required to determine an optimal system configuration. For example, the DELSOL3 code was implemented in System Advisor Model [24] and was capable of first generating an approximately optimal solar field layout, tower height, and receiver size, then characterizing the solar field efficiency and receiver flux profile over a range of solar positions in less than ten seconds using a standard laptop computer. In comparison, a single run in SolTrace for a power tower system with 5,000 heliostats and a peak flux uncertainty of 1.1% (1×10^6 rays) requires just under two hours on a standard laptop computer running 4 parallel threads. By integrating the analytical and MCRT engines, SolarPILOT provides rapid layout capabilities with more flexible MCRT characterization options.

2. Tool description

SolarPILOT provides layout, characterization, parametric simulation, plotting, and optimization capabilities via a graphical user interface (Figure 1. Limited functionality is also currently available through a C++ Application Programming Interface (API). SolarPILOT has been integrated into SAM via the API, and now serves as the power tower characterization engine. An important aspect of SolarPILOT is the integration of both analytical and raytrace methods in the software. The following sections describe the methodologies in more detail.

2.1. Analytical methods

SolarPILOT extends the Hermite method implemented in DELSOL3 by applying the optical model to individual heliostats to simulate whole-field performance. This higher-resolution approach differs from DELSOL3 in its treatment of individual heliostats, rather than a coarse cylindrical coordinate zonal grid. The implementation of the Hermite model for individual heliostat images requires several enhancements to maintain computational speed and accuracy. Specifically, this tool implements a novel method for dynamic heliostat grouping to reduce the expense of intercept factor evaluation, methods for approximating annual productivity with a subset of time steps throughout the year, a polygon clipping method to accurately calculate inter-heliostat shadowing and blocking, methods for receiver and tower geometry optimization, and an trigonometric image transform algorithm that maintains intercept factor accuracy for small heliostat images. Each of these improvements enable SolarPILOT to perform accurate and efficient computation, and they are discussed in more detail in the following subsections.

2.1.1. Review of the DELSOL3 Analytical Method

DELSOL3 approximates each zone as a single large heliostat in a radial-stagger arrangement, and performance is evaluated at the center point of the zone. Shadowing and blocking are also calculated at this point, assuming a regular distribution of surrounding heliostats also in the radial-stagger arrangement. The code adjusts for "slip planes" (discontinuities in the solar field layout) by reducing the shadowing/blocking loss proportionally to the number of heliostats removed at the slip plane boundary [11]. These assumptions work well for regular, symmetric heliostat fields with a large number of heliostats in a radial-stagger layout. However, current field layout techniques often differ from this historical approach, and alternative solutions offer improved efficiency, reduced land area, exclusions for culturally sensitive areas or topographic features, and accommodation for uneven land as exemplified by [25, 26, 27].

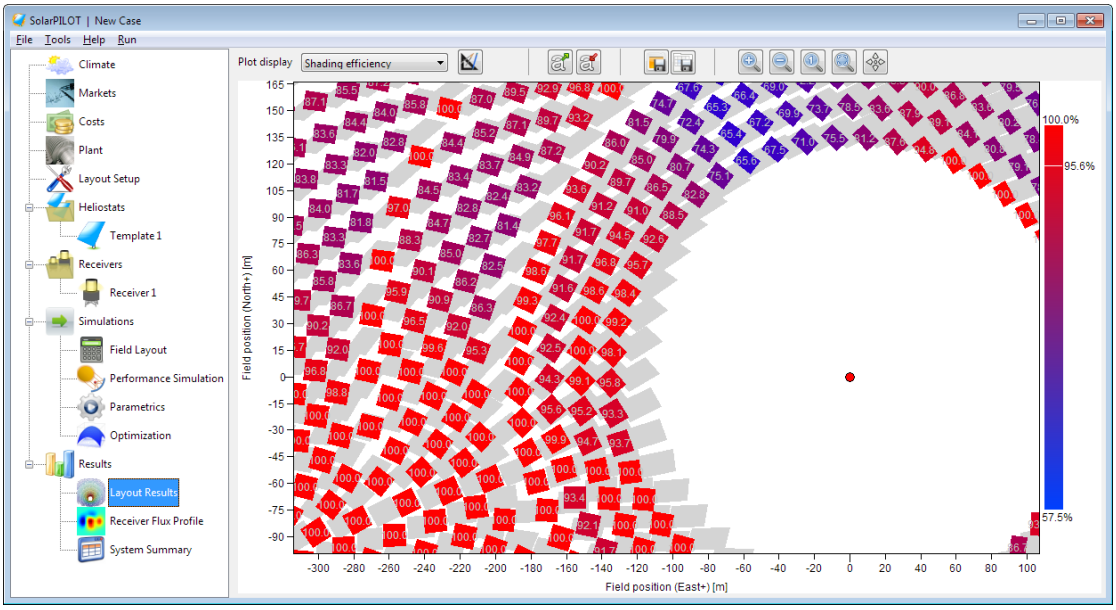


Figure 1: SolarPILOT graphical interface showing a selected region of the heliostat field after an afternoon simulation with shadowing efficiency data overlay.

The computational speed advantage for the Hermite series approximation lies in its application of characterization coefficients that do not all require recalculation for each heliostat or simulation. The analytical form of the flux image as it appears on the receiver plane is shown in Eq. refeq:1 , taken from [5].

$$F(x,y) = \frac{1}{2\alpha_x\alpha_y}\exp\left[-\frac{1}{2}\left(\frac{x}{\alpha_x}\right)^2-\frac{1}{2}\left(\frac{y}{\alpha_y}\right)^2\right] \cdot\left\{\sum_{i=0}^{I}\sum_{j=0}^{J-i}A_{i,j}H_i\left(\frac{x}{\alpha_x}\right)H_j\left(\frac{y}{\alpha_y}\right)\frac{1}{i!\,j!}\right\} \quad (1)$$

The equation assumes that the image can be modeled primarily as two-dimensional normal distribution, but the flux image shape at any position (x, y) is scaled by evaluating a series expansion with Hermite polynomial terms $H_i(x)$, $H_j(y)$, and $A_{i,j}$. The components of the coefficients are evaluated for a given sun shape, mirror geometry, optical error contributions, and relative position to the tower. These individual terms are evaluated only when they change during the simulation, so a flux profile can be quickly developed for heliostats of identical geometry once the initial coefficient analysis is complete. The distribution is normalized by two coefficients – α_x and α_y, which represent the standard deviation of the image distribution in the x and y directions relative to the image plane.

DELSOL3 evaluates the fraction of the heliostat image that is intercepted by the receiver (intercept or spillage efficiency) using a numerical integration method known as Gauss-Hermite quadrature [5]. Each heliostat image is modeled using a 2D Gauss-Hermite polynomial for which no analytical integral exists, but the integral can be approximated by evaluating the flux density expression at various points in the distribution. By carefully choosing the location of these points to coincide with abscissa in the polynomial expression, relatively few points are required to calculate the integral. DELSOL3 uses this algorithm along with a weighting function for each point (that has also been derived analytically) to determine the intercept factor for each image with only 16 points. The integral bounds coincide with the extents of the receiver, and the evaluation points lie in between these extents.

2.1.2. *Shadowing and Blocking*

SolarPILOT calculates shadowing and blocking using a vector projection and clipping method. Neighboring heliostats are tested for potential interference by projecting vectors from the heliostat corners along the direction of either the tower (blocking) or sun position (shadowing). If a projected vector intercepts an adjacent heliostat, blocking or shadowing are enforced according to the position of the interception. This method assumes that neighboring heliostats lie in parallel planes – a good assumption for all but very small solar fields. This assumption results in the simplification that shadowed or blocked regions are rectangular. Overlap of shadowing and blocking is neglected, so the blocking/shadowing efficiency is conservative.

The shadowing and blocking algorithm is as follows:

i. Each heliostat **H** is assigned a list of neighbors (see §2.1.3) that may block or shadow (“interfere” with) the heliostat. An interfering heliostat is subsequently denoted as **K**.

ii. A vector aiming at subject of interference (the receiver for blocking or the sun position for shadowing) is calculated for the interfering heliostat $\hat{\mathbf{i}}_\mathbf{K}$.

iii. Heliostats are tested for the possibility of interference:

   a. The first test requires that the interfering heliostat **K** is within view of the interfered heliostat **H**. The dot product is calculated between the heliostat normal vector and the interfering heliostat subject vector.

$$v = \hat{\mathbf{n}}_\mathbf{H}\cdot\hat{\mathbf{i}}_\mathbf{K} \quad (2)$$

If the dot product is non-positive, **K** cannot interfere

with **H** and the loss is zero.

b. The maximum interference length is calculated for heliostat **H** based on the position in space of each heliostat $\mathbf{P_H}$ and $\mathbf{P_K}$, the heliostat structure height $h_\mathbf{H}$, the interfering heliostat tracking vector $\hat{\mathbf{t}}_\mathbf{K}$, and the zenith angle of the heliostat tracking vector $\phi_{t,\mathbf{K}}$.

$$L_{int} = \frac{\mathbf{P}_{\mathbf{K},k} - \mathbf{P}_{\mathbf{H},z} + h_\mathbf{H} \sin \phi_{t\mathbf{K}}}{\hat{\mathbf{t}}_{\mathbf{K},k} \sqrt{\hat{\mathbf{t}}^2_{\mathbf{K},i} + \hat{\mathbf{t}}^2_{\mathbf{K},j}}} + h_\mathbf{H} \hat{\mathbf{t}}_{\mathbf{K},k} \quad (3)$$

If the distance separating the two heliostats in question is greater than L_{int}, the heliostats cannot interfere and the loss is zero. L_{int} is limited to $100 \times h_\mathbf{H}$ during very low sun elevation angles to limit the number of potential interfering heliostats and the associated computational requirement.

iv. The interference vector $\hat{\mathbf{i}}_\mathbf{K}$ is projected from the two top corners of **K** onto a plane containing **H**. The plane intersection points are tested for containment within **H**.

v. The interfered region area is calculated based on the intersection position of $\hat{\mathbf{i}}_\mathbf{K}$ within **H**. The total interference efficiency is equal to the complement of the interference area divided by the total heliostat area.

2.1.3. Dynamic Heliostat Grouping

Power tower solar fields contain thousands or tens-of-thousands of individual heliostats. Often, performance of neighboring heliostats is very similar, and one heliostat can be used to represent the performance of a small group of neighboring heliostats. The most accurate optical performance calculations consider each heliostat individually (SolarPILOT default behavior), and the least accurate consider all heliostats in a group to have identical performance, including shadowing, blocking, intercept factor, atmospheric attenuation, and cosine losses (DELSOL3 behavior). Accuracy improves as the zone size approaches the domain of a single heliostat.

Accuracy can also improve by considering heliostat loss mechanisms separately and only approximating those losses that are computationally expensive to calculate. For example, cosine loss is one of the most significant losses but its computational expense is trivial, so each heliostat can be considered individually. Optical intercept factor (the amount of light captured by the receiver from any heliostat image) is equally significant but is comparatively much more expensive to calculate. Therefore, we have adopted a mixed approach of calculating simple losses such as cosine, attenuation, etc., individually and the expensive intercept factor loss using a zonal approximation.

These so-called optical zones include groups of neighboring similarly performing heliostats. The challenge with the zonal approach is that the intercept factor can depend strongly on position, and change in intercept factor is nonlinear as a function of radial and circumferential position. Consider Figure 2 showing intercept factor as a function of radial position for a particular representative field geometry. The intercept factor stays constant until the image begins to spill off the receiver near 250m radius. The intercept factor then drops precipitously but eventually stabilizes as distance from the tower increases.

