Building Technology & Urban Systems Division
Energy Technologies Area
Lawrence Berkeley National Laboratory

Automated production of optimization-based control logics for dynamic façade systems, with experimental application to two-zone external venetian blinds

Brian Coffey, Andrew McNeil, Thierry Nouidui, Eleanor S. Lee

Lawrence Berkeley National Laboratory

Energy Technologies Area
December 2013 (rev. May 2022)

This work was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy,
Building Technologies Office, of the U.S. Department of Energy
under Contract No. DE-AC02-05CH11231.

Copyright Notice

Automated Production of Optimization-Based Control Logics for Dynamic Facade Systems, with Experimental Application to Two-Zone External Venetian Blinds

Brian Coffey, Andrew McNeil, Thierry Nouidui, Eleanor S. Lee

Lawrence Berkeley National Laboratory, 1 Cyclotron Rd, Berkeley, CA 94720, USA

Abstract

The primary goal of this research is to devise a system that produces controllers for complex fenestration systems that perform nearly as well as Model Predictive Control but at a level of cost and implementation complexity that rivals simple heuristic controls. To this end, a cloud-based automated controller production system has been set up for a motorized external Venetian blind device, with a simple web interface that can be used by non-experts. The computation cost per controller is in the range of a few dollars, and the control logic is simple enough to be implemented on small and cheap distributed controllers. The web interface allows the user to specify some details of their particular building and window configuration, including orientation, latitude, interior geometries, and lighting and HVAC system parameters. Upon submittal, a cloud-based system configures the necessary files and commands, and then runs thousands of optimizations with them. Once the calculations are finished, the system produces a lookup table and interpolation-based controller scripts that can be used on a simple and cheap distributed controller. This paper describes the underlying models and optimization processes. It also describes the resulting control logics for two cases tested at Lawrence Berkeley National Laboratory's Advanced Windows Testbed Facility: illuminance maximization subject to glare constraints; and lighting + HVAC energy minimization. The performance of the model-based controllers produced by the automated web-based system are compared to a heuristic 'block beam' controller in physical experiments at the Testbed. The experimental results are supplemented by simulation experiments with the same configuration as the Testbed. The results show the illuminance maximizing controller significantly outperforms the heuristic controller in terms of glare avoidance, and also outperforms it in terms of hours of daylight autonomy. The energy minimizing controller also outperforms the heuristic controller. This paper also discusses how the web-based system may be extended to consider other configurations, such as electrochromic windows and thermally massive HVAC systems. Potential roles for this type of system within the building design and construction industry are discussed.

Keywords: Model Predictive Control (MPC), dynamic facades, Complex Fenestration Systems (CFS), Bi-directional Scattering Distribution Functions (BSDF)

1. Introduction

In a study evaluating greenhouse gas (GHG) emission reduction strategies based on cost minimization criteria, the International Energy Agency (IEA, 2011) concluded that half of the GHG reductions required to meet a 2°C target would need to be achieved through energy-efficiency measures (EEMs). Techno-economic analyses have identified significant cost-savings potential for many building energy efficiency measures, but in reality, real estate market barriers such as split incentives, hidden costs, transaction costs, risk, and access to capital have slowed market realization of this potential. This "energy paradox", where there is a fundamental contradiction between slow market uptake of energy efficiency technologies and profitability of the measures, has in part been countered by sustainable, green building practices where adoption of EEMs has increased despite the paradox (Eichholtz, Kok and Quigley, 2013) and more lately by concerns for human health and well-being particularly in light of the recent COVID-19 pandemic. Market adoption of healthy building practices by early adopters has been driven by corporate reputation, belief in financial benefits in the long term, and social responsibility, but to achieve mainstream adoption, empirical evidence on the positive financial impacts of EEMs on real estate premiums are still needed (Tan et al., 2021).

Natural daylight through windows and skylights has long been advocated for by both the architectural and real estate industry as being both beneficial for humans and for property valuation. Daylight has been linked positively to human physiological and psychological health, increased productivity, and higher workplace satisfaction (Boyce, Hunter and Howlett, 2003; Aries, Veitch and Newsham, 2010; Veitch, Christoffersen and Galasiu, 2013). In a study involving a sample of 5154 office spaces, Turan et al. (Turan et al, 2020) found that occupied spaces with high amounts of daylight have a 5-6% value premium over spaces with low amounts of daylight in the high-density urban environment of Manhattan. Daylighting also has an estimated potential to reduce lighting energy use by 1.055 $\times 10^{18}$ joules in the U.S. (Arasteh et al., 2006) with additional significant savings potential worldwide given that lighting accounts for 15% of global electricity consumption. Daylight is also an invaluable asset in the event of an extended power out-

age. Managing daylight through windows, however, is a complicated affair involving trade-offs between aspects that positively affect human health and indoor environmental quality (i.e., daylight and view) and those that negatively affect building energy efficiency and comfort (i.e., solar gains, glare, thermal discomfort). Energy-efficient, healthy, and cost-effective solutions are highly dependent on the particulars of individual site, building, and space characteristics and occupant tasks and orientation to the windows (Gentile et al, 2021).

Automated, operable shading can provide such solutions, but delivering high performance at low cost has been a challenge historically for two reasons: (1) from a technical perspective, heuristic controls tend to be far from optimal, but determining optimal control for particular configurations requires complex models and algorithms to account for glare, daylighting and the effects of solar gains on HVAC energy use; and (2) from an economics or business perspective, the return on investment for any one building implementation tends to be poor because of the cost of one-off implementation of complex customized control. Solving this twin challenge could not only facilitate adoption of automated shading systems by the buildings industry, thus reducing energy use and associated GHG emissions, but could also help to spur further development of more advanced active facade technologies, endorsing the vision of future buildings that more intelligently manage the interface between indoors and outdoors.

The primary goal of this research is to devise a system that automatically produces controllers that perform nearly as well as Model Predictive Control but at a level of cost and implementation complexity that rivals heuristic controls. MPC is a common controls technique in other fields. It consists of a repeated real-time solution of a finite-time-horizon optimization problem with a model of the system under consideration. It has received significant research attention in the buildings industry over the past decade, but has not yet made significant inroads into the mass market (Drgona et al, 2020). Among the reasons for this lack of market penetration include the relatively poor payback period on complex energy-minimizing building controls because of the one-time cost associated with their development and installation. MPC has been successful in the chemical processing industry because each plant is valuable enough to warrant significant investment in customized controls. It has been successful in some aerospace and automotive applications because of a combination of unit value and production environments that allow for a concentration of controls expertise to influence a large number of units. Buildings, on the other hand, tend to be designed as unique projects, and the energy cost savings available through each individual MPC implementation tend to be relatively modest.

The techniques described herein address these general challenges associated with model-based controls (MBC) in building systems. In particular, a MBC configuration is automatically generated based on a few simple inputs via a web-based interface and sampled using the Hookes Jeeves generalized pattern search (GPS) algorithm to produce an approximation to its behavior. The output is then encapsulated in a simple lookup table controller that can be implemented at low cost. Section 2 provides an overview of the controller construction process, including the web interface, the automated cloud-computing process to produce a tailored near-optimal lookup table for control, and how the results are used in simulation and/or in physical control hardware. Section 3 describes the underlying Radiance and Modelica models used in the process. Section 4 describes the optimization processes that produce near-optimal controllers using those models. Section 5 describes some simulation-based analyses of the controllers' performance. Section 6 describes physical experiments with the controllers at a full-scale outdoor testbed. Section 7 provides some additional general discussion, including next steps in development, and Section 8 draws some conclusions.

2. Overview of the Controller Construction Process

Construction of the controller is intended to be conducted by non-expert users who specify the building zone's internal geometry and external solar obstructions, click a button and receive customized near-optimal control logic for a small computing cost. The complicated analysis associated with developing the site-specific controller occurs "under the hood", but both the user interface and the resulting controllers are meant to be simple and easy to use. Figure 1 shows an example web interface. The user tailors the control logic to their particular latitude, facade orientation, lighting system and HVAC configuration, then chooses from various control objectives and constraint options. For the purposes of this study, the web interface has been set up to configure a controller for a two-zone, outdoor, motorized, horizontal louver system for a testbed private office in the Advanced Windows Testbed at the Lawrence Berkeley National Laboratory. In a more developed product, the user interface would provide a library of shade options and allow the user to upload their site-specific zone geometries.

Once users have defined their desired configuration through the web interface, a central machine takes the user inputs and sets up the controller construction process, launches other cloud-computing instances, performs thousands of optimizations on them, and then collects and processes the results before sending them to the user. The controller construction process consists of two major stages: (1) determination of glare constraints (if applicable) and (2) optimization over a grid of conditions. The process, and the underlying models used in the process, are described in Sections 3 and 4. One output file describes the conditions tested and the optimal values of the shade position. Two Python files can also be emailed to the user, one of which provides a desktop version of an interpolation controller based on the lookup table, and the other provides a version of the controller that is ready to use on an appropriately configured embedded controller. The last available option is for the user to receive some of the detailed process outputs, which include a listing of all of the points tested during the optimizations - this latter option is intended for reporting and/or verification purposes.

The current solution was tested with the Raspberry Pi platform (Figure 2), which is significantly more powerful than needed for this purpose, but is much easier to develop when

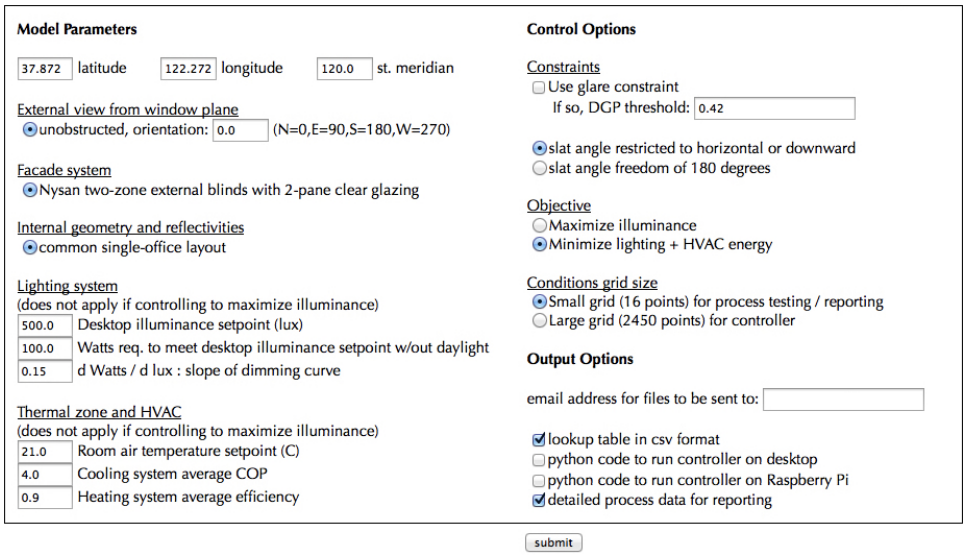
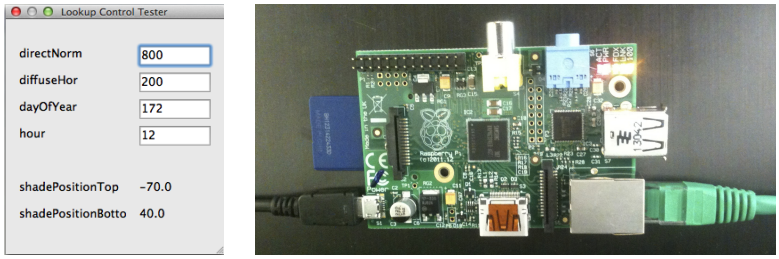


Figure 1: Simplified example of a web interface to the control logic constructor



(a) (b)

Figure 2: (a) Screenshot of controller GUI app, (b) Raspberry Pi

compared to less expensive distributed controllers. In general, the hardware needs only to be able to store a small lookup table worth of data (approximately 50-200 KB when stored in flat text file format), and perform a four- or five-dimension nearest-neighbor interpolation at each control timestep (15 min in our test cases) given sensor readings.

3. Underlying Models

3.1. Overview

The following models form the basis for the two controllers described in Sections 4.2 and 4.3: (1) maximize daylight illuminance without glare so as to reduce lighting energy use (”illumMax”), or (2) minimize HVAC and lighting energy use (”energyMin”). Forecasting loads based on the thermal mass of the building is not included; the controller is designed to base control on conditions within the present time step. MPC control involving thermal mass was investigated in a prior study (Coffey, 2012).

Equations 1 to 3 summarize the three major calculation groups in the underlying models, where u_1 is the upper blind position, u_2 is the lower blind position, w_1 is the day of year, w_2 is the time of day, w_3 is the direct normal radiation (in W/m^2), w_4 is the diffuse horizontal radiation (in W/m^2), and w_5 is the outdoor dry bulb temperature (in °C).

The HVAC+lighting power consumption calculation is somewhat more involved than the other two; the desktop illuminance is used as a sub-component within it to determine lighting energy use, along with three other major subcomponents, as shown in Figure 3. Note that the HVAC+lighting power calculation is the only one of the three that requires the outdoor temperature as an input.

3.2. Radiance Calculations

Figure 4(a) illustrates parts of the Radiance model used for the visible spectrum calculations, and Figure 4(b) illustrates the main parts of the Radiance model used to calculate the solar gains on the interior surfaces. Note that the sky model, BSDFs and interior matrices are different between the two models, but the exterior view matrices are the same.

For the case of the Advanced Windows Testbed Facility, which is used in the case studies below, the exterior view matrices and interior view matrices are set to match the particulars of the perimeter office space geometries and reflectivities of the Facility. There are no exterior obstructions to the south-facing facade at the Facility, so the exterior view matrices are for an unobstructed view. The interior view matrices for the

$$\text{glare (DGP)} = \text{glare}\,(u_1, u_2, w_1, w_2, w_3, w_4) \quad (1)$$

$$\text{desktop illuminance} = \text{illuminance}\,(u_1, u_2, w_1, w_2, w_3, w_4) \quad (2)$$

$$\text{HVAC + lighting power consumption} = \text{energy}\,(u_1, u_2, w_1, w_2, w_3, w_4, w_5) \quad (3)$$

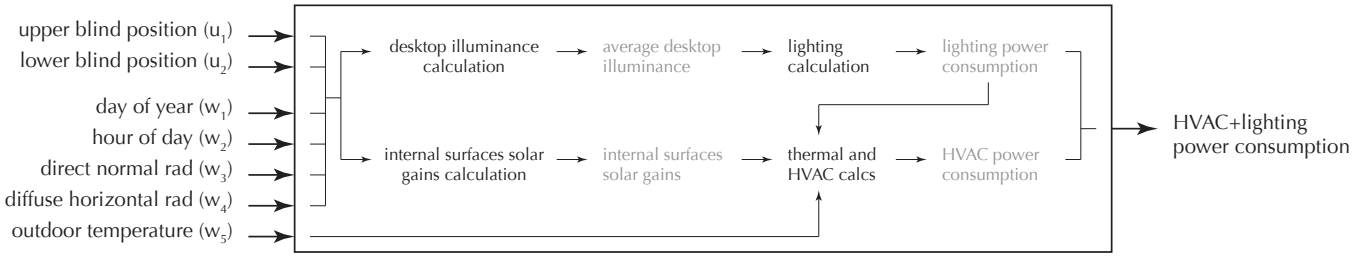


Figure 3: Overview: HVAC+Lighting Power Consumption Calculation Components

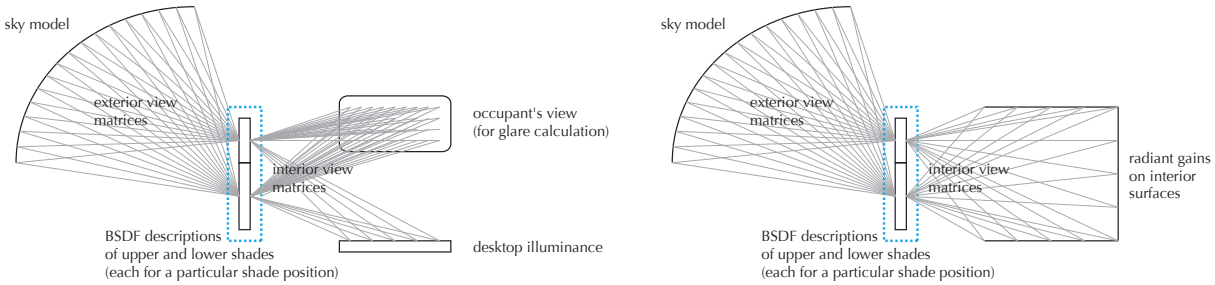


Figure 4: Radiance models: (a) visible spectrum, (b) full solar spectrum

desktop illuminance and glare calculation are based on calculations made with a Radiance model description of the office space. The office space and its visible spectrum instrumentation is illustrated in Figure 5, where L1-L6 are lux sensors at the workplane level (note that only L3-L6 are used to calculate the average desktop illuminance) and C1 and C2 are HDR cameras used to 'measure' glare - note that C1 (located at the back of the room) faces the window, while C2 (located at the occupant's expected seating location) faces the desk and wall. For comparison, as described in the Glare Calculation section below, a glare calculation viewpoint is also considered at the same location as C2 but facing the window. But aside from this comparison, the glare calculation is completed for the space position corresponding to C1, and may be compared to the 'measured' glare values for that position.

3.2.1. Glare Calculation

The glare calcation uses dctimestep and the Radiance-derived internal matrices to construct a HDR image of what camera C1 would see, and then uses evalglare to calculate the daylight glare probability (DGP), as described in Equations 4 and 5, where $\mathbf{V}_{i_{glare}}$ is the internal view matrix for glare for shading control zone area i, $\mathbf{T}_{visible}(u_i)$ is the BSDF matrix for the visible spectrum for the given shade position u_i, $\mathbf{D}_i$ is the external view matrix for the shading control zone area i, and $\mathbf{s}_{visible}$ is the sky vector.

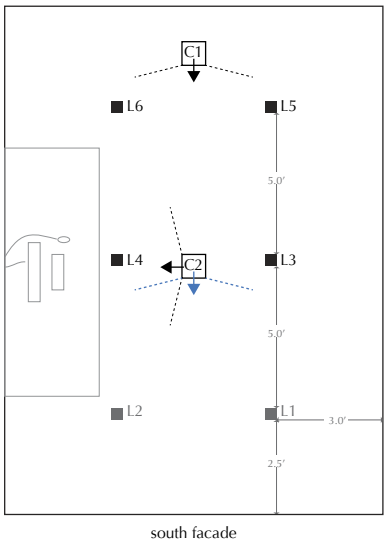


Figure 5: Office space geometries and visible spectrum instrumentation at the Advanced Windows Testbed Facility

$$\text{C1view.hdr} = \sum_{i=1}^{2} \mathbf{V}_{i_{glare}} \mathbf{T}_{visible}(u_i) \mathbf{D}_i \mathbf{s}_{visible} \quad (4)$$

$$DGP_{C1} = \text{evalGlare (C1view.hdr)} \quad (5)$$

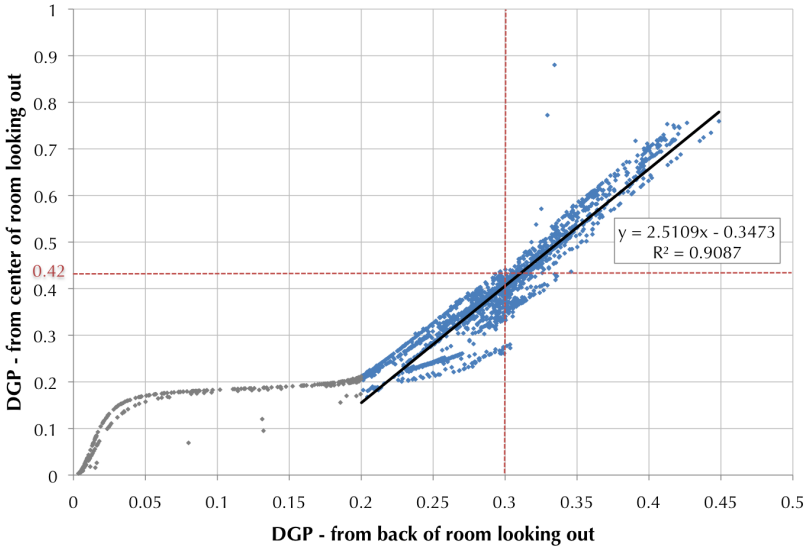


Figure 6: Comparison of calculated DGP values at two positions in the room

Work by (Wienold, 2009) suggests that DGP values greater than 0.35 correspond to perceptible glare, and values greater than 0.42 are potentially disruptive. As discussed in the optimization processes section below, an important input that must be decided upon is what DGP value to use as a threshold that the controller should try to stay below. We are using the camera position C1 as our glare calculation position so that we can more easily compare it to 'measured' data, but we may also want to avoid potentially disruptive glare for a person sitting at position C2 and looking towards the window. If we select a low enough DGP threshold for the C1 position, we can avoid discomfort glare at the C2 position. Figure 6 shows how the calculated DGP values at the two positions compare, and suggests that a DGP threshold of 0.30 for C1 should keep the DGP at C2 below 0.42 almost all of the time.