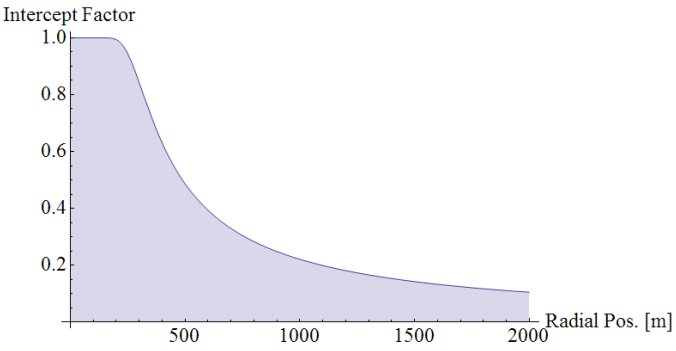


Figure 2: Intercept factor of a 4 × 4m heliostat on a 10 × 10m receiver as a function of radial distance. The heliostat focal length is 100m and total optical error is 5 mrad.

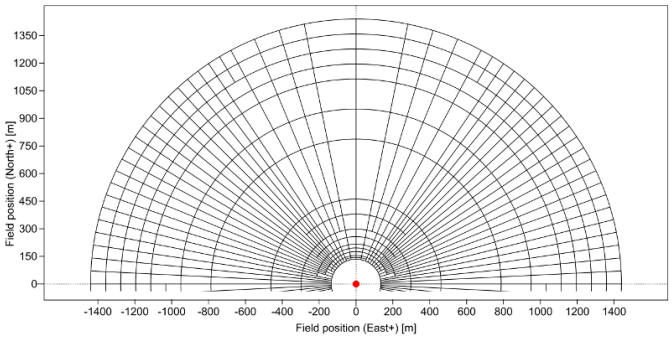


Figure 3: Optical intercept grouping mesh for a north-facing receiver. Each element corresponds to a maximum variation of 5% optical intercept factor.

Both radial and circumferential position can impact intercept factor, especially for receivers with a planar aperture (flat plate or cavity receivers). The rate of change of intercept factor as a function of radius r and azimuthal heliostat position can be determined for planar and cylindrical receivers. The higher the magnitude of these derivatives, the more quickly the intercept factor changes with position, and the smaller the group of heliostats that can be represented by a single calculation.

View factor is the limiting case of intercept factor that is, intercept factor cannot exceed the view factor between a heliostat and a receiver of infinite size. In order to maximize intercept efficiency calculation accuracy, zones should be most densely concentrated in the areas with the highest derivative value in the lower rightmost plot. Conversely, areas that show little change in intercept efficiency can be grouped together in increasingly large number with minimal impact on accuracy.

SolarPILOT implements the concept of variable mesh density and applied it to the derivative of optical intercept. The procedure is as follows:

i. Break a field into a rough set of large zones (mesh elements). Each element may span 45 and 2 tower heights (radially), for example.

ii. Evaluate each element at the centroid location to determine the derivative of intercept factor and view factor with respect to radius and azimuth angle .

iii. If the change in intercept factor across the zone exceeds a user-specified tolerance, split the zone into two sub-zones.

iv. Evaluate sub-zones and split if necessary, *ad infinitum*

The resulting mesh will contain elements appropriately sized such that all heliostats falling within the element share an intercept factor value within the given tolerance. Figure 3 shows an example of an optical mesh for a north-based field. The highest gradient in intercept factor is concentrated near regions of high flux incidence angle. Note that not all zones contain heliostats. Empty zones do not penalize algorithm performance as they are simply not considered during performance calculations.

Mapping heliostats to optical zones. The meshing model discussed above allows definition an efficient and meaningful optical mesh based on local intercept factor derivatives. This mesh is nonregular, and each element may have different radial and azimuthal extents. Therefore, no simple transformation exists between heliostat position in Cartesian coordinates and zone location. In fact, it is computationally nontrivial to determine which zone a heliostat should belong to. In the worst case, one would have to run through the entire list of mesh elements once for each heliostat to define groupings, requiring exponential computational effort.

To get around this problem, we have devised a binary element tree that guarantees that a small number of computations n_{cmax} are required to locate a heliostat within some tolerance δ. The computational expense for the binary element tree is logarithmic, with the worst case being:

$$n_{cmax} \leq 2\left\lceil \log(2)^{-1} \max\left\{\log\left(\frac{\Delta r}{\delta r}\right), \log\left(\frac{\Delta \beta}{\delta \beta}\right)\right\}\right\rceil \qquad (4)$$

In order to demonstrate the computational efficiency of this method, the following example is provided. A typical field will have a radial extent of $\Delta r = r_{max} - r_{min} = (8-1)h_{tower} = 7 \cdot h_{tower}$ where r_{max} and r_{min} equal the minimum and maximum heliostat extents radially normalized by the tower height h_{tower}. A heliostat's radial position can uniquely described with coordinate tolerance of $\delta r = 0.01 \cdot h_{tower}$ since this corresponds to a distance less than the minimum heliostat spacing for a typical field. In the azimuthal direction, the heliostat separation at r_{max} is $\Delta\beta = 2\pi h_{tower} \cdot r_{max}$, and we use $\delta r = \delta\beta$. The number of computations required to locate a point in the field for this example is no more than $n_{cmax} = 20$. The total number to locate each of the N_h heliostats is then proportional to $N_h \cdot 20$, which is a vast improvement over N_h^2 for the worst exponential case.

The binary element tree works by dividing the coordinate space into halves radially and azimuthally. Each half is then divided again and again until the size of the zone is sufficiently small. Divisions are done alternatingly between radial and azimuthal directions. The decision to select the inner/outer (or clockwise/counter-clockwise) half is determined with an integer value. Inner/ccw values are denoted with '0' and outer/cw with '1' – hence the binary structure.

The number of characters the identifying key is determined by the required resolution (size) of the zone. Larger zones require fewer characters to uniquely identify them, while smaller zones require more. This procedure lends itself to mesh structures with variable element size, which is an important feature for SolarPILOT implementation.

Each heliostat is assigned a binary coordinate tag that uniquely identifies its location using the same binary procedure. Any heliostat that falls within a particular zone will begin with the same binary character string as the zone itself, enabling quick association of heliostat-to-zone.

The binary mesh tree is recursive such that each element independently can decide whether to split or remain intact. The algorithm also allows unidirectional splitting if the required tolerance has been met in only one direction. That is, if the mesh reaches sufficient resolution regarding derivatives in the radial direction, additional splits are permitted in the azimuthal direction only while the radial extent remains unchanged. Once the mesh has been defined in the first element after a split, the algorithm calls again to define the mesh in the remaining half. The resulting data structure resembles a tree, with the largest elements branching into many sub-elements which can branch further. A similar but distinct *k-t* binary tree method is described in [28]. Only "terminal" elements that is, elements that contain no sub-elements – are permitted to accept heliostats into their group. This ensures the uniqueness of each element.

2.1.4. Efficient Annual Performance Prediction

SolarPILOT selects the heliostat positions to include in the final layout by estimating the annual performance of each heliostat in the field, then ranking their performance and selecting the most productive heliostats first. Although heliostat field systems are optically complex, their performance over time is reasonably tracked as a linear summation. That is, for approximately similar sun positions (and thus optical efficiencies), the power delivered by the field is the summation of the set of solar resource values times the field area times the average optical efficiency. Because it is often computationally expensive to evaluate field optical performance, it is often impractical to determine annual field performance by simulating all daytime hours in the year. Instead, a subset of hours can be simulated and the annual approximation projected from that sample. SolarPILOT includes several methods and tuning parameters to configure the annual performance estimate simulation set. These are:

Single simulation point. The field performance is evaluated at a single sun position and solar resource. Each heliostat's performance is characterized by a single production value.

Annual simulation. Every daytime hour of the year is simulated, and each heliostat's performance is characterized by the accumulation of all hours in the year. This option – while thorough – is computationally expensive and most often unnecessary.

Limited annual simulation. A subset of days and hours of those days are chosen at regularly spaced intervals throughout the year for simulation. At each hour, corresponding weather data is used to determine heliostat field productivity. Each heliostat is characterized by the set of productivity values generated in the simulation. A sufficiently large number of days must be used to achieve convergence, and the number depends on the seasonal and daily weather variability. Convergence is typically

achieved with 12 simulation days at an hourly or bi-hourly resolution.

Representative profiles. This option mimics the *Limited annual simulation*, but generates averaged weather profiles for the selected days rather than using specific weather days from a weather file. This option demonstrates convergence with 4 simulation days and bi-hourly resolution and is observed to be the most effective option.

Annual efficiency map. SolarPILOT can also generate a lookup table of optical efficiency as a function of solar position, then run an annual simulation drawing from the lookup table rather than generating performance data from the first-principles model.

In addition to climate effects, local markets can shape the temporal value of power production. Some utilities such as Southern California-Edison, San Diego Gas & Electric, and Pacific Gas & Electric provide payments to electricity producers based on time-of-day and day-of-the-year that reflect increased demand during certain time periods. A plant optimized for revenue would consider increasing power production during the most profitable hours of the year at the expense of overall electricity production. SolarPILOT allows specification of temporal revenue weighting factors that are considered during layout.

2.1.5. *Field layout methodology*

SolarPILOT is designed to generate heliostat field layouts with individual heliostat coordinates unlike the zonal approach in DELSOL3. Several layout options are possible, including permutations on the radial-stagger layout and a "corn field" layout. The code is easily extendible to generate alternative layout patterns, and heliostat coordinates can be imported and simulated by the user. The layout procedure is as follows:

i. Generate all possible heliostat positions within the land boundaries.
ii. Place heliostats at the positions according to the heliostat geometry template rules (if applicable).
iii. Simulate the performance of all heliostats at the field design simulation time step(s) specified by the user, using weather data if applicable.
iv. Sort heliostats by performance-to-cost ratio.
v. Simulate solar field performance at the design point solar position and DNI. The single design point may be noon on the summer solstice, noon on the equinox (spring), noon on the winter solstice, solar zenith, or a user-specified sun position.
vi. Select the first N heliostats that generate sufficient power to meet the design-point thermal power requirement.

Land boundaries can be specified point-wise as a set of polygons. Each polygon can represent an "inclusion" area or region of exclusion. Land bounds may also be specified as minimum and maximum radial limits that either scale with tower height or are fixed distances. Heliostat positions are generated within the entire land boundary considering all of the constraint types that are in use. Figure 4 shows a heliostat field built within a non-circular boundary.

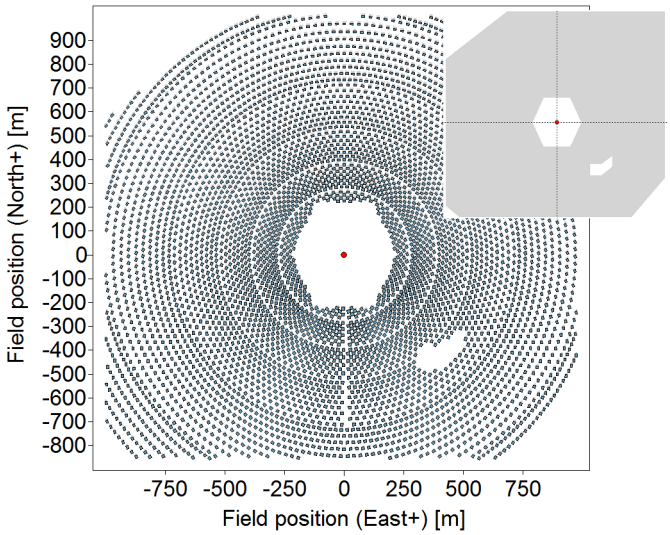


Figure 4: Land-restricted layout with land boundaries overlaid in the top right corner. The available land area is defined by a single inclusion area and two exclusion regions.

2.1.6. *Intercept Factor for Small Images*

One challenge with adapting the Hermite approximation method to individual heliostats is the relative small size of the heliostat image compared to the receiver surface. The Gaussian quadrature method for integrating the flux density equation and determining the intercept factor uses a grid of 16 points to evaluate the density equation. A problem arises when the heliostat image is significantly smaller than the quadrature grid. If the image is much smaller than the apparent receiver size (e.g. 1 m^2 heliostats on a 10×10m receiver), the image can effectively evade detection by the integral algorithm and yield a very low intercept factor when the entire image is – in fact – contained well within the extents of the receiver. This problem can be corrected by scaling the quadrature grid to better match the size of the heliostat image. Unfortunately, exact knowledge of the image size requires a full evaluation of the Gauss-Hermite function over the receiver domain, and executing such a computationally intensive evaluation negates the appeal of the analytical form. However, the image size can be reasonably approximated as a simple Gauss-normal distribution with standard deviation (σ_x, σ_y) describing the elliptical shape of the image. The limits of the integral are then some constant factor times the standard deviation. With a sufficiently high factor chosen (but sufficiently low to avoid the numerical issue described above), the quadrature grid can be scaled to fit within the new limits and the integral can be performed.

This process is straightforward for cylindrical receivers. Each heliostat aimpoint lies at some vertical position along the center-axis of the receiver as it appears to that heliostat. In effect, the heliostat image – while vertically displaced from the receiver centerline – is always centered horizontally on the apparent rectangle of the receiver. The image is elongated in the vertical direction in proportion to the cosine of the angle between the receiver surface normal and the incident image vector (heliostats closer to the tower are distended more so than dis-

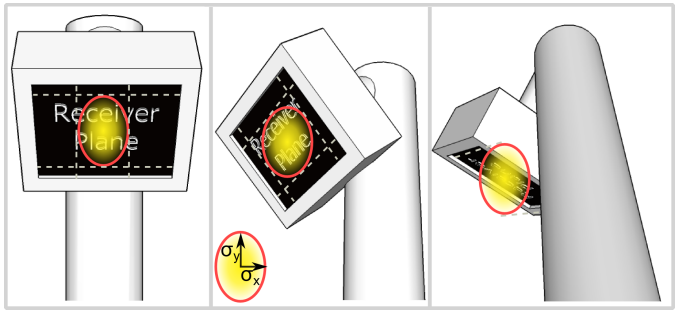


Figure 5: Heliostat image on the receiver plane in three position scenarios. Bounding-box image extents are shown for each case, indicating the dependence of "scaled receiver" size on the heliostat's receiver view.

tant heliostats). Importantly, the heliostat images are not significantly skewed or rotated in the horizontal direction because the incident vectors from each heliostat have no azimuthal component, only vertical displacement. The consequence of these observations is that the quadrature grid can be scaled in a very simple manner:

$$w_{quad} = \min 4\sigma_x, r_{rec}] \quad (5)$$

$$h_{quad} = h_{rec} \quad (6)$$

The quadrature grid width is the minimum of four times the x-direction image size and the receiver radius r_{rec}, while the quadrature height h_{quad} is always left as the receiver height h_{rec}. This approach was implemented in the original DELSOL3 code and has proven to be adequate.

The cavity/flat-plate receiver case is somewhat more difficult. With the view angles between each heliostat and the receiver surface varying significantly over the extent of the heliostat field, the quadrature grid size cannot be scaled according to σ_x, σ_y alone. The issue is illustrated in Figure 5. The first case (left) shows a heliostat image projected onto the receiver plane. The quadrature limit of the image is drawn at $5\sigma_x$, $5\sigma_y$, and dotted tangent lines indicate the revised integration width for the grid. The same image projected from different points in the field result in widely differing quadrature limits as shown in the remaining cases (center, right). Furthermore, the difference between image and receiver coordinate systems is quite pronounced. These complications apparently prevented any practical quadrature scaling method from being implemented in DELSOL3, and the resulting behavior could be severe, as illustrated in Figure 6. Figure 6a shows the resulting "optimal" layout positions for a large flat-plate receiver with small heliostats. The unexpected gap corresponds to images that are small enough and oriented in such a way as to be overlooked by the integration algorithm. The layout shown in Figure 6b uses the corrected scaling algorithm.

The corrected scaling algorithm follows a simple procedure:

i. Given the position and orientation of the heliostat and receiver, the receiver corner points are translated into the coordinate system of the heliostat image plane. The "image plane" is a theoretical plane that is normal to the vector following the reflected heliostat image.
ii. The quadrature grid must be scaled while maintaining its original receiver coordinate system. The slope of the quadrature grid bounds is equal to the slope of the receiver bounds when projected onto the image plane. The slopes are calculated using the translated receiver corner point coordinates in step (i).
iii. The quadrature limit is determined by locating the intersection point between the ellipse of the projected image and a tangent line of slope equal to that calculated in step (ii). (See dotted lines in Figure 5.)
iv. The radius of the ellipse at the two tangent points give the quadrature width and height in the image plane coordinate system.
v. The final quadrature width and height are calculated by translating the width and height from step (iv) back into the receiver plane coordinate system.

The key calculation in this process is the expression relating the ellipse radius to the slope of a tangent line. In particular, we wish to express the radius of the ellipse as a function of tangent line slope. The equation of an ellipse with width and height σ_x, σ_y is shown in Eq. 7:

$$\left(\frac{x}{\sigma_x}\right)^2 + \left(\frac{y}{\sigma_y}\right)^2 = 1 \quad (7)$$

The derivative $^{dx}/_{dy}$ is:

$$\frac{dy}{dx} = \frac{-\sigma_y x}{\sigma_x \sqrt{\sigma_x^2 - x^2}} \quad (8)$$

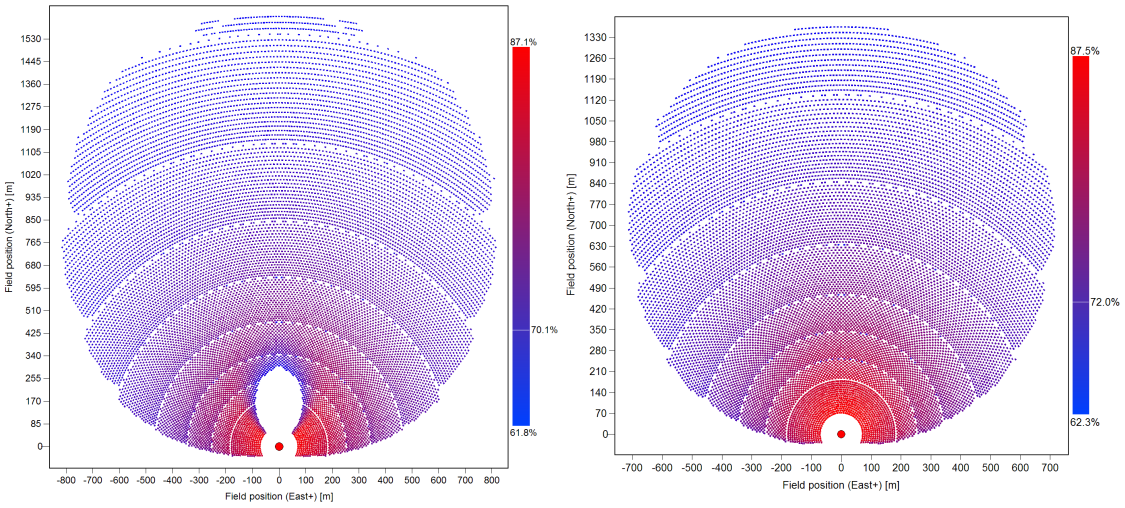
Equation 8 can be solved for x, substituted into the radius equation $r = \sqrt{x^2 + y^2}$, and rearranged to express radius as a function of tangent slope as shown in Eq. 9.

$$r = \sqrt{\frac{\sigma_x^4\sigma_y^2 + (\sigma_x^2 + (\sigma_x^3 - 1)\sigma_y^2)\left(\frac{dy}{dx}\right)^2}{\sigma_x^3\left(\sigma_x\sigma_y + \left(\frac{dy}{dx}\right)^2\right)}} \quad (9)$$

This relationship provides a closed-form calculation of the required quadrature width given receiver corner coordinates projected onto the heliostat image plane.

2.2. *SolTrace integration*

SolTrace is a MCRT code designed for solar applications [17]. Incoming solar radiation can be characterized in any variety of shapes, and the code handles optical error distributions and multiple reflections. SolarPILOT has integrated SolTrace directly through an application programming interface (API) that calls to SolTrace's core tracing functions. The primary strength of SolTrace is characterizing the performance of well-defined geometry, and typical use involves definition of geometry including tracking angles externally or in the built-in scripting language. As SolarPILOT is designed for power tower system layout, it serves as an interface for geometry definition, rapidly generating the required system geometry for SolTrace runs. The combination of analytical and MCRT tools means SolarPILOT can quickly calculate optimized heliostat aim points using analytical characteristics, then generate a detailed MCRT



(a) Layout before correction

(b) Layout after correction

Figure 6: Illustration of the impact of quadrature scaling on the heliostat field layout.

flux profile using SolTrace. This capability is especially useful for cavity-type receivers that analytical methods cannot adequately characterize because of complex view factors and multiple reflections.

While SolTrace offers several key advantages compared to analytical methods such as multiple-reflection characterization and the capability to analyze more complex geometries, it does not provide detailed optical loss information broken down by loss mechanism. The reported results from a SolTrace run consist of the number of intersected rays on all modeled surfaces. Information on what proportion of the optical loss is due to cosine, atmospheric attenuation, spillage, etc, is not directly calculated. Therefore, the best use of MCRT techniques is in conjunction with an analytical approach that provides insight into loss mechanisms.

3. Model Verification

SolarPILOT was developed as an extension to DELSOL3. Therefore, the performance of SolarPILOT is compared to DELSOL3 to verify the correct implementation of the new model. Because SolarPILOT includes several improvements over DELSOL3 in field layout techniques, characterization accuracy, and other features previously discussed, the comparison study matches thermal power delivered at design at the base of the tower (i.e. after reflective, convective & emissive, and piping losses) and inspects the various loss components modeled by each software package. The case study models a large, cylindrical, molten salt receiver with large multi-paneled heliostats. The field layout is radial-stagger, and all input parameters are matched as closely as possible to define analogous cases. Table 1 shows a summary of input parameters for each case.

The single-point simulation comparison shown in Table 1 indicates excellent agreement across the range of modeled physical effects. Performance is also compared at a matrix of sun positions. The discrepancy between total optical efficiency of the solar field (DELSOL3 subtracting SolarPILOT) is shown in Figure 7. These results show good agreement throughout the range of sun positions, but error increases substantially at very low sun elevation angles. The reason for this disagreement is not fully clear, but possible causes include differences in shadowing and blocking calculations or potential inaccuracy in the image shape model as applied to single heliostats at severe reflection angles. Irregardless, the heliostat field typically will not operate at low sun positions because of shadowing and blocking effects, and solar resource is typically very low at times when the sun is near the horizon, so the effect of the inaccuracy is

Table 1: Parameters for the comparison case study and simulation results.

Parameter	**Units**	**DELSOL3**	**SolarPILOT**
Thermal power output	MWt	669.9	
Reference DNI	W/m^2	950	
Heliostat area	m^2	144.4	
Heliostat total refl.	%	89.1	
Tower height	m	203.3	
Receiver height	m	20.41	
Receiver diameter	m	17.67	
Receiver absorptance	%	0.94	
Azimuthal spacing factor	-	n/a	1.96
Slip plane reset limit	-	4/3	1.31
Design weather model	-	Meinel mod.	TMY3 data
Max. field radius	$\cdot h_{tower}$	7.5	9.0
Reference design time	-	Equinox, solar noon	
Compared simulation time	-	Solstice, solar noon	
Result	**Units**	**DELSOL3**	**SolarPILOT**
Number of heliostats	-	8,947	8,945
Power incident on rec.	MWt	767.93	766.87
Power at receiver base	MWt	680.77	679.46
Cosine efficiency	%	80.3	80.6
Blocking efficiency	%	99.3	99.0
Shadowing efficiency	%	100.0	100.0
Atmospheric transmit.	%	91.6	91.2
Heliostat reflection	%	89.1	89.1
Intercept efficiency	%	96.3	96.0
Absorption efficiency	%	94.0	94.0
Thermal efficiency	%	94.65	94.60
Total efficiency	%	58.8	58.8

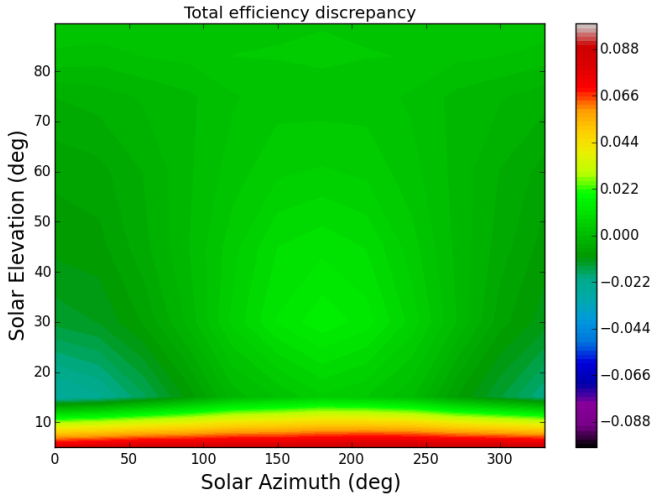


Figure 7: The difference in calculated optical efficiency between DELSOL3 and SolarPILOT as a function of sun position for the case shown in Table 1. The difference shown is DELSOL fractional efficiency minus SolarPILOT fractional efficiency at each evaluated sun position.

minimal on annual energy production.