3.2.2. Desktop Illuminance Calculation

Again using dctimestep and the Radiance-derived internal matrices, the illuminance is calculated for four points on the workplane, points L3-L6 in Figure 5 (Equation 6), and the average of these values is then calculated (Equation 7).

$$[l_3, l_4, l_5, l_6] = \sum_{i=1}^{2} \mathbf{V}_{i_{illum}} \mathbf{T}_{visible}(u_i) \mathbf{D}_i \mathbf{s}_{visible} \tag{6}$$

$$l_{daylight} = \frac{\sum_{i=1}^{4} l_i}{4} \tag{7}$$

3.2.3. Solar Gains Calculation

The solar gains on each of the interior surfaces are also calculated using dctimestep. This forms a part of the HVAC+lighting power consumption calculation (Equation 3 and Figure 3). The calculation of the absorbed solar gains is also a matrix calculation (with the absorbed matrix coefficients calculated using Window 7). Note that relative to the gains on the internal surfaces, the absorbed solar gains in the glazing are very small.

$$\mathbf{Q}_{interiorSolarGains} = \sum_{i=1}^{2} \mathbf{V}_{i_{solarGains}} \mathbf{T}_{solar}(u_i) \mathbf{D}_i \mathbf{s}_{solar} \tag{8}$$

3.3. Energy Calculations using Python and Modelica

The energy calculation (Equation 3) starts with the computation of the desktop daylight illuminance and the interior solar gains, as described above. Given the desktop daylight illuminance, a simple calculation is performed to determine the artificial lighting requirement. The artificial lighting requirement and solar gains are then input into a Modelica-based calculation of the space thermal loads and resulting HVAC energy consumption.

3.3.1. Lighting Energy Calculation

The artificial lighting is controlled locally, dimming as necessary to maintain an estimated 500 lux on the desktop.[1] The artificial lighting requirement, in lux, is calculated simply as shown in Equation 9. The lighting power consumption, in W, is then calculated as shown in Equation 10, which is based on the correlation with measured data from an overnight test of artificial light output, shown in Figure 7. Note that the model assumes that all of the lighting energy use also ends up in the space as a convective heat gain.

[1] The lighting control is actually based on a ceiling sensor that has been calibrated to roughly match the workplane lux measurements, rather than using the workplane lux measurements directly, to more accurately approximate standard daylight dimming control practice.

$$l_{artificialReq} = \max(0, 500\ \text{lux} - l_{daylight}) \tag{9}$$

$$P_{lighting} = \begin{cases} 1.6185\, l_{artificialReq} + 26.642 & \text{if } l_{artificialReq} < 32.912 \\ 0.4256\, l_{artificialReq} + 65.903 & \text{if } l_{artificialReq} \geq 32.912 \end{cases} \tag{10}$$

$$P_{thermalLoad} = \text{ModelicaModel}\,(w_1, w_2, w_3, w_4, w_5, P_{lighting}, \mathbf{Q}_{interiorSolarGains}) \tag{11}$$

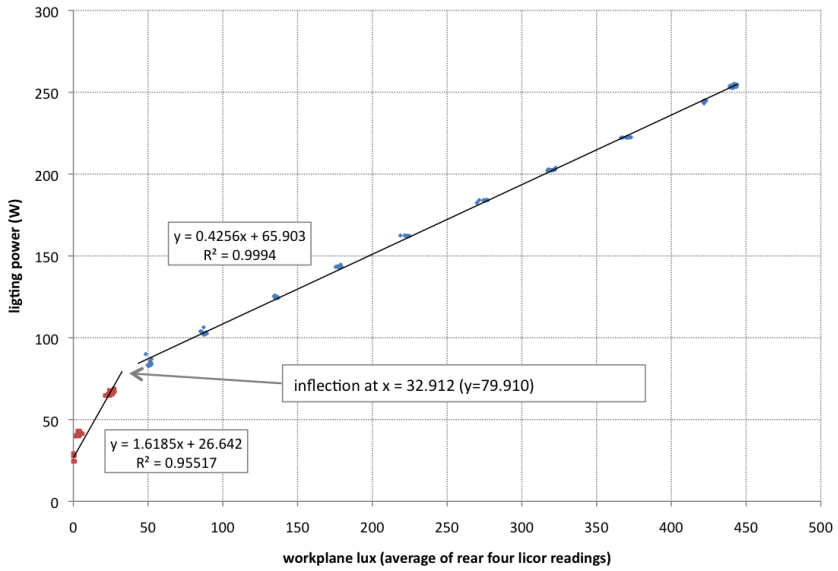


Figure 7: Lighting power model

3.3.2. Heating / Cooling Requirement Calculation

Note that all of the calculations up to this point are independent of the outdoor temperature, while the heating / cooling requirements and power consumption are dependent on it. The calculation uses a Modelica thermal model, as noted in Equation 11. Figure 8 illustrates some key variables in the Modelica thermal zone model, built using the Modelica Buildings Library (Wetter et al., 2018).[2] The model is run for a 15 minute simulation to reach a near steady state (there is very little thermal mass in the model), and the heating/cooling requirement at that point is used.

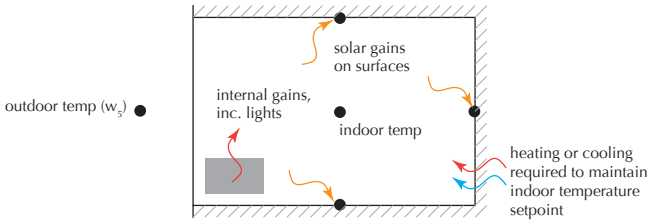


Figure 8: Modelica model - key variables

[2]The model was constructed in Dymola and exported as an executable using a Binary Model Export license, which allows that particular model to be simulated on a multitude of machines without any further licensing.

3.3.3. Power Consumption Calculations

The HVAC power is calculated from the heating / cooling requirement with the assumption of a constant heating efficiency $\mu_{heating}$ and a constant $COP_{cooling}$, as shown in Equation 12. Both values are modifiable by the user. In the case study described herein, they are assumed to be 90% and 3.5. The total energy is then calculated simply as the addition of the calculated artificial lighting use and the calculated HVAC energy use, as shown in Equation 13.

$$P_{HVAC} = \begin{cases} \frac{P_{thermalLoad}}{\mu_{heating}} & \text{if } P_{thermalLoad} < 0 \\ \frac{P_{thermalLoad}}{COP_{cooling}} & \text{if } P_{thermalLoad} \geq 0 \end{cases} \tag{12}$$

$$P_{total} = P_{HVAC} + P_{lighting} \tag{13}$$

3.4. Summary

Figure 9 shows some of the intermediary and final outputs of the models for three sets of conditions and the full range of possible shading position values. Figure 10 shows an example day of simulated values compared to measured values from the Advanced Windows Testbed Facility.

4. Optimization Processes

The controller construction process consists of two major stages: (1) determination of glare constraints (if applicable) and (2) optimization over a grid of conditions. The

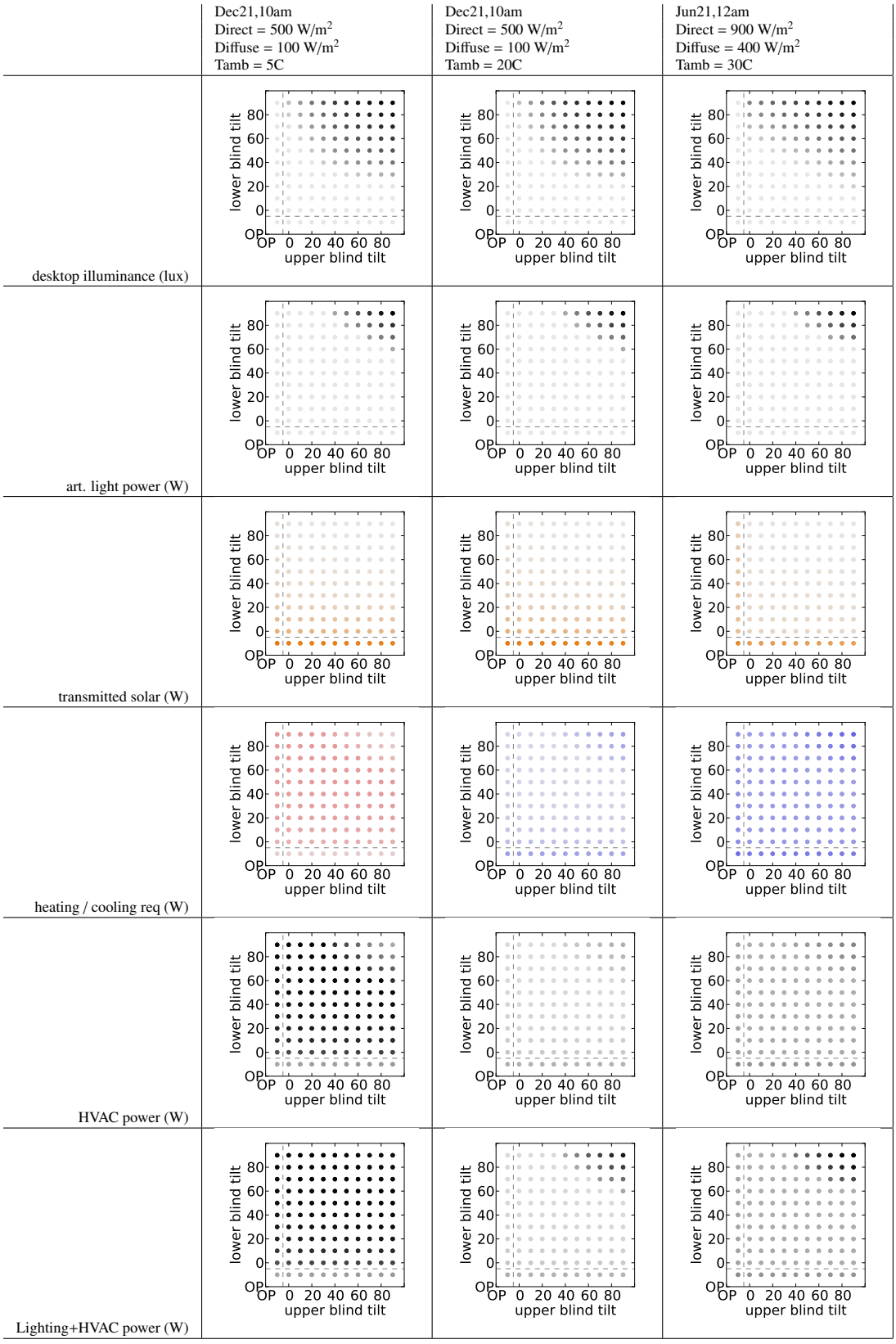


Figure 9: Model outputs as function of shade positions and conditions

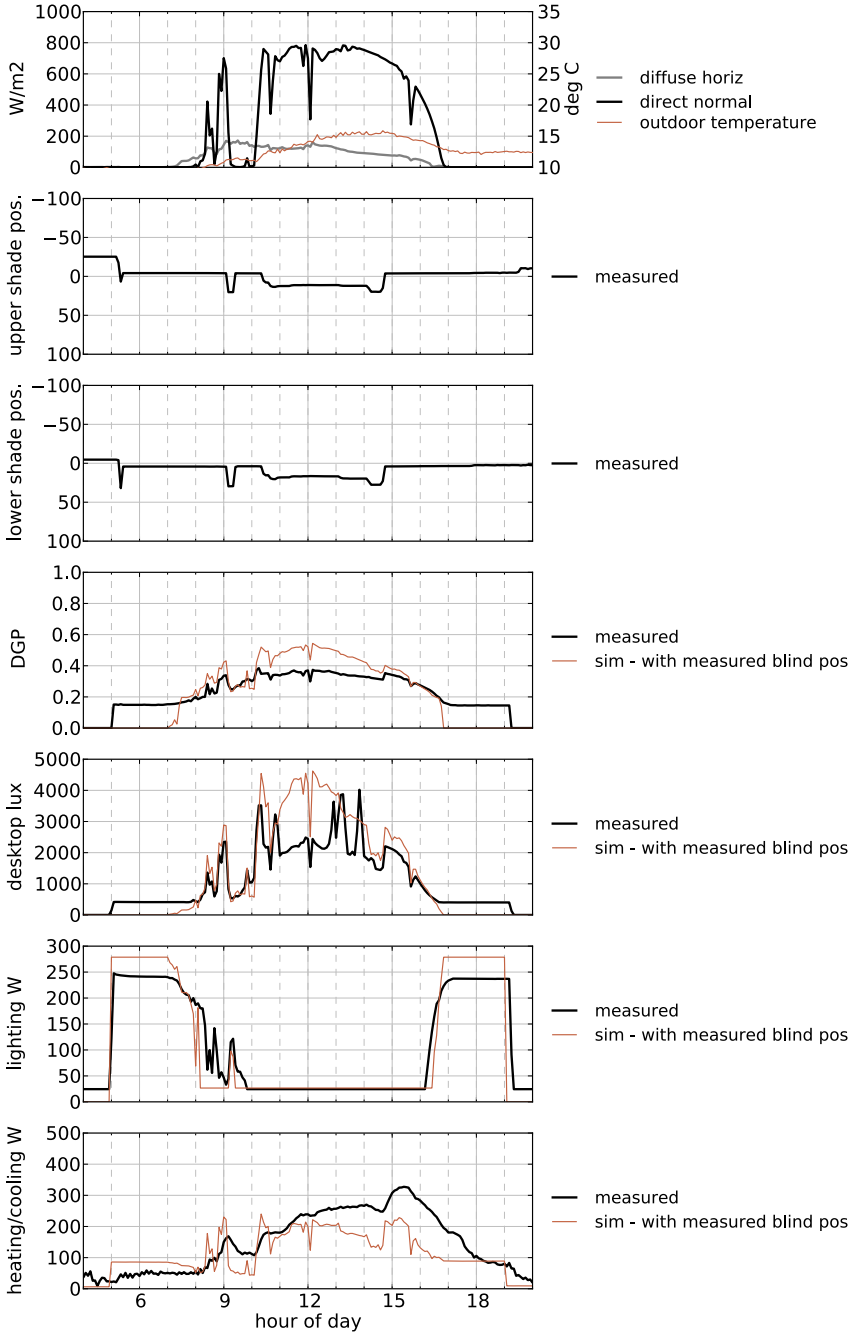


Figure 10: Model outputs versus measured data: November 21 ('heuristic' controller running)

models described in the previous section are used to calculate values at particular points (ie. for particular vectors of $\{u_1, u_2, w_1, w_2, w_3, w_4, w_5\}$ values), as determined by the processes described in this section.

4.1. Constraint: Discomfort Glare Probability (DGP)

Figure 11 shows the derivation of glare constraint lines (in red) for 4 particular sets of solar conditions, alongside the glare model outputs for every possible lower blind tilt angle (in black). Note that in all the graphs throughout, for simplicity of representation, the fully open position is considered as having an angle of -10. In each of the rows in Figure 11, the three graphs on the left show DGP as a function of the lower blind tilt angle, given some constant upper blind tilt angle (noted directly above the graph). The constraints are defined by finding the points whose DGP values are closest to but less than the user-specified DGP threshold. The graph on the right of each row summarizes the constraints over the full set of possible upper blind positions. The three graphs on the left are thus detailed slices through the graph on the right. (Note that the axes are changed in the graph on the right, with the upper blind position

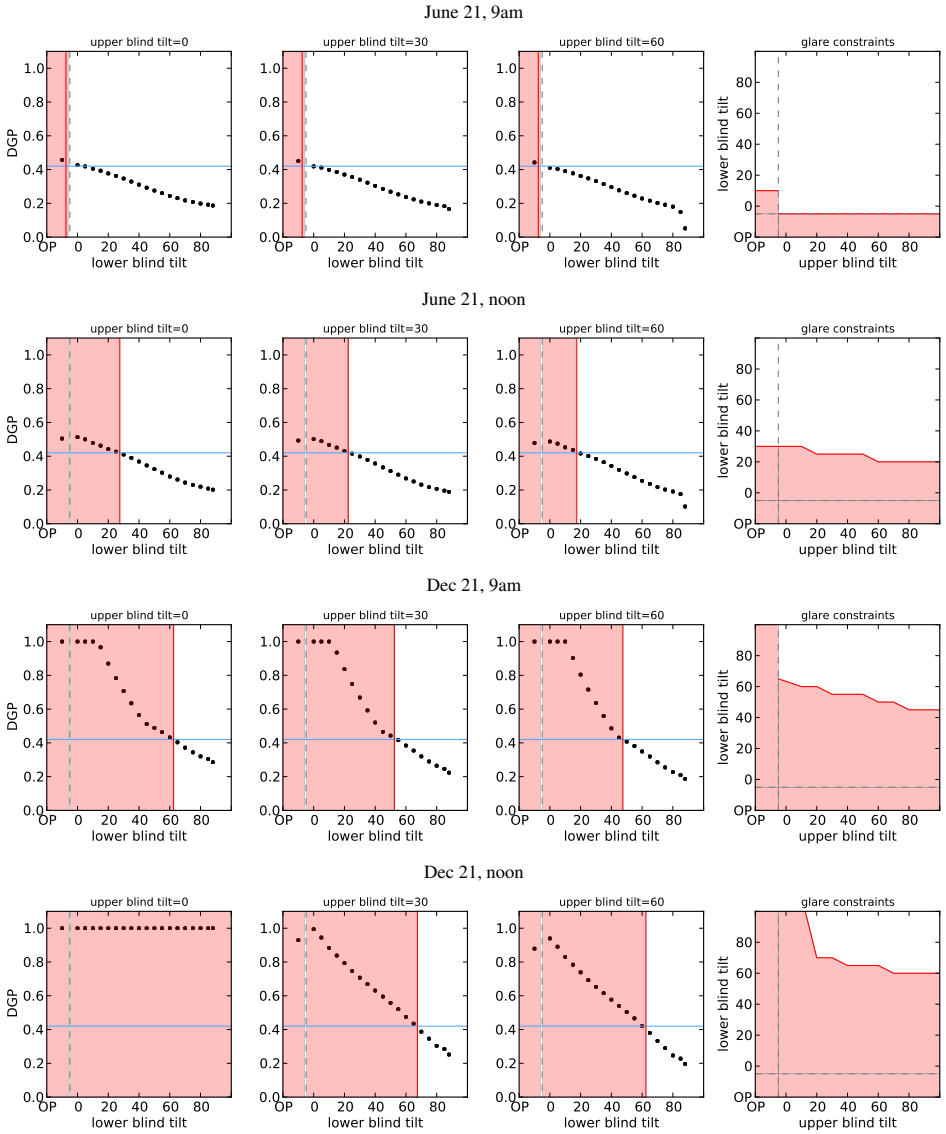


Figure 11: DGP and Glare Constraints - Direct Normal = 900 W/m^2, Diffuse Horiz = 100 W/m^2

shown on the horizontal and the lower blind position shown on the vertical.) In short, the graphs on the right show the unacceptable combinations of upper and lower blind positions as shaded red.