4. Conclusions

A new model for calculating solar field layouts and performance characteristics for power tower systems is developed and described. The tool employs both an analytical Hermite polynomial expansion flux mapping technique and MCRT with SolTrace. SolarPILOT is intended for use as a third-party validation tool for existing private industry models, a research and screening tool, and a platform for development of new modeling or design techniques. The tool models a variety of solar field, heliostat, receiver geometries, and optical scenarios. One primary strength of SolarPILOT is its extension of the analytical flux technique from DELSOL3 onto individual heliostats. This allows a more flexible design process, including possible asymmetry, topography variation, and geometry variation within the heliostat field.

SolarPILOT has been integrated into NREL's SAM software as the power tower design and characterization engine. This integration is via an API, and future work on the software will involve improvement to the API for general use.

Nomenclature

API Application Programming Interface
MCRT Monte-Carlo Ray-Tracing
SAM System Advisor Model
SolarPILOT Solar Power tower Integrated Layout and Optimization Tool

Acknowledgements

Funding for this project was provided to NREL by the U.S. Department of Energy under Contract No. DE-AC36-08GO28308. The authors gratefully acknowledge our colleagues Greg Kolb (ret.) and Cliff Ho at Sandia National Laboratory in Albuquerque, New Mexico, for their contributions to DELSOL3 and guidance on the methodologies. We also thank Professors Rob Braun and Alexandra Newman in the Department of Mechanical Engineering at Colorado School of Mines for technical review of the material in this paper.

References

[1] A. B. Zavoico, Solar Power Tower Design Basis Document, Tech. Rep. July, Sandia National Laboratory, Albuquerque, NM, 2001.
[2] H. Reilly, G. J. Kolb, An Evaluation of Molten-Salt Power Towers Including Results of the Solar Two Project, Tech. Rep., Sandia National Laboratory, Albuquerque, NM, 2001.
[3] J. E. Pacheco, R. Bradshaw, D. Dawson, W. De la Rosa, R. Gilbert, S. Goods, M. Hale, P. Jacobs, S. A. Jones, G. J. Kolb, M. Prarie, H. Reilly, S. Showalter, L. L. Vant-Hull, Final Test and Evaluation Results from the Solar Two Project, Tech. Rep. January, Sandia National Laboratory, Albuquerque, NM, 2002.
[4] M. J. Wagner, Simulation and predictive performance modeling of utility-scale central receiver system power plants, Masters thesis, University of Wisconsin - Madison, 2008.
[5] T. Dellin, An improved Hermite expansion calculation of the flux distribution from heliostats, Tech. Rep., Sandia National Laboratory, Livermore, CA, 1979.
[6] P. Leary, J. Hankins, User's guide for MIRVAL: a computer code for comparing designs of heliostat-receiver optics for central receiver solar power plants, Tech. Rep., Sandia National Laboratories, Livermore, CA, 1979.
[7] M. Walzel, F. Lipps, Vant-Hull, A solar flux density calculation for a solar tower concentrator using a two-dimensional Hermite function expansion, Solar Energy 19 (1977) 239–256.
[8] F. Lipps, L. L. Vant-Hull, A Cellwise method for the optimization of large central receiver systems, Solar Energy 20 (1978) 505–516.
[9] C. K. Ho, Software and Codes for Analysis of Concentrating Solar Power Technologies, Tech. Rep. December, Sandia National Laboratory, Albuquerque, NM, 2008.
[10] P. Garcia, A. Ferriere, J.-J. Bezian, Codes for solar flux calculation dedicated to central receiver system applications: A comparative review, Solar Energy 82 (3) (2008) 189–197.
[11] B. Kistler, A user's manual for DELSOL3: A computer code for calculating the optical performance and optimal system design for solar thermal central receiver plants, Tech. Rep., Sandia National Laboratory, Albuquerque, NM, 1986.
[12] M. Izygon, P. Armstrong, C. Nilsson, N. Vu, TieSOL A GPU-Based Suite of Software for Central Receiver Solar Power Plants, in: Proceedings of the 2011 SolarPACES International Symposium, Granada, Spain, 2011.
[13] M. Kiera, Beschreibung und Handhabung des Programmsystems HFLCAL, Tech. Rep., Interatom Report, 1986.
[14] C. Vittitoe, F. Biggs, A User's Guide to Helios: A computer program for modeling the optical behavior of reflecting solar concentrators, Tech. Rep., Sandia National Laboratory, Livermore, CA, 1981.
[15] B. Belhomme, R. Pitz-Paal, P. Schwarzbozl, S. Ulmer, A New Fast Ray Tracing Tool for High-Precision Simulation of Heliostat Fields, Journal of Solar Energy Engineering 131 (3) (2009) 031002.
[16] M. J. Blanco, J. M. Amieva, A. Mancillas, The Tonatiuh Software Development Project: An Open Source Approach to the Simulation of Solar Concentrating Systems, in: ASME 2005 International Mechanical Engineering Congress and Exposition, Orlando, FL, 2005.
[17] T. Wendelin, SolTRACE: A New Optical Modeling Tool for Concentrating Solar Optics, in: ASME 2003 International Solar Energy Conference, Kohala Coast, HI, 2003.
[18] M. C. Stoddard, S. E. Faas, C. J. Chiang, J. A. Dirks, SOLERGY - A Computer Code for Calculating the Annual Energy from Central Receiver Power Plants, Tech. Rep. May, Sandia National Laboratory, 1987.
[19] National Renewable Energy Laboratory, System Advisor Model, 2012.
[20] European Commission, SOLGATE - Solar hybrid gas turbine electric power system, Tech. Rep., European Commission, Luxembourg, 2005.

[21] L. Feierabend, Thermal Model Development and Simulation of Cavity-type Solar Central Receiver Systems, Masters thesis, University of Wisconsin - Madison, 2010.

[22] S. H. Teichel, Modeling and Calculation of Heat Transfer Relationships for Concentrated Solar Power Receivers, Masters thesis, University of Wisconsin - Madison, 2011.

[23] S. Jones, D. Neal, J. Gruetzner, R. Houser, VSHOT: a tool for characterizing large, imprecise reflectors, in: Conference on space processing of materials, 1996.

[24] M. J. Wagner, S. Klein, D. Reindl, Simulation of Utility-Scale Central Receiver System Power Plants, in: Proceedings of the 2009 ASME 3rd International Conference on Energy Sustainability, ASME, San Francisco, CA, 2009.

[25] M. Sanchez, M. Romero, Methodology for generation of heliostat field layout in central receiver systems based on yearly normalized energy surfaces, Solar Energy 80 (7) (2006) 861–874.

[26] X. Wei, Z. Lu, Z. Wang, W. Yu, H. Zhang, Z. Yao, A new method for the design of the heliostat field layout for solar tower power plant, Renewable Energy 35 (9) (2010) 1970–1975.

[27] C. J. Noone, M. Torrilhon, A. Mitsos, Heliostat field optimization: A new computationally efficient model and biomimetic layout, Solar Energy 86 (2) (2012) 792–803, other concepts to look into: * Fibonacci sequence / golden ratio.

[28] W. H. Press, Numerical recipes 3rd edition: The art of scientific computing, Cambridge university press, 2007.

[29] M. Sengupta, M. J. Wagner, Impact of Aerosols on Atmospheric Attenuation Loss in Central Receiver Systems, in: Proceedings of the 2011 SolarPACES International Symposium, Granada, Spain, 2011.

Concentrating Solar Power Energy Dispatch Optimization in a Detailed Annual Simulation Model

Michael J. Wagner[a,b,*], Alexandra Newman[c,d], Robert Braun[c,d]

[a]*Mechanical Engineer*
[b]*National Renewable Energy Laboratory, Thermal Systems Group, 15013 Denver West Parkway, Golden, CO 80401*
[c]*Professor of Mechanical Engineering*
[d]*Colorado School of Mines, Department of Mechanical Engineering, 1500 Illinois Street, Golden, CO 80401*

Abstract

A model for thermal energy storage dispatch optimization is formulated and implemented in the National Renewable Energy Laboratory's *System Advisor Model* (SAM). The dispatch optimization model considers future expected performance on a daily basis and develops an operating strategy to maximize plant revenue over the next-day time horizon. While a simplified energy production forecast model is used to pose the optimization problem, the resulting optimized operating strategy is executed by the detailed techno-economic simulation code for which SAM has been historically used. This paper discusses the various sub-models used in this process, provides the optimization model formulation and solution techniques, and presents comparative results for a system with and without dispatch optimization. The results indicate that the efficacy of dispatch optimization is significantly beneficial, but varies with plant capacity factor and electricity markets. Several aspects of plant design and operation that are external to this study will also likely benefit from dispatch optimization (for example, a reduction in the number of turbine start-stop cycles over time).

Keywords: Dispatch Optimization, Concentrating Solar Power, CSP, Energy Storage, System Control, Simulation, System Advisor Model, SAM, Mixed-Integer Linear Programming, MILP

1. Background

Concentrating Solar Power (CSP) systems are often capable of efficiently storing thermal energy generated during daylight hours for use in generating electricity when sunlight is not available. Thermal energy storage (TES) technologies can vary [1], but a mature TES technology utilizing high-temperature molten salt as the storage media has been successfully implemented in CSP tower systems [2, 3] and in parabolic trough systems [4] in an "indirect" manner through use of an intermediate oil-to-molten salt heat exchanger. The sizing of the TES system is determined during the design process, and the optimal TES size is a function of the desired plant capacity factor, the sizing of the solar field and power cycle subsystems, plant location, and project economics, among other factors. Thermal energy storage sizing can also depend on the intended operational scheme of the plant. For example, a plant that intends to operate primarily during high-revenue morning or afternoon periods but reduce production during daylight hours will require more TES capacity than a plant with an identical capacity factor that will generate power during daylight hours. As CSP plants target dispatch during high-revenue periods rather than focus on minimizing the average cost of energy, dispatch optimization is of increased importance. The intelligent dispatch of stored energy can greatly enhance the value of electricity by providing firm capacity, ancillary services, and by generating electricity during time periods where electricity rates are especially high [5].

1.1. Related Work

The primary challenge of developing a dispatch optimization model for CSP technologies is the conflicting tradeoff between performance simulation accuracy and tractability of the Mixed Integer (Linear) Program (MIP). Performance simulation software for CSP is complex and relies on non-linear engineering relationships to accurately predict plant performance. Models at the component and subsystem level derive expected performance from first-principles and empirical observation of the heat transfer, thermodynamic, and mechanical behaviors of the system. Although several software packages exist that are capable of executing simulation of CSP systems [6, 7], the mathematical structure of the simulation is often inexpressible as a system of explicit, simultaneous equations and is rather more of a mathematical "black box". This feature makes the optimization process more difficult and interdicts the possibility of knowing whether any particular configuration is truly *the* optimal solution. Black-box simulations are even more difficult to optimize since the expense of generating solution space derivatives is non-trivial ("derivative-free" problems). On the other hand, systems of equations that are exclusively comprised of linear and integer terms (that is, each term has a single variable

*Corresponding author
Email addresses: michael.wagner@nrel.gov (Michael J. Wagner), anewman@mines.edu (Alexandra Newman), rbraun@mines.edu (Robert Braun)
URL: www.nrel.gov/csp (Michael J. Wagner)

that is either continuous, integral, or binary) have special mathematical characteristics that allow more efficient determination of the optimality of a particular solution. A broad and effective class of solvers and modeling languages exist for MIP's (e.g. [8, 9]) that make problems with many thousands of variables and constraints tractable.