These illustrative graphs show calculated DGP values for all of the blind positions (black dots). However, it is not necessary to simulate every point - we only need to evaluate a few points to locate the places where the DGP curve crosses the DGP threshold line. For the case shown in these figures, where the slat angles are restricted to horizontal (0°) and downwards (positive): for each of the possible values of the upper blind position, one optimization routine is run, starting at a lower blind tilt angle of +90° and searching in a negative direction (by bounds of decreasing size, using the GPS Hookes Jeeves algorithm in GenOpt) until it finds the cross-over point (the red line on the graphs on the left); and for each of the possible values of the upper blind position, a single-simulation check is then also made for the case of the open lower blind position, to see if it produces an acceptable DGP. (For the case where the slat angles are not restricted to horizontal and downwards, but instead may vary from +90° to -90°, the approach is the same, except that two optimization routines are used, starting at the two extremes and moving towards the middle, with two red lines being produced.)

For the production of the controller itself, the glare constraint lines are calculated over a grid of 2450 possible solar conditions, as shown in Table 1.

Table 1: Conditions grid used for glare constraints calculations

	range	number of points
day of year	June 21 - Dec 21	7
hour of day	6 - 18	7
direct normal radiation (W/m^2)	100 - 1000	10
diffuse horizontal radiation (W/m^2)	100 - 500	5
total number of points		2450

4.2. Control Objective Option #1: Illuminance Maximization

4.2.1. Mathematical Definition

For the case of illuminance maximization, the optimal control problem for any given set of solar conditions is defined in Equation 14, where, as before, u_1 is the upper blind position, u_2 is the lower blind position, w_1 is the day of year, w_2 is the hour of day, w_3 is the direct normal radiation in W/m^2 and w_4 is the diffuse horizontal radiation in W/m^2, and where 'OP' stands for the fully open (retracted) position and γ is the upper bound on the blind tilt angle (-90 if unrestricted, 0 if restricted to horizontal and downward only).[3]

$$\begin{aligned} \max_{u_1,u_2} \quad & \text{illuminance}(u_1, u_2, w_1, w_2, w_3, w_4) \\ \text{s.t.} \quad & \text{glare}(u_1, u_2, w_1, w_2, w_3, w_4) \leq 0.42 \\ & u_1, u_2 \in [\gamma, 90] \cup [OP] \end{aligned} \tag{14}$$

To increase computational efficiency, the glare constraint is first made explicit, as described in the previous section. For any given values of (w_1, w_2, w_3, w_4), the constraints are then defined by two (if $\gamma = 0$) or three (if $\gamma = -90$) lines through the (u_1, u_2) space. The case where the blind tilt angle is restricted to horizontal and downwards ($\gamma = 0$) is shown in Equation 15.

$$\begin{aligned} \max_{u_1,u_2} \quad & \text{illuminance}(u_1, u_2, w_1, w_2, w_3, w_4) \\ \text{s.t.} \quad & (u_2 \leq \text{constraintLine1}(u_1, w_1, w_2, w_3, w_4)) \text{ OR} \\ & (u_2 \geq \text{constraintLine2}(u_1, w_1, w_2, w_3, w_4)), \\ & \forall u_1, u_1, u_2 \in [0, 90] \cup [OP] \end{aligned} \tag{15}$$

4.2.2. Illustrative Case: Maximizing illuminance subject to glare constraints

Figure 12 illustrates the illuminance-maximizing optimization algorithm (for the downward-only case), with its four starting points, applied to a representative grid of 16 conditions cases. (In the case of unrestricted blind tilt, 7 starting points are used for the optimization.) The underlying black-grey-white dots are the calculated illuminance values (the same as those shown in Figure 9), and the red lines and shaded sections are the calculated glare constraints (as per Figure 11). The point shown as a dark blue star is the optimum point - the point that produces the highest average desktop illuminance while staying within the constraints. The light blue dots show the points that were tested by the algorithm: note that not very many of them are required to find the optimal point.

Example results for the 16 illustrative conditions are shown in Figure 13. As expected, the optimal points are always to be found on the glare constraint lines: in order to increase illuminance, the blinds will always tend towards greater openness, until they run into the glare constraint.

[3] In the case where the glare constraint is not considered, the optimization problem definition is the same minus the constraint lines. (But note in this case that the solution is trivial, since it will always move towards the (OP,OP) point.)

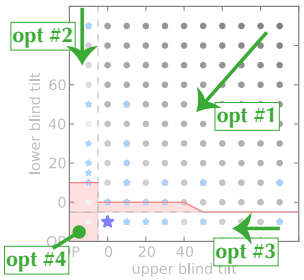


Figure 12: Illustration of the optimization process

The optimization process is repeated for the 2450 points shown in Table 2. For each set of solar conditions being considered, the optimization process finds a single 2-dimensional optimal point (u_1, u_2). These points are then placed in a lookup table and used with nearest neighbor interpolation to determine the approximate optimal control response for any given set of conditions. Some of the calculated optimum points (in dark blue), with nearest-neighbor interpolation between them (in black), are shown in Figure 14.

Table 2: Conditions grid for illuminance maximization optimization

	range	number of points
day of year	June 21 - Dec 21	7
hour of day	6 - 18	7
direct normal radiation (W/m^2)	100 - 1000	10
diffuse horizontal radiation (W/m^2)	100 - 500	5
total number of points		2450

4.3. Control Objective Option #2: Energy Minimization

4.3.1. Mathematical Definition

For the control objective option of minimizing the combined HVAC and lighting power consumption, the optimal control problem for any given set of solar conditions is defined in Equation 16, which is analogous to the previous case, except with the energy model used in the place of the illuminance model.[4] (Note that w_5, outdoor temperature, appears here but not in the previous case.)

$$\begin{aligned} \min_{u_1,u_2} \quad & \text{energy}(u_1, u_2, w_1, w_2, w_3, w_4, w_5) \\ \text{s.t.} \quad & \text{glare}(u_1, u_2, w_1, w_2, w_3, w_4) \leq 0.42 \\ & u_1, u_2 \in [-88, 88] \cup [OP] \end{aligned} \tag{16}$$

As before, with the glare constraints made explicit, the case where the blind tilt angle is restricted to horizontal and downwards ($\gamma = 0$) is shown in Equation 17.

[4] Note that again, if the glare constraint is not considered, the optimization problem is the same minus the constraint lines. However, in this case, the optimal solution is not trivial.