It is sometimes possible to approximately represent nonlinear or derivative-free problems using a MIP formulation. However, this typically requires the introduction of a large number of additional variables and equations (or constraints) relating those variables into the formulation. As the problem grows in complexity and dimensionality, the amount of time required to identify the optimal solution also grows. For a CSP dispatch optimization problem, each hourly time step throughout the year has a number of independent variables associated with solar field generation, power cycle generation, the energy state of components in the system, operational modes, and others. Each variable is also independent in time. The variables are related by constraints describing sequence of operations, limitations on the value of a variable given the value of another variable, and so on. Each constraint relating one hour to another is separate such that the total number of constraints is proportional to the total number of variables in the problem. It is clear, then, that an annual hourly simulation treated within a single formulation is a tremendously large problem that may be intractable for many MIP solvers.

This difficulty can be assuaged by reducing the scope of the problem or by employing heuristic models. The previous approach implemented in NREL's System Advisor Model (SAM) [10] uses a simple heuristic that allows the user to specify requirements that must be met before thermal storage can be dispatched. These conditions do not consider expected resource availability or market pricing, but determine the operational state of the power cycle based on the current charge state of TES and the hour of the day. The heuristic approach can improve plant production during high-value hours as exemplified by SAM or Guédez et al. [11], but can ultimately decrease the utilization of the solar field throughout the year because of TES over-charge situations.

A natural approach for simplifying the problem while still implementing MIP techniques is outlined by Madaeni et al. [12]. The authors use SAM to generate an hourly thermal power production profile throughout the year. The profile is considered as fixed input to the MIP model originally outlined in [13] and factors in the simulated performance of the solar field, but omits interactions with thermal storage and the power cycle. The latter subsystems are modeled as part of the MIP formulation such that the charge state of TES and electricity production from the cycle are calculated outputs from the optimization model. This approach improves tractability by utilizing a detailed model to generate fixed input while utilizing a simplified energy balance model to characterize TES charge and power cycle generation. Furthermore, Madaeni et al. employ a rolling time horizon for optimization. A 48-hour time horizon is optimized every 24 hours, allowing day-to-day holdover while revisiting the optimal schedule on a daily basis. The approach used in the current work follows the format presented in [12] to a large degree. Importantly, the current work uses the optimized schedule only to control the detailed simulation, whereas the Madaeni et al. work uses the results from the MIP as the actual estimate of plant production throughout the year.

1.2. Goals of the current work

Dispatch optimization is of interest for improving the profitability of existing or planned CSP facilities, but it is also of great interest to policymakers and researchers who seek to better understand the projected performance of CSP systems under various deployment and grid operations scenarios. This is well established by [5, 12, 13] and others, but their work considers the dispatchability of CSP systems from the perspective of grid integration. In this paradigm, CSP systems are designed at an energy-flow and system sizing level to best understand the suitability of CSP for meeting grid and market demands. The current work undertakes dispatch optimization from a *technology design and operations* perspective, seeking to understand the impact of using CSP as a dispatchable generator on subsystem and component performance. Accordingly, this work fills the gap between *prescriptive* grid-level models on the one hand that indicate desired technology performance subject to high-level operational requirements (e.g. plant start up, maximum energy generation, etc.) and *descriptive* performance simulations on the other hand whose primary concern is to synthesize expected plant productivity and financial return over time given specific component or subsystem thermo-mechanical performance models.

Estimates of CSP technology performance are ascertained using complex engineering and financial simulation software such as SAM. NREL developed SAM to simulate the techno-economic performance of renewable technologies including CSP, wind, geothermal, photovoltaic, biomass, solar hot water, and generic systems[1]. Each technology can be paired with a financial model to evaluate the financial performance of a project within particular market, incentive, and cost environments. SAM estimates CSP plant productivity and financial return by simulating the detailed behavior of plant subsystems and components over time – accounting for weather variability, operational states, the transient energy state of the system, and plant operations requirements. A single annual "macro simulation" consists of thousands of sequential "micro-simulations" within a time series, and the plant behavior at any given time step may depend on the state of the system in the previous time step(s). A Molten Salt Power Tower (MSPT) model is included in SAM with the configuration illustrated in Figure 1.

In the current work, we have developed a new MIP formulation that is implemented within the detailed SAM annual-hourly performance simulation model. The MIP solves a simplified, approximated problem that considers the dispatch of thermal storage and operation of the solar field and power cycle subsystems on an hourly basis over a forecast horizon. By default, the optimization horizon is 48 hours (this can be changed by the

[1] SAM is free to download and use, and the tools developed in the current work are freely available at the SAM website [14]

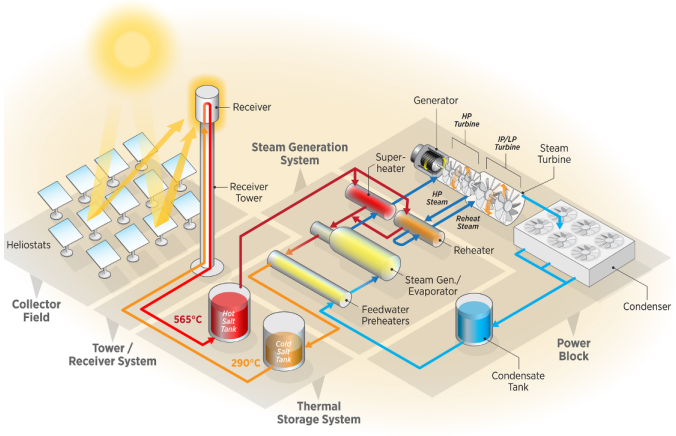


Figure 1: Molten Salt Power Tower system configuration that is modeled in NREL's SAM. The system consists of a heliostat field, molten salt receiver, direct TES system, steam generation system, Rankine power cycle, and heat rejection system.

user), and the dispatch schedule is re-optimized every 24 hours. The additional 24 hours in the optimization window provides sufficient information to allow energy to carry over in storage from day to day. The exact implementation of the model within the simulation environment is described in detail in Subsection 2.1 below.

Direct implementation of the MIP within the simulation model has several advantages. Most importantly, our approach uses the simplified dispatch optimization problem to guide the simulation plant controller over the course of the optimized day. This guidance streamlines the number of suitable operational modes that the controller can choose from. Ultimately, the performance output from the model is the result of the detailed performance calculation engine rather than output from the approximate MIP model.

This approach allows investigation of detailed plant performance issues that may arise from optimized scheduling that are too complex to be easily represented in a simplified MIP model. For example, the thermal stress associated with frequent thermal cycling of power generation equipment may lead to an increase in the frequency of required maintenance [15]. A detailed model can capture these thermo-mechanical impacts when the plant control is influenced by optimized dispatch scheduling.

2. Modeling Approach

In this work, we establish the dispatch schedule that maximizes revenue from electricity sales on a daily basis with an optimization time horizon of 48 hours (mirroring [12]). The time horizon duration selected for this analysis was determined based on computation time required to identify the optimal dispatch schedule and the diminishing returns of an extended window. The time series model is hourly due largely to the widespread availability of hourly data and due to SAM's default approach of modeling systems at an hourly resolution.

The plant dispatch schedule is primarily concerned with determining when and to what extent the power cycle (turbine, generator, condenser, and associated equipment) will operate. During operation, the power cycle consumes stored thermal energy from the TES system. The solar field generates thermal energy that is subsequently stored in the TES system, and energy generation is affected by the optical and thermal efficiency of the solar field, by the intensity of the available solar resource, and by the operational state of the solar field.

Before the power cycle can produce electricity, start-up requirements must be satisfied, including both a minimum start-up period duration and a minimum energy state requirement. In the latter case, the power cycle equipment cools during shutdown periods and must overcome the system's thermal inertia to begin generating steam that powers the turbine. In the case of the start-up duration requirement, power cycle equipment is sensitive to the rate of heating and can't move from a cool shutdown state to a heated operational state too quickly. Equipment manufacturers require a minimum start-up duration to avoid thermal stress and mechanical failure risks otherwise present during rapid temperature ramp-up. Both the energy and duration start-up requirements must be met before the cycle can begin producing power. This is implemented as a constraint on the maximum energy delivered for start-up during any given time period. Although the duration of start-up must take at least a minimum number of time steps, longer start-up durations are allowed in practice based on energy availability, and the model must provide this flexibility.

Two start-up scenarios are possible for the power cycle. Firstly, "cold" start-up occurs when the power cycle has shut down for any period of time and seeks to restart. Secondly, "hot" start-up occurs when the power cycle has been in *standby* mode and seeks to restart. Cold restart requires an additional energy contribution, whereas hot restart can happen immediately (from the perspective of the hourly model).

Standby is a mode of operation where a small (but non-trivial) amount of thermal energy is consumed during each time period to maintain the power cycle equipment in a hot state, ready to quickly ramp up for electricity generation. However, no electricity is produced in standby mode. Consequently, maintaining the power cycle in standby mode is of value if multiple start-up cycles are anticipated over a relatively short time span, or if the energy or time penalty for start-up is sufficiently high to justify the small rate of energy consumption by the power cycle.

The receiver must likewise proceed through a start-up period before producing useful thermal energy. Receiver operation is less complicated than the power cycle in that energy production is virtually always desirable and always coincides with the availability of the solar resource. The receiver must satisfy both minimum energy state and minimum start-up duration requirements, but receiver standby operation is not typical in practice nor modeled in this context. Receiver and power cycle start-up sequences are not necessarily coordinated, so both systems may operate independently with shared interest only in the energy state of the TES system. In some cases, the receiver must curtail energy generation to avoid over-charging thermal stor-

age (thus wasting solar energy), and the power cycle can only generate power when the charge state of TES is non-zero.

2.1. Model Implementation

The implementation of the dispatch optimization model within SAM is illustrated in Figure 2. The SAM interface provides both input pages and output display. The user selects the technology and financial model, then modifies the input pages to emulate their technology configuration of interest. Upon completing the technology and simulation setup process, SAM simulates the technology and financial performance. To accomplish this task, information is sent from the interface to the SAM Simulation Core (SSC) where data matrices and input/output arrays are managed. Next, the MSPT technology model is invoked. This technology module contains detailed performance calculators for determining weather data and the performance of the collector and receiver, power block, and TES systems. The MSPT module also configures the data connections between these interdependent subsystem calculators. A *CSP controller* determines the best operational mode given the current state of the entire system and ambient/solar conditions. The *CSP solver* ensures that all of the interconnected inputs and outputs among the calculators agree with respect to the thermodynamic state of the system.

Figure 2 delineates recent contributions to the model. Namely, a production forecast model, a MIP formulation, and implementation of the formulation within a MIP solver are now part of the simulation model. As SAM executes the MSPT simulation, the production forecast model, MIP, and solver are periodically called to determine the optimal dispatch schedule for the upcoming 24 hour period. Each of these models are discussed in more detail.

Forecast Model. The performance projection model uses simplified, light-weight algorithms to "look ahead" in the simulation and forecast the thermal energy production during the upcoming optimization time horizon. This projection is based on expected weather data, approximate thermal and optical efficiency during the projection period, and an assumption of normal plant operation. The simplified models use performance curves that are generated from a sensitivity analysis using the detailed performance models when the simulation is first executed.

Engineering Performance Model. The engineering performance model (the SAM MSPT controller, solver, and detailed calculators) evaluates plant behavior and productivity over time using computationally expensive algorithms. This model makes operational decisions using the optimized dispatch profile generated by the Forecast Model. Performance can deviate between the Forecast Model and the engineering performance model, so the Forecast Model profile serves as an operational target only.