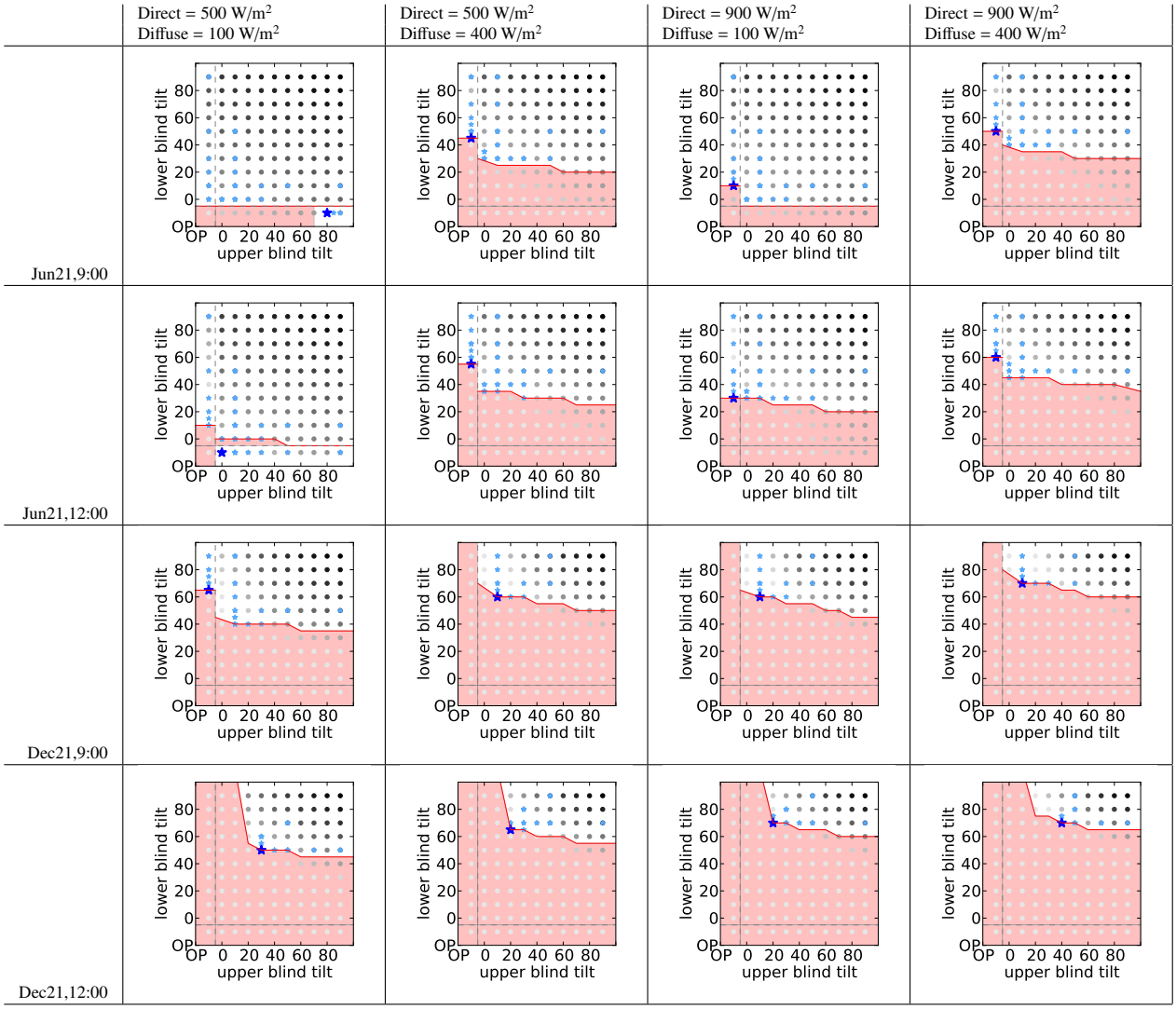


Figure 13: Illuminance, constraints, optimization process and optima

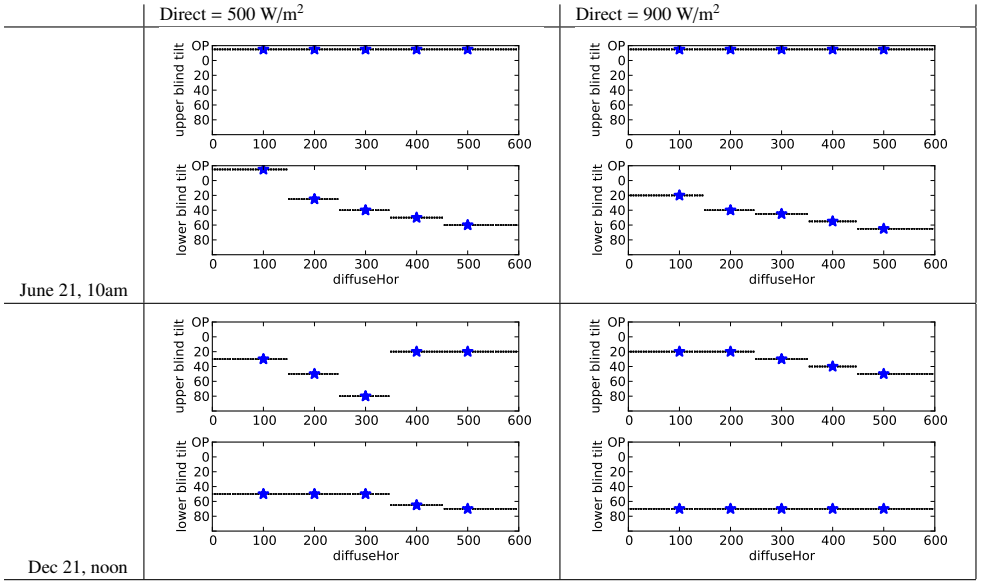


Figure 14: Optimal control set points from a more refined lookup table, with nearest-neighbor interpolation

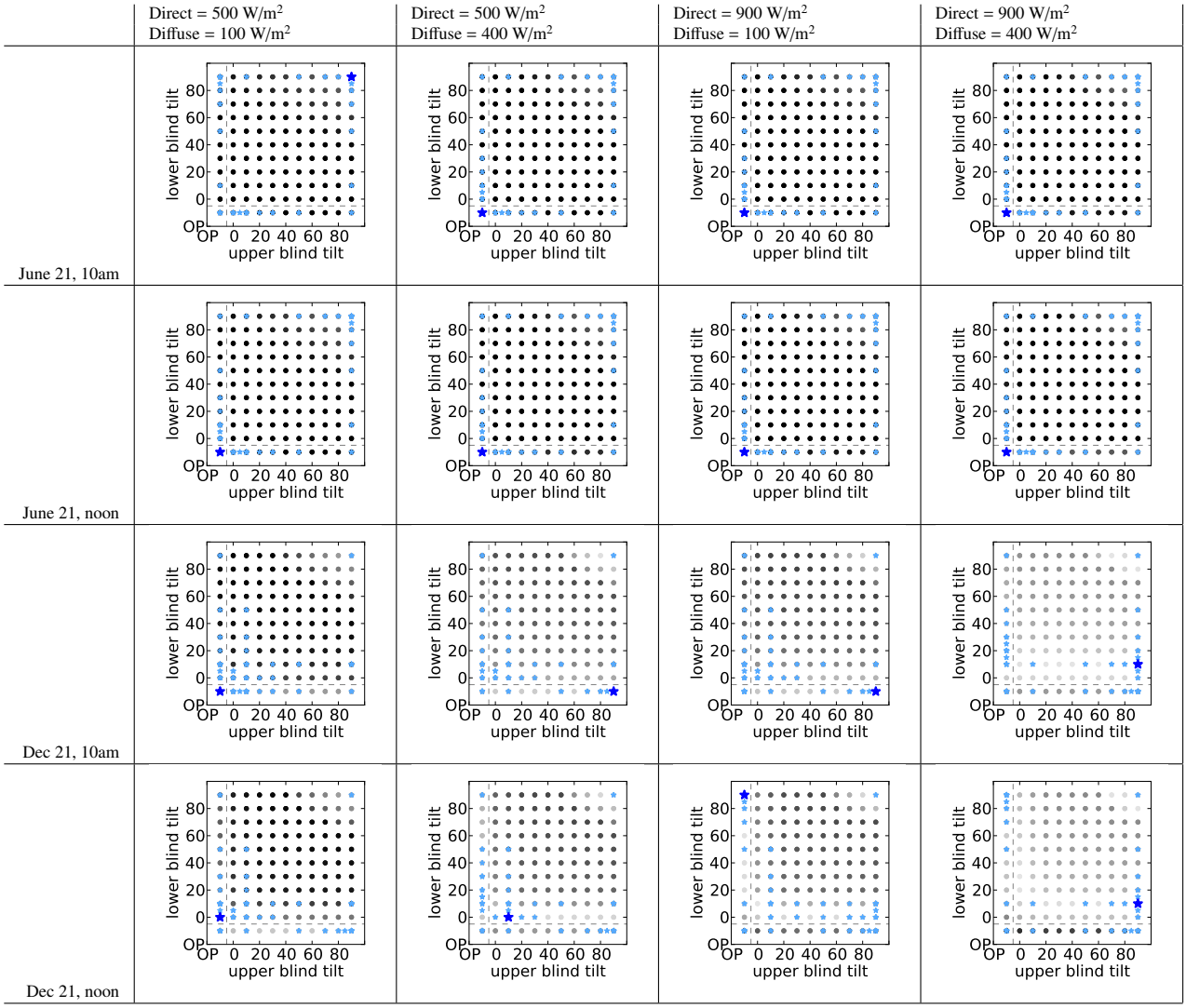


Figure 15: Minimization of total (HVAC + lighting) power required as function of shade positions, outdoor temp. = 5C

$$\min_{u_1,u_2} \ \text{energy}(u_1, u_2, w_1, w_2, w_3, w_4, w_5) \tag{17}$$
$$\text{s.t.} \quad (u_2 \leq \ \text{constraintLine1}(u_1, w_1, w_2, w_3, w_4)) \ \text{OR}$$
$$(u_2 \geq \ \text{constraintLine2}(u_1, w_1, w_2, w_3, w_4)),$$
$$\forall u_1, u_1, u_2 \in [0, 90] \cup [OP]$$

4.3.2. *Illustrative Case: Minimizing energy without glare constraints*

As in the previous section, Figure 15 shows how the overall optimization comes together, but in this case without the glare constraint option used. The underlying black-grey-white dots are the calculated power consumption values. The light blue dots show the points that were tested by the optimization algorithm. The point shown as a dark blue star is the optimum point.

The optimization process is repeated for the 17150 (=2450*7) points shown in Table 3. Some of the calculated optimum points (in dark blue), with nearest-neighbor interpolation between them (in black), are shown in Figure 16.

Table 3: Conditions grid for illuminance maximization optimization

	range	number of points
day of year	June 21 - Dec 21	7
hour of day	6 - 18	7
direct normal radiation (W/m^2)	100 - 1000	10
diffuse horizontal radiation (W/m^2)	100 - 500	5
ambient temperature ($^\circ C$)	0 - 30	7
total number of points		17150

5. Performance of Controllers in Simulation

5.1. Overview

For comparison with the physical implementation studies described in the next section, the simulation studies were based on the Advanced Windows Testbed Facility (LBNL Building 71T). And as with the physical studies, three different controllers were tested: two different optimization-based controllers were

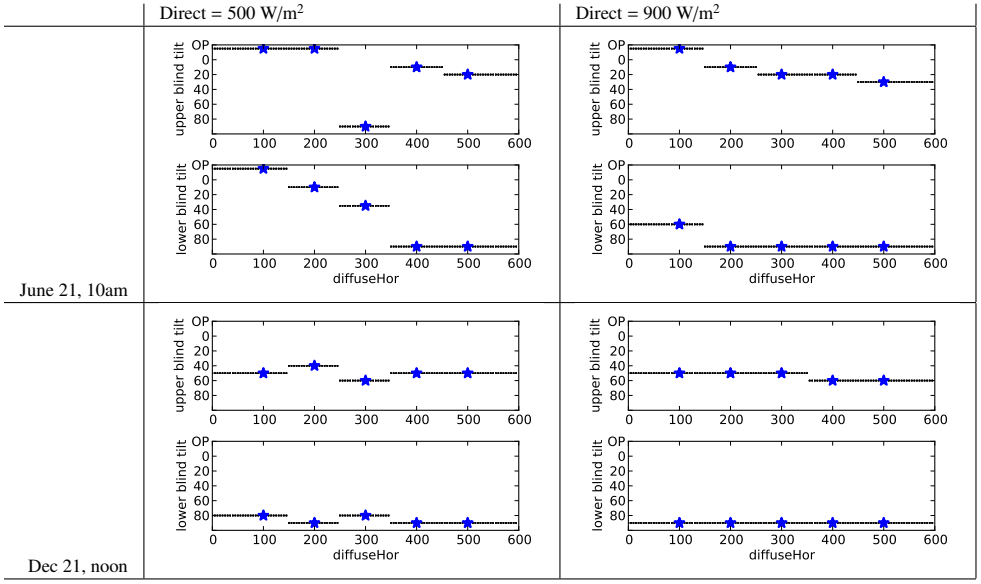


Figure 16: Optimal control set points, with nearest-neighbor interpolation (temp=20C)

tested, alongside a simple heuristic controller for comparison, as described below. The simulations were carried out for sampled days over a full year, using measured weather data from the Testbed Facility.