MIP Mathematical Formulation. The mixed integer problem model represents the performance and operation of the plant in the upcoming time horizon using the Forecast Model and various operational constraints represented in a form convenient for optimization.

3. Mathematical Formulation

3.1. Parameters and Sets

The following MIP takes as fixed parameters the Forecast Model output and parameters relating to the initial operational state of the system at the beginning of each optimization time horizon. These input parameters include the collector and receiver energy generation profile, the expected cycle conversion efficiency profile as a function of ambient temperature only, and the energy price or tariff profile. Initialization parameters are used to set variable values at $t = 0$. A number of additional parameters define operational limits, start-up requirements, and problem formulation parameters. These items are presented in Table 1.

Table 1: Parameters and sets used in the model.

Symb.	Units	Description
		Sets
$\mathcal{T}$		Set of all time steps in the optimization time horizon $T = \lvert\mathcal{T}\rvert$
		Vector Parameters
Q_t^{in}	kWt	Energy generated by the solar field, $t \in \mathcal{T}$
P_t	¢/kWt-hr	Electricity sales price, $t \in \mathcal{T}$
η_t^{amb}	-	Cycle efficiency adjustment factor, $t \in \mathcal{T}$
		Scalar Parameters
τ	hr	Frequency of optimization problem execution
T		Cardinality of time step set $\lvert\mathcal{T}\rvert$
E^u	kWt-hr	Max energy storage quantity
E^r	kWt-hr	Required energy consumed to start receiver
E^c	kWt-hr	Required energy consumed to start cycle
Q^u	kWt	Max thermal power input to cycle
Q^l	kWt	Min operational thermal power input to cycle
Q^{ru}	kWt	Max power used per period for receiver start-up
Q^{rl}	kWt	Min operational power delivered by receiver
Q^c	kWt	Max power used per period for cycle start-up
Q^b	kWt	Standby thermal power consumption per period
L^r	kWe/kWt	Receiver pumping power per unit power produced
Δ	hr	Time step duration
$\mathbb{M}$		A sufficiently large number
		Variable initialization parameters
s_0	kWt-hr	Initial thermal energy storage charge
u_0^{ru}	kWt-hr	Initial receiver start up charge state
u_0^{csu}	kWt-hr	Initial power cycle start up charge state
y_0	-	Initial power cycle operational state
y_0^{csb}	-	Initial power cycle standby state

3.2. Variables

The MIP variables include both continuous and binary items. The variables describe energy (thermal $kW_t - hr$ or electric $kW_e - hr$) states and power flows (thermal kW_t or electric kW_e) in the system. Continuous variables "x" and "u" describe power and energy information relating to the receiver, power cycle, and TES. Charge state and power flow terms are captured in the model. Binary variables "y" are used to enforce operational modes and sequencing such that start-up must occur before normal operation, for example. All variables are subscripted with time t, indicating that each variable may assume a different value with time. Variables are presented in Table 2.

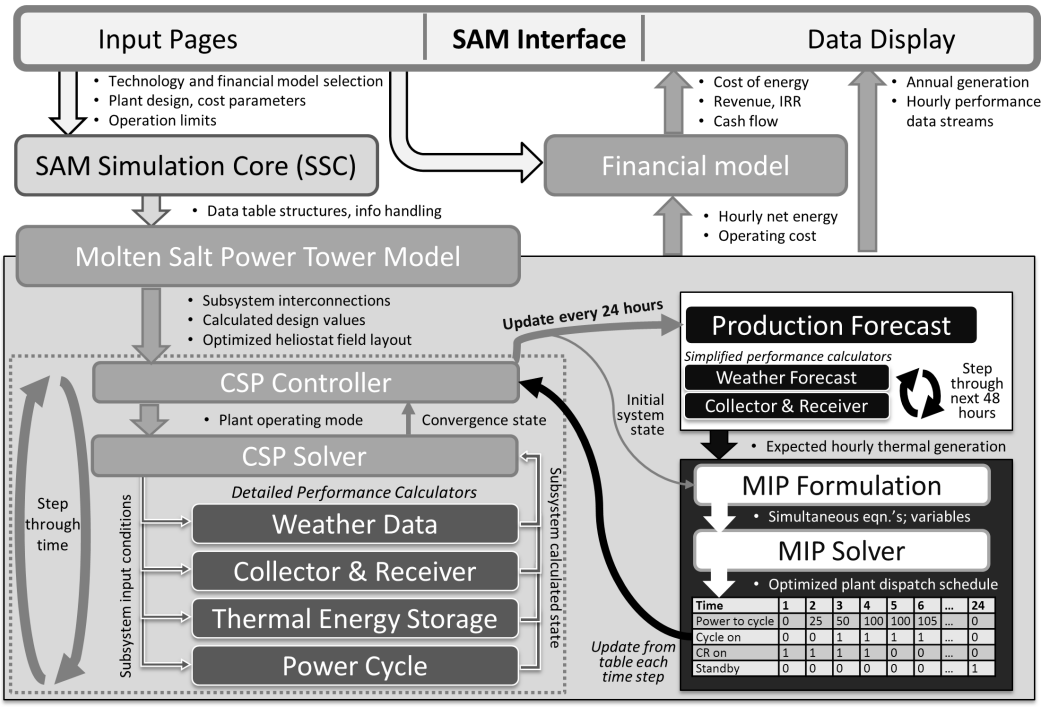


Figure 2: Information flow in the SAM MSPT model. Current work focuses on the Production Forecast, MIP Formulation, and MIP solver.

Table 2: Variables used in the model.

Symb.	Units	Description
		Continuous
x_t	kWt	Cycle thermal power consumption at time t
x_t^r	kWt	Power delivered by the receiver at time t
x_t^{rsu}	kWt	Receiver start-up energy consumption at time t
u_t^{rsu}	kWt-hr	Receiver start-up energy inventory at time t
u_t^{csu}	kWt-hr	Cycle start-up energy inventory at time t
s_t	kWt-hr	TES reserve quantity at time t (auxiliary variable)
		Binary
y_t^r		1 if receiver is generating "usable" thermal power at time t; 0 otherwise
y_t^{rsu}		1 if receiver is starting up at time t; 0 otherwise
y_t^{csu}		1 if cycle is starting up at time t; 0 otherwise
y_t^{csb}		1 if cycle is in standby mode at time t; 0 otherwise
y_t		1 if cycle is generating electric power at time t; 0 otherwise

3.3. Objective Function

The objective function quantifies electricity sales during the optimization time horizon. The objective is to maximize the product of productive cycle energy consumption, the price of energy, and the efficiency at which energy can be converted during each time period. Parasitic losses associated with operation of the receiver HTF pump are subtracted from energy generation from the cycle.

$$\text{Maximize} \sum_{t\in\mathcal{T}} P_t(\eta_t^{amb} x_t - L^r(x_t^r + x_t^{rsu})) \tag{1}$$

3.4. Constraints

The relationships among the variables and between variables and parameters are established with a set of simultaneous constraining equations. The constraints are each presented below topically with a brief description.

3.4.1. Receiver start-up

Receiver start-up constraints include:

$$u_t^{rsu} \le u_{t-1}^{rsu} + x_t^{rsu} \quad \forall t \in \mathcal{T} \tag{2}$$
$$x_t^{rsu} \le Q^{ru} y_t^{rsu} \quad \forall t \in \mathcal{T} \tag{3}$$
$$y_t^{rsu} \le \mathbb{M} Q_t^{in} \quad \forall t \in \mathcal{T} \tag{4}$$
$$u_t^{rsu} \le E^r y_t^{rsu} \quad \forall t \in \mathcal{T} \tag{5}$$
$$y_t^r \le \frac{u_t^{rsu}}{E^r} + y_{t-1}^r \quad \forall t \in \mathcal{T} \tag{6}$$
$$x_t^r + x_t^{rsu} \le Q_t^{in} \quad \forall t \in \mathcal{T} \tag{7}$$
$$x_t^r \le Q_t^{in} y_t^r \quad \forall t \in \mathcal{T} \tag{8}$$
$$x_t^r \ge Q^{rl} y_t^r \quad \forall t \in \mathcal{T} \tag{9}$$
$$y_t^{rsu} + y_{t-1}^r \le 1 \quad \forall t \ge 2 \in \mathcal{T} \tag{10}$$
$$y_t^r \le \mathbb{M} Q_t^{in} \quad \forall t \in \mathcal{T} \tag{11}$$

The receiver start-up and energy generation process is controlled with two binary indexed variables. These variables enable thermal power production (y_t^r) and thermal power consumption for receiver start-up (y_t^{rsu}). When the receiver is in power producing mode, the quantity of useful thermal energy generated is tracked by the continuous variables x_t^r. The rate of thermal power consumption for receiver start-up is given by a parameter Q^{ru}, and is accounted for when receiver start-up is active ($y_t^{rsu} = 1$).

First we consider receiver start-up inventory and the criteria that must be satisfied in order for the receiver to produce useful power. Constraint 2 tracks start-up energy "inventory" from time step to time step. An inequality is used rather than an equality to simplify the constraint set. Inventory is naturally maximized by the problem. Constraint 3 ensures that the actual

energy used for receiver start-up is less than the ramp rate limit. Constraint 4 prevents receiver start-up from occur in time periods with trivial solar resource. Inventory can only be nonzero in time steps where the receiver is starting up by constraint 5. Constraint 6 ensures receiver production operation is allowed only when start-up has been completed or when the receiver was previously operating. The total power produced by the receiver must account for both the available energy Q_t^{in} and any start-up energy consumption as required by constraint 7. The receiver can only generate thermal power when the receiver is in power producing mode by constraint 8. The receiver energy generation must satisfy a minimum threshold. Constraint 9 is enforced because of minimum pump turn-down ratios and heat transfer requirements in the receiver. Constraint 10 ensures that the receiver start-up mode does not persist while the receiver is operating in power producing mode by disallowing start-up in the time step following normal power production operation. Constraint 11 ensures that the receiver power production mode does not persist when no energy is available to the receiver.

3.4.2. Power Cycle start-up

Power cycle start-up largely mirrors receiver start-up and is recapitulated briefly below. Power cycle start-up constraints include:

$$u_t^{csu} \leq u_{t-1}^{csu} + Q^c y_t^{csu} \quad \forall t \in \mathcal{T} \tag{12}$$
$$u_t^{csu} \leq \mathbb{M} y_t^{csu} \quad \forall t \in \mathcal{T} \tag{13}$$
$$y_t \leq \frac{u_t^{csu}}{E^c} + y_{t-1} + y_{t-1}^{csb} \quad \forall t \in \mathcal{T} \tag{14}$$
$$x_t + Q^c y_t^{csu} \leq Q^u \quad \forall t \in \mathcal{T} \tag{15}$$
$$x_t \leq Q^u y_t \quad \forall t \in \mathcal{T} \tag{16}$$
$$x_t + Q^c y_t^{csu} \geq Q^l y^t \quad \forall t \in \mathcal{T} \tag{17}$$
$$y_t^{csu} + y_{t-1} \leq 1 \quad \forall t \geq 2 \in \mathcal{T} \tag{18}$$
$$y_t^{csb} \leq y_{t-1} + y_t^{csb} \quad \forall t \in \mathcal{T} \tag{19}$$
$$y_t^{csu} + y_t^{csb} \leq 1 \quad \forall t \in \mathcal{T} \tag{20}$$
$$y_t + y_t^{csb} \leq 1 \quad \forall t \in \mathcal{T} \tag{21}$$

Several complications are introduced in the cycle operation and start-up process as a result of "standby" operation mode. Standby mode consumes energy without producing electric power in order to maintain the power cycle at a high energy state. No start-up penalty is enforced when beginning normal operation from standby mode.