5.2. Descriptions of the Three Controllers Tested

Baseline Heuristic Controller, 'heuristic'

As a baseline for comparison, a simple 'block beam' algorithm was implemented, based on the controller 'VB-E1n-auton1' description on page 35 of (Lee and Selkowitz, 2009): when the outdoor brightness exceeds 40,000 lux, the blind angle is set to the minimum required to entirely block the direct beam (a geometric calculation); when the outdoor brightness is less than 40,000 lux, the blind angle is kept at horizontal. The idea of this heuristic control logic is to allow maximum daylight admittance (without fully retracting the blinds) when it is cloudy, but to block the direct beam when it is sunny.

Optimization-Based Controller #1, 'illumMax'

The construction of this controller by the cloud-based automated tool is described above. The objective was set to maximize illuminance. The glare constraint is used and set to DGP $\leq$ 0.28.[5] The horizontal-and-below-only constraint was also used.

Optimization-Based Controller #2, 'energyMin' The construction of this controller is also described above. The objective was set to minimize energy consumption. No glare constraint was used in its construction. The horizontal-and-below-only constraint was used.

[5]Earlier tests highlighted a simple and potentially very useful improvement to the controllers: the optimizations should use DGP and desktop-lux thresholds that are slightly inside of their proper values, so that when the interpolation-based controller misses by a little bit, it does not go over the threshold.

5.3. Simulation Results: Cumulative Comparisons

Figure 17 and Table 4 describe the relative performance of the three controllers, with respect to glare, desktop illuminance and energy consumption, for the simulations using measured weather data sampled over the full year.

Note that the 'illumMax' vastly outperforms the 'heuristic' controller in terms of glare control while still providing adequate daylighting (including more daylight than the 'heuristic' controller under cases of limited outdoor light): it provides a 79% reduction in the incidence of DGP > 0.30.

Also note that the 'energyMin' controller provides significantly better energy performance than does the 'heuristic' controller. (The 'illumMax' controller also provides good energy performance because it is often blocking potentially high solar gains, but note that the 'energyMin' controller does outperform it on that metric.) The annual lighting plus HVAC energy use for this one perimeter zone is reduced by 18%.

6. Example Controller Implementations at the Advanced Windows Testbed Facility

6.1. Overview

The Advanced Windows Testbed Facility (LBNL Building 71T) was used as a testing ground for the controllers being developed. Room B was configured with a two-zone external Venetian blind system. The same optimization-based controllers and simple heuristic controller were tested as were used in the simulation tests, as described above. In all three control cases, the control logic was embedded on a Raspberry Pi that was connected to the sensors and actuators at 71T via the http interface that had been previously set up for such tests. The Raspberry Pi reads the sensor readings and sends the control setpoints for the blind positions every 15 minutes.

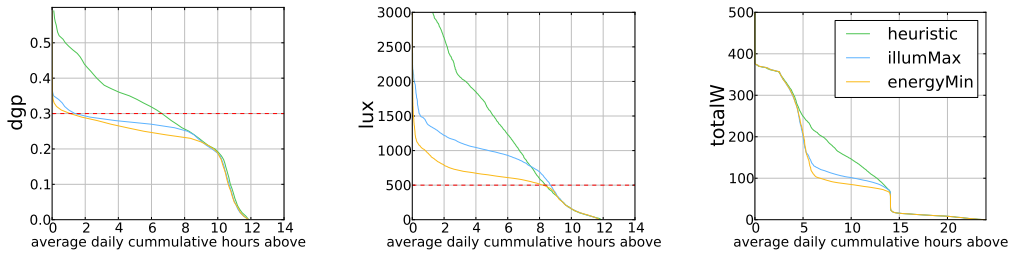


Figure 17: Cumulative results from simulations

Table 4: Summary results for annual simulations

	absolute values			% better than heuristic	
	heuristic	illumMax	energyMin	illumMax	energyMin
ave hrs/day with DGP ≥ 0.30	6.63	1.38	1.09	**79.2%**	83.6%
ave hrs/day with DGP ≥ 0.35	4.67	0.09	0.04	98.0%	99.2%
ave hrs/day with daylight ≥ 500 lux	8.38	8.68	8.24	**3.5%**	-1.7%
ave illum during daylight hours (lux)	1403	834	573	-40.6%	-59.2%
ave lighting + HVAC power (W)	132.30	113.55	108.03	14.2%	**18.3%**

6.1.1. Testing Configurations: Measurements and Simulations

The experiments were carried out over ten days in late summer. Only one controller could be physically tested at any time, and they were cycled daily (2 days were allotted to the 'heuristic' control, 5 to 'illumMax', and 3 to 'energyMin'). The results of the physical experiments constitute the 'measured data' comparison. It provides a concrete comparison of performance, but suffers from the inability to compare the controllers under identical conditions.

To provide a supplemental way of comparing the controllers' behavior and performance, the three controller implementations were simulated (with the same Radiance and Modelica models described earlier) for all of the days in the experimental period. This allows for a comparison of the three controllers under identical conditions. It also provides a comparison point between the simulation studies shown in the previous section and the experimental results shown here.

6.2. Experimental and Simulation Results: Example Details

Figure 18 provides an example of one day of detailed results. Note that on this day, the 'heuristic' control option was being physically tested. Note that the behaviors of the 'illumMax' and 'energyMin' controllers make sense in the light of their objectives. The 'illumMax' controller keeps the modeled DGP just below 0.30 and still provides around 1000 lux on the workplane. The 'energyMin' controller, on the other hand, maintains a lower desktop illuminance - it keeps it effectively at 500 lux through the middle of the day, just enough to avoid having the automated lights turn on, and otherwise blocking as much solar heat gain as possible because the exterior zone is in cooling mode - and has the least modeled energy use of the three. In contrast to these, the 'heuristic' controller does very poorly on the modeled DGP metric, and uses significantly more energy.

6.3. Experimental and Simulation Results: Cumulative Comparisons

Figure 19 and Table 5 describe the relative performance of the three controllers, with respect to glare, desktop illuminance and energy consumption, for the two types of tests: measured data from the experimental period, and simulations using the same weather data as the experimental period.

Note that the measured results do not cohere very strongly with the simulated results. As discussed in the next section, more attention needs to be given to model calibration in future work. And the small number of days with cycled implementations does not lend itself to very robust comparisons between the controllers' performance in the 'measured data' case.

The experimental implementations did provide an opportunity to demonstrate implementation feasibility, to make sure that we had worked through the details necessary to make it operational, and to better understand what needs further work. And the control behavior for the optimization-based controllers were generally reasonable and sensible.

7. Discussion

7.1. Possible Improvements to Models and their Calibration

As the physical experimental results highlight, the models do not match reality as closely as they should. The Modelica models were roughly calibrated, but not to a great precision, and the Radiance models were taken as given: more systemic calibration of the models would almost certainly improve the performance of the controllers in physical implementation.

However, in the overall context of the web-based controller-producer, the idea of calibration leads to some tricky questions that still need to be worked out. Should the user be asked to upload historical data for the calibration process? Should the controller hardware collect the necessary data itself (what then are the necessary connections to other sensors?), and after some

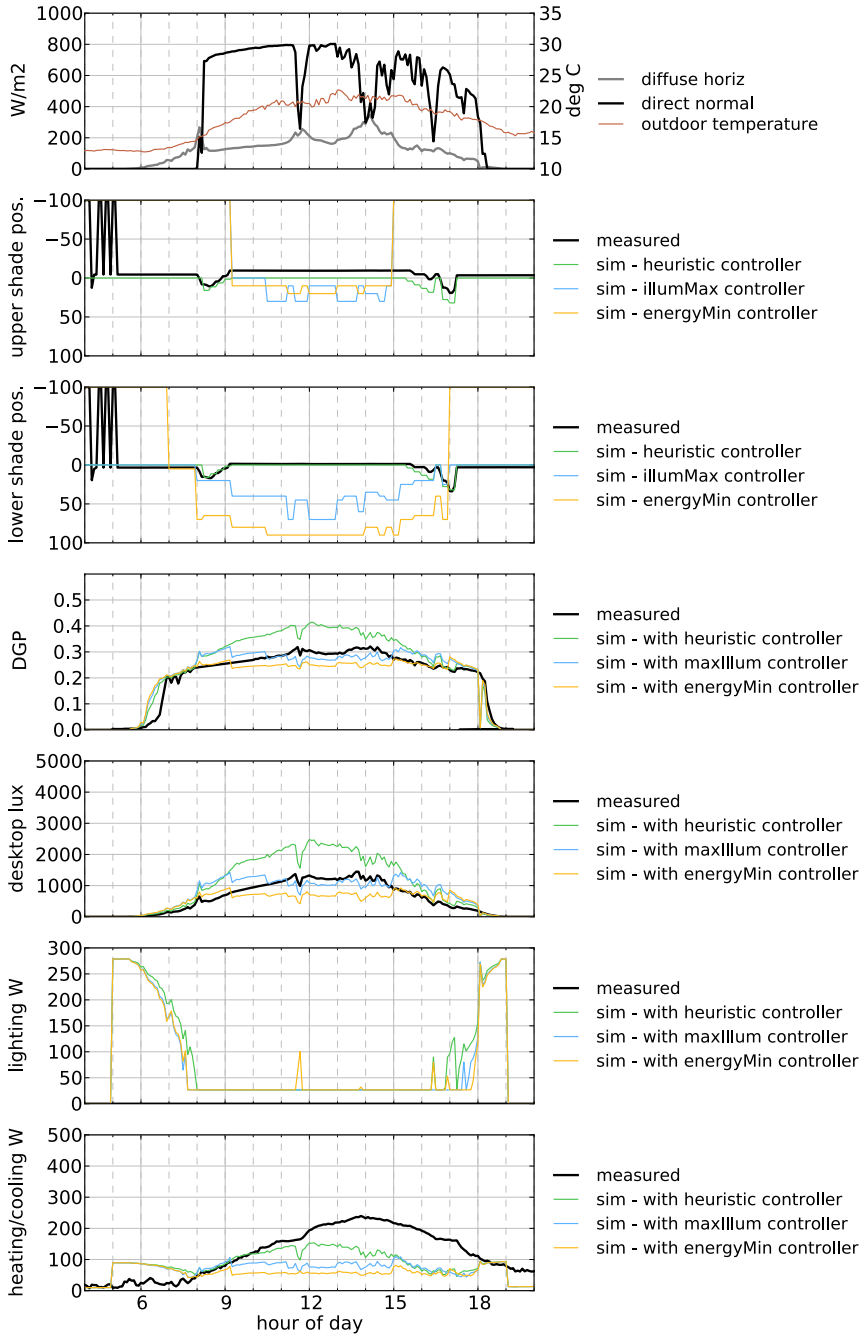


Figure 18: Experimental result details, representative day ('heuristic' controller running)