Constraint 12 tracks start-up energy inventory, and constraint 13 ensures nonzero inventory. Constraint 14 allows normal cycle operation only when start-up has been completed, when the cycle was previously operating, or when the cycle has been in standby mode. Constraint 15 limits cycle energy consumption during start-up, and constraint 16 enforces a maximum production limit. When operating, the cycle must produce a minimum amount of power. This is often referred to as the "turn-down ratio" limit. The minimum quantity can be adjusted if the cycle happens to also be starting up in a give time step, and this is handled in constraint 17. Start-up mode persistence is prevented in constraint 18. Cycle standby mode can be entered when the cycle was previous producing power. Standby mode can persist, but must form a contiguous span as shown in constraint 19. Finally, standby and start-up modes can't coincide (constraint 20), nor can standby and power producing mode (constraint 21).

3.4.3. Energy Balance

Several additional constraints are enforced:

$$s_t - s_{t-1} = \Delta \cdot [x_t^r - (Q^c y_t^{csu} + Q^b y_t^{csb} + x_t)] \quad \forall t \in \mathcal{T} \tag{22}$$
$$s_t \leq E^u \ \forall t \tag{23}$$
$$s_t \geq (-2 + y_{t+1}^{rsu} + y_t + y_t^{csb} + y_{t+1}) \cdot \frac{\max\left[0, Q_{t+1}^{in} - E^r/\Delta\right]}{Q_{t+1}^{in}} Q^u \Delta \quad \forall t \leq T-1 \in \mathcal{T} \tag{24}$$

Energy in, out, and stored in the TES system must balance as shown in constraint 22. the balance includes energy and power terms, hence power terms are multiplied by the time step to ensure unit compatibility. Constraint 23 ensures that energy in storage is less than the upper limit. Constraint 24 deals with an artifact arising from the difference between the modeling time resolution (hourly) and the amount of time required to start the plant, which may not be in units of whole hours. If the power cycle is running or in standby in time step t and in time step $t+1$, and if the receiver starts up in time $t+1$, then the minimum charge level in TES in time t must be sufficient to carry operation through the receiver start-up period. Note that $y_t + y_t^{csb} \leq 1$ is enforced elsewhere.

4. Dispatch Optimization Case Study

To demonstrate the dispatch optimization methodology implemented in SAM, the Power Purchase Agreement (PPA) price is calculated for a range of plant solar multiples and TES sizes. The solar multiple in this study is defined as the ratio between the thermal generation from the solar field under design conditions to the thermal energy consumption required by the power cycle at design conditions. As the solar multiple increases, the amount of thermal storage also typically must be increased.

The PPA price is an indicator of the baseline price of energy that an electricity buyer (e.g. a utility) will agree to pay a power producer. The PPA price is a useful surrogate for the profitability of a project in that it accounts for the variability in electricity value with time of day and time of year. As it is applied in SAM, the PPA price is multiplied by the hour-by-hour time of use (tariff) rate to determine the value of electricity generated by the plant over time. SAM calculates the PPA price assuming a target Internal Rate of Return (IRR) (11% in the current study) and an annual escalation rate of 1%. For this reason – and somewhat counterintuitively – a low PPA price is desirable. From the perspective of a power producer, a low PPA price with a fixed IRR indicates that a larger share of markets are accessible for electricity sales. A low calculated PPA price also enables an increase to project IRR should the *actual* PPA price be higher than the calculated price.

4.1. Analysis approach

The demonstration considers four market scenarios, three of which have been adopted from Guédez et al.. The fourth market scenario is the "generic summer peak" tariff schedule used as the SAM default for the MSPT model. The three Guédez et al. tariff schedules are shown in Figure 3. These schedules are adopted from the South Africa Department of Energy's Renewable Energy Independent Power Producer Program, which has put forward three market scenarios to highlight the importance of market or tariff factors on renewable deployment.

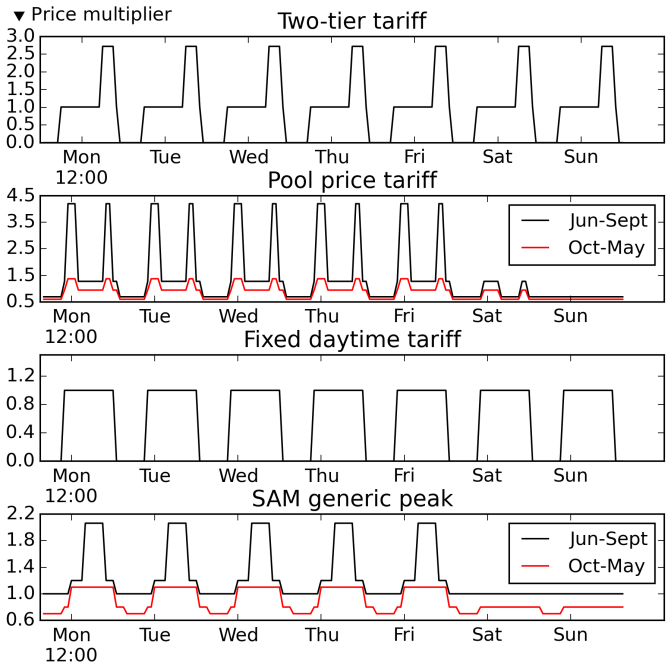


Figure 3: Market pricing scenarios presented by Guédez et al. [11]. These tariff schedules are implemented to determine the impact of dispatch optimization on system sizing.

The two-tier tariff market incentivizes daytime production with an evening spike. The pool-price tariff introduces an additional morning spike and weights incentives seasonally. The fixed daytime tariff allows sales during daytime hours, but is unique its binary incentives – no revenue is gained from second-tier operation. Finally, the SAM generic peak schedule combines features from the two-tier tariff and the pool price tariff.

Each market scenario is evaluated in SAM using the MSPT model presented above. A summary of key design parameters is provided in Table 3.

For this analysis, SAM is configured to automatically determine the best heliostat field layout given the specified solar multiple and other design parameters. This is accomplished using a nonlinear derivative free optimization scheme that improves the estimated present value of the plant by varying tower height, receiver height, and receiver diameter. The layout optimization is executed for each unique solar multiple using the BOBYQA search algorithm [16] from the NLOpt library [17] which is implemented in SAM.

Table 3: Case study plant design and control parameters

Parameter	**Units**	**Value**
Gross electrical output	MWe	115
Cycle design efficiency	%	41.2
Cycle design thermal input	MWt	278.1
Cycle maximum output	MWe	120.75
Cycle minimum output	MWe	28.75
Cycle start-up energy	MWt-hr	57.5
Cycle start-up time	hr	0.5
Cycle standby consumption	MWt	23
Receiver max. output (relative*)	-	1.2
Receiver min. output (relative)	-	0.25
Receiver start-up energy (relative)	-	0.25
Receiver start-up time (relative)	-	0.2
Receiver HTF temperature	°C	574
Heat rejection technology	-	Air cooled
Heliostat size	m^2	144.4
Maximum receiver flux	kW/m^2	1,000
Hours of TES	hr	1 . . . 18
Solar multiple	-	0.8 . . . 3

**Relative to receiver thermal input design point.*

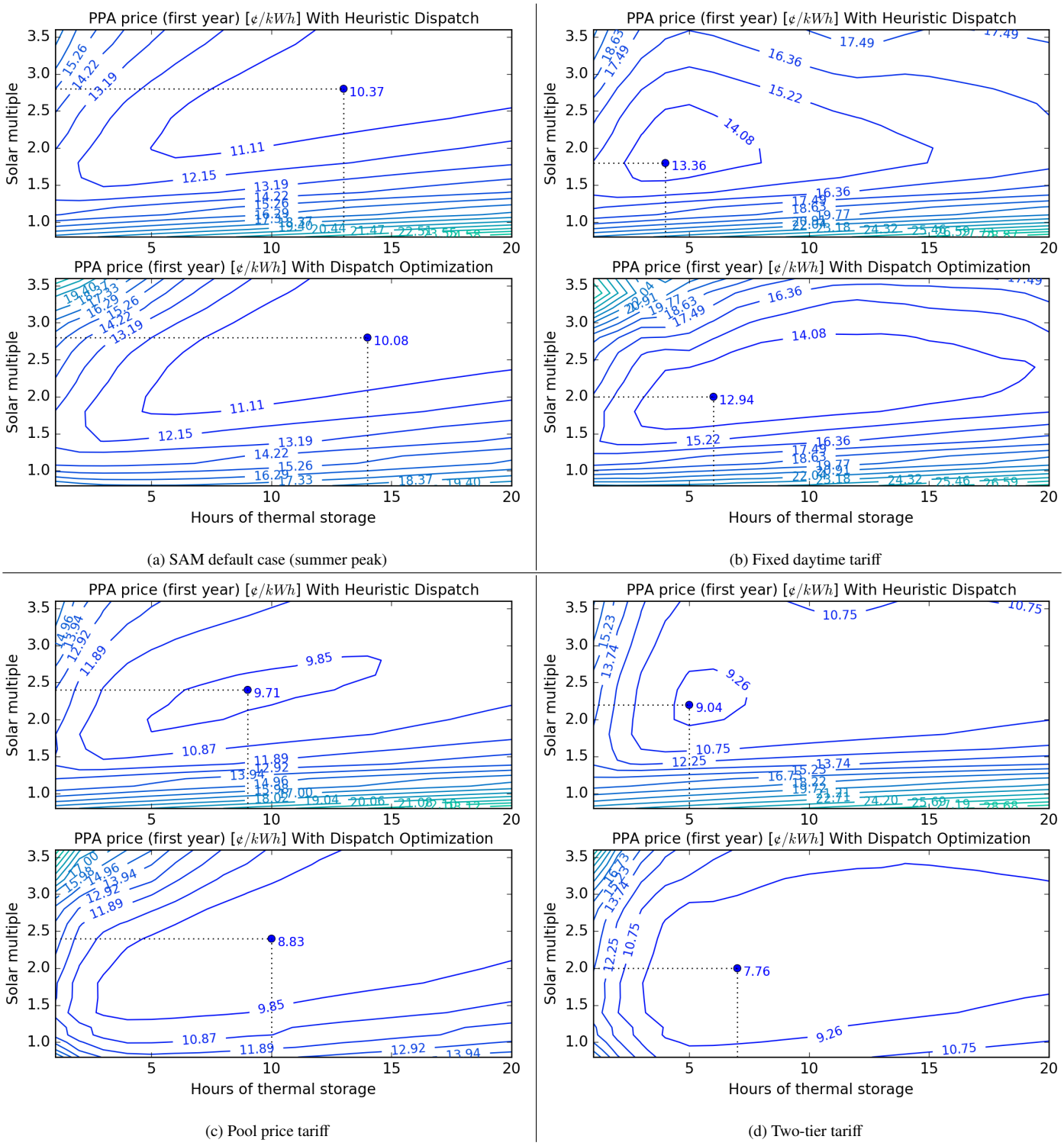
4.2. Results

The results of the PPA analysis are shown in Figure 4. Contour lines indicate constant PPA price, and the best PPA price for each scenario is plotted and labeled on each figure. Each market scenario includes a plot of PPA price with heuristic dispatch control (top plot of each quadrant) and PPA price with the dispatch optimization model developed as part of this work (bottom plot of each quadrant).

Several observations arise from the data. Firstly, the PPA price is generally lower in the dispatch optimization cases. This is true for both the lowest single PPA value and throughout the solution space. Looking specifically at the lowest PPA price in each scenario, heavily weighted schedules (pool price and two-tier) lead to more substantial PPA price reductions. The best-case reduction for each case is shown in Table 4. This finding indicates that dispatch optimization is an essential aspect of plant operation for "peaker" markets that provide relatively short time windows of high-value energy pricing. We also observe that systems operating in less heavily weighted market structures still benefit significantly from dispatch optimization.

Table 4: Reduction in PPA price at the optimal point for each case

Case	**Heur.**	**Disp.**	**Rel.**
SAM default	10.37	10.08	2.9%
Fixed daytime	13.36	12.94	3.1%
Pool price	9.71	8.83	9.1%
Two-tier	9.04	7.76	14.2%



(a) SAM default case (summer peak)

(b) Fixed daytime tariff

(c) Pool price tariff

(d) Two-tier tariff

Figure 4: Analysis results for tariff scenarios presented in Figure 3. The power purchase agreement price (PPA) is plotted as a function of solar multiple and hours of thermal energy storage. Each case is presented for the heuristic model and for the dispatch optimization model. The optimal sizing for each case is shown. The heliostat field layout was optimized for each solar multiple by SAM.

Secondly, dispatch optimization alters the size of TES and the solar multiple at which PPA price is minimized. This implies that dispatch optimization should not be relegated to only operational analyses, but rather should be part of the screening and design process. Table 5 describes the change in sizing between the heuristic and optimized dispatch cases for the four tariff scenarios.

An important feature of dispatch optimization is the apparent reduction in production variability during high-value time periods. This behavior is illustrated in Figure 5 for the Pool

Table 5: Location of minimum PPA price for each scenario.

	Solar multiple		Hours TES	
Case	**Heur.**	**Disp.**	**Heur.**	**Disp.**
SAM default	2.8	2.8	13	14
Fixed daytime	1.8	2.0	4	6
Pool price	2.4	2.4	8	10
Two-tier	2.2	2.0	5	7

Price tariff case. Plots 5a and 5b show the charge state of TES over the course of the day for each day of the year. Each traced line on the plot corresponds to the charge state profile for a particular day. Plot 5a shows the charge profile for heuristic dispatch, and plot 5b shows the profile for optimized dispatch. Also shown on the plot are the tariff multiplier schedules for summer (red) and winter (blue) that determine the revenue associated with generation during a particular hour of the day.

Dispatch optimization substantially changes the daily operational profile for the Pool Price case, and this behavior is representative of the other cases studied. Where heuristic dispatch allowed TES to discharge in the evening and late night hours, optimized dispatch typically reserves some quantity of TES to allow morning start-up. The TES profiles show that heuristic dispatch is much more operationally repetitive in comparison to optimized dispatch. This implies that optimized dispatch procedures account for externalities aside from the current charge state of TES and the electricity production target. Namely, the optimized procedure accounts for expected resource availability and future pricing to determine when TES should be used.

Plots 5c and 5d show the distribution of electricity generation for each hour of the day over the course of the year. Each box-whisker plot describes the variability in electricity generation for each day at the specified hour. The box limits indicate the extents of the 1st and 3rd quartiles. The whiskers correspond to twice the inner quartile range. Points that lie outside of this limit are plotted individually and are considered distribution outliers – though physically, each point represents the electricity generation from a particular hour in the simulation.

Plot 5c shows that electricity production is highly variable in the early morning, evening, and nighttime hours (tall boxes) and is less variable during daytime hours (narrow boxes). The variability in electricity generation is inversely related to solar resource availability – an intuitive observation. However, production is highly variable during peak revenue hours as shown by the tariff multipliers. Plot 5d shows the optimized dispatch case. Most notably, the dispatch optimization reduces electricity generation variability during peak revenue hours. Both the morning and evening tariff peaks demonstrate reliable generation throughout the year.

5. Conclusions

We develop and implement a MIP model to optimize the TES dispatch schedule for a molten salt power tower plant. In contrast to existing work, the MIP model is constructed within the SAM simulation environment and is used to guide the CSP simulation controller as plant performance is evaluated over an annual-hourly simulation. Because SAM's detailed performance model generates the simulation output, the accuracy and validity of the output data are not compromised by the approximate nature of the MIP formulation.

A case study demonstrating the methodology shows that a plant operating in one of several distinct tariff markets will benefit financially from dispatch optimization. Markets with heavily weighted pricing schemes or narrow windows of high revenue are of special interest, as PPA price for the optimally sized plant can improve by 10-15%. The case study also demonstrated that dispatch optimization is an important factor in both the operational scheme of the plant, but also the initial design.

The current dispatch optimization capabilities in SAM provide an opportunity for future work investigating how component and subsystem performance is affected by the demands of maximizing revenue under various market schedules. Specifically, future work will begin to incorporate forecast uncertainty, component operations and maintenance requirements and associated costs, and other aspects of plant operation that cannot be captured exclusively in a MIP model.

References

[1] A. Gil, M. Medrano, I. Martorell, A. Lázaro, P. Dolado, B. Zalba, L. F. Cabeza, State of the art on high temperature thermal energy storage for power generation. Part 1-Concepts, materials and modellization, Renewable and Sustainable Energy Reviews 14 (1) (2010) 31–55.

[2] R. I. Dunn, P. J. Hearps, M. N. Wright, Molten-salt power towers: Newly commercial concentrating solar storage, Proceedings of the IEEE 100 (2) (2012) 504–515.

[3] J. I. Burgaleta, S. Arias, D. Ramirez, Gemasolar: The First Tower Thermosolar Commercial Plant with Molten Salt Storage System, in: SolarPACES Conference, 2013.

[4] U. Herrmann, B. Kelly, H. Price, Two-tank molten salt storage for parabolic trough solar power plants, Energy 29 (5-6) (2004) 883–893.

[5] P. Denholm, M. Mehos, Enabling Greater Penetration of Solar Power via the Use of CSP with Thermal Energy Storage, Tech. Rep. November, Golden, CO, 2011.

[6] C. K. Ho, Software and Codes for Analysis of Concentrating Solar Power Technologies, Tech. Rep. December, Sandia National Laboratory, Albuquerque, NM, 2008.

[7] P. Griffin, K. Huschka, G. Morin, SOFTWARE FOR DESIGN, SIMULATION, AND COST ESTIMATION OF SOLAR THERMAL POWER AND HEAT CYCLES, in: Proceedings of the 2009 SolarPaces Internation Symposium, Berlin, Germany, 2009.

[8] R. Fourer, D. Gay, B. Kernighan, AMPL, vol. 119, Boyd & Fraser, 1993.

[9] R. E. Rosenthal, GAMS A User s Guide, Tech. Rep. July, GAMS Software GmbH, Frechen, Germany, 2013.

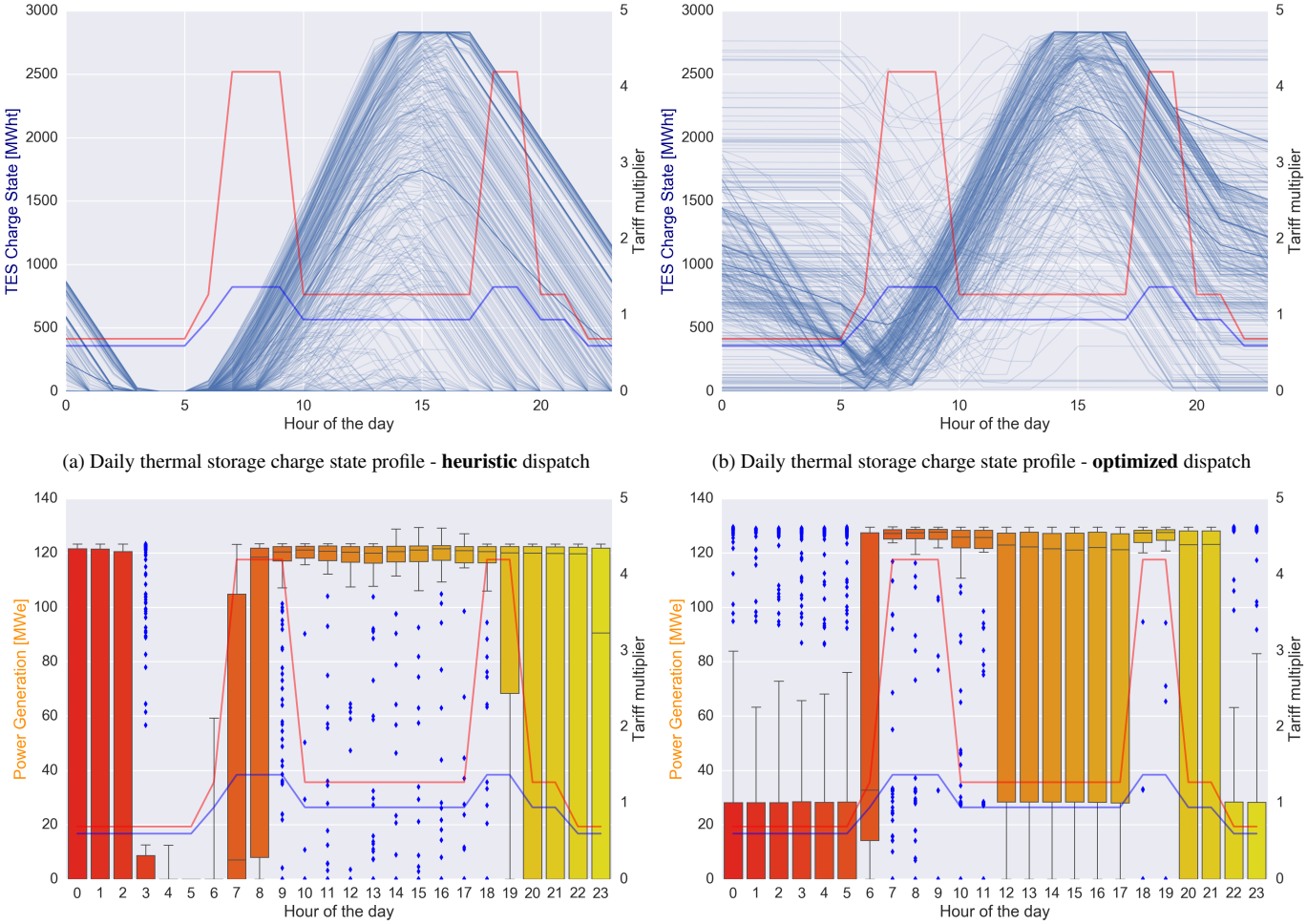
[10] N. Blair, A. P. Dobos, J. Freeman, T. Neises, M. Wagner, T. Ferguson, P. Gilman, System Advisor Model, SAM 2014.1.14 : General Description, Tech. Rep. February, National Renewable Energy Laboratory, Golden, CO, 2014.

[11] R. Guédez, F. Ferragut, P. L. Défense, M. Topel, I. Callaba, C. D. Pérez-segarra, A METHODOLOGY FOR DETERMINING OPTIMUM SOLAR TOWER PLANT CONFIGURATIONS AND OPERATING STRATEGIES TO MAXIMIZE PROFITS, in: Proceedings of the ASME 2015 9th International Conference on Energy Sustainability, American Society of Mechanical Engineers, San Diego, CA, 1–15, 2015.

[12] S. H. Madaeni, R. Sioshansi, P. Denholm, How Thermal Energy Storage Enhances the Economic Viability of Concentrating Solar Power, Proceedings of the IEEE 10 (2) (2012) 335–347.

[13] R. Sioshansi, The value of concentrating solar power and thermal energy storage, Energy, IEEE Transactions on 1 (3) (2010) 173–183.

[14] National Renewable Energy Laboratory, System Advisor Model, 2012.



(a) Daily thermal storage charge state profile - **heuristic** dispatch

(b) Daily thermal storage charge state profile - **optimized** dispatch

(c) Annual variability of electricity generation - **heuristic** dispatch

(d) Annual variability of electricity generation - **optimized** dispatch

Figure 5: Comparison of performance profiles for the Pool Price tariff schedule. Plots 5a and 5b show traces of the TES charge state for each day of the year. Plots 5c and 5d show boxplots of daily electricity production variability over a year grouped by hour of the day. Each boxplot indicates the mean annual electricity generation by hour, the first and third quartile limits (box limits), and two times the interquartile range (whiskers). "Outliers" are shown as points. Summer (red) and winter (blue) tariff multipliers are overlaid on each plot.

[15] R. Guédez, J. Spelling, B. Laumert, T. Fransson, Reducing the Number of Turbine Starts in Concentrating Solar Power Plants Through the Integration of Thermal Energy Storage, in: Proceedings of the ASME Turbo Expo 2013, San Antonio, Texas, 1–10, 2013.

[16] M. Powell, The BOBYQA algorithm for bound constrained optimization without derivatives, University of Cambridge NA Report NA2009/06 .

[17] S. G. Johnson, The NLopt nonlinear-optimization package, 2014.