Table 5: Summary results

		absolute values			% better than heuristic	
		heuristic	illumMax	energyMin	illumMax	energyMin
measured data	**ave hrs/day with DGP ≥ 0.30**	0.96	0.70	0.00	**27.0%**	100.0%
	ave hrs/day with DGP ≥ 0.35	0.00	0.00	0.00	0.0%	0.0%
	ave hrs/day with daylight ≥ 500 lux	7.92	8.17	8.81	**3.2%**	11.2%
	ave illum during daylight hours (lux)	440	459	340	4.5%	-22.6%
	ave lighting + HVAC power (W)	119.72	160.20	133.15	-33.8%	**-11.2%**
simulations over measured data period	**ave hrs/day with DGP ≥ 0.30**	7.11	1.03	0.25	**85.6%**	96.5%
	ave hrs/day with DGP ≥ 0.35	4.56	0.00	0.00	100.0%	100.0%
	ave hrs/day with daylight ≥ 500 lux	9.13	9.82	9.16	**7.5%**	0.3%
	ave illum during daylight hours (lux)	1321	854	552	-35.4%	-58.2%
	ave lighting + HVAC power (W)	121.47	102.52	96.09	15.6%	**20.9%**

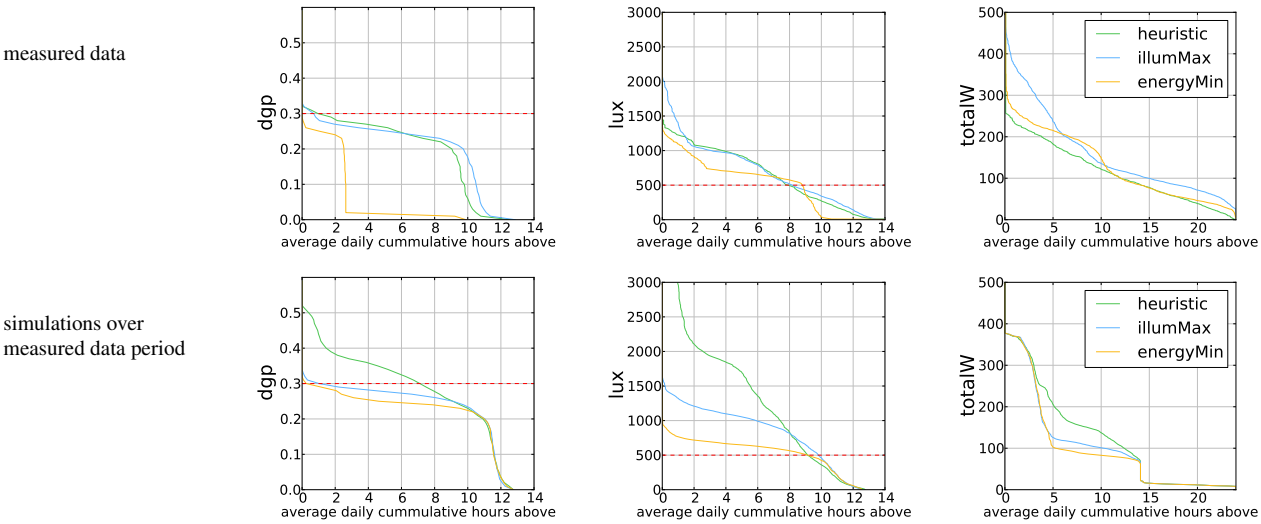


Figure 19: Cumulative results from experiments and simulations

time (and/or periodically) redo the cloud-based calculations to update the control logic using models that are calibrated to the data? Or should the overall idea be revised slightly such that the web-based app produces a starting point for the controller and a learning algorithm that helps it get better with in-situ data?

This is certainly a soft spot in the overall idea described in this report, but it is likely not an insurmountable challenge to find ways of capturing the benefits of the outlined approach while still benefitting from properly calibrated models.

7.2. Possible Improvements to the Optimization, Sampling and Approximation Methods

More work could be done to make the optimization-based controller production computations faster and more efficient. One of the biggest possible time-savers at this stage would be the use of interpolation of glare constraints to reduce the computation time required to calculate those constraint lines. Adaptive sampling methods, rather than the simple regular 'conditions grid', may also produce more accurate and/or more efficient approximations to the MPC without much increase in implementation complexity. Similarly, other controller implementations, beyond the simple lookup-table 1-Nearest-Neighbor interpolation used herein, could provide more nearly-optimal control responses while potentially also decreasing the storage-space and/or computational requirements of the embedded controller.

7.3. Practical Implementation Notes

Note that in its current configuration, the control logic is open-loop; there is no feedback in the system as presently designed. Ideally, the blind positions suggested by this model-based control logic would be used as a starting point (perhaps set every 15 min to 1 hr) that can then be tweaked (perhaps on a 1-minute or 5-minute timescale). For the 'illumMax' case, the blind positions would be moved towards the closed direction if the glare is too high or towards the open direction if the glare is fine. However, this would require some sort of proxy measurement for glare, which is tricky and/or expensive, and so many not be worth it in most cases. For the 'energyMin' case, however, this would certainly be feasible to slightly open or close the blinds based on real-time measured HVAC and lighting energy use.

The 'energyMin' controller is often keeping the blinds nearly fully closed, and the resulting loss of view is likely to bother occupants. Further constraints on the objective function to maintain view could be added to the controller-maker system. In any case, for practical implementation, any of these controllers should be configured with an interface to allow for temporary overrides by occupants. But even with that in place, one should try to set up the control algorithm such that the override is rarely used. (Though, as a side benefit, the history of such overrides could potentially be used as a proxy for measured glare being over the comfort threshold.)

Another implementation option might be to use the 'illumMax' control when an occupant is present, and the 'energyMin' control (modified for the case where the lights are always off) when no occupants are present, again with the option of temporary occupant overrides.

7.4. Possible Extensions

The tool has been set up (and guided by the nature of the dc-timestep calculation method) to allow for one to easily swap out particular components, such as the interior view matrix or the facade system BSDFs, without having to change any of the rest of the calculation procedure. This means that other active facade elements (such as electrochromic windows or other types of shades or glazings) can be added to the list of possible facade elements chosen by the user, as can other types of interior or exterior geometric configurations. It also means that a user could upload their own BSDF set and/or their own external or internal matrices. With some additional functionality added to the tool, the user could even be allowed to simply upload a CAD

file describing the interior and exterior geometry, and the automated system could automatically calculate the matrices before starting the MPC calculations with them.

Other possible extensions to the tool include the consideration of any arbitrary number of zones (instead of only considering the case of two zones, as done so far), and the option of coupling the facade control with that of thermally massive HVAC systems (building on previous work in this project, (Coffey, 2012)).

7.5. Potential Roles in Industry

This project was carried out with the non-expert user in mind. The intent has been to create an app that could be used by a facade system installer or curtainwall assembler, or an architect or design engineer.

One possible use case is that of a shading device company or curtainwall assembly company with a fixed set of components that customers can choose from. The company hires an expert to set up BSDF files for each of the components, and to modify the web-based app to provide their product line as options. The customized version of the app can then be used by salespeople, architects, installers or others, producing control logics for those shading systems that are customized to the specific internal and external geometries of the window and perimeter zone under consideration.

8. Conclusions

A cloud-based automated controller production system has been set up for a motorized external Venetian blind device, with a simple web interface that can be used by non-experts. The computation cost per controller is in the range of a few dollars. The computation time and expense can be decreased with some improvements. The control logic is simple enough to be implemented on small and cheap distributed controllers.

Annual simulation experiments suggest that the 'illumMax' controller (illuminance maximization subject to glare constraints) can very significantly reduce the occurrence of times with high glare probability (79% less frequently than a 'heuristic' block-beam controller in the tests), while also providing good daylighting performance. The simulation experiments also suggest that the 'energyMin' controller (lighting + HVAC energy minimization) can provide significantly better energy performance, with estimated annual savings of 18% of lighting and HVAC energy use in south-facing perimeter zones, without any increase in controller hardware cost, and hopefully negligible difference in configuration cost, when compared with the 'heuristic' controller.

These two control configuration cases were also physically implemented and tested at the Advanced Windows Testbed Facility. A Raspberry Pi was used to control the blinds using the automatically-produced control logic. The tests demonstrated the technical feasibility of the approach and provided an opportunity to learn what needs greater focus in further work.

The next steps towards market uptake include investigations into how this cloud-based app idea can fit into existing design, implementation and business models. Future research work should include extensions to the system to include other dynamic facade elements and HVAC systems, such as electrochromic windows and thermally massive HVAC systems. This research can also be used to guide further work in related building control application areas.

9. Acknowledgments

This work was supported by the California Energy Commission through its Electric Program Investment Charge (EPIC) Program on behalf of the citizens of California and the Assistant Secretary for Energy Efficiency and Renewable Energy, Building Technologies Program, of the U.S. Department of Energy, under Contract No. DE-AC02-05CH11231.

The authors thank the following experts for reviewing this report: Christoph Gehbauer, Taoning Wang, and David Blum, LBNL.

References

Arasteh, D., Selkowitz, S., Apte, J., and LaFrance, M. Zero energy windows. *ACEEE Summer Study on Energy Efficiency in Buildings*, LBNL-60049, 2006.

Aries, M.B., Veitch, J.A. and Newsham, G.R. Windows, view, and office characteristics predict physical and psychological discomfort. *J. Environmental Psychology 30(4): 533-541*, 2010.

Boyce, P., Hunter, C. and Howlett, O. The benefits of daylight through windows. *Troy, New York: Rensselaer Polytechnic Institute*, 2003.

Coffey, B. Integrated control of operable fenestration systems and thermally massive HVAC systems: Methods and simulation studies of energy savings potential. *DOE / CEC PIER Technical Report*, 2012.

Drgona, J., J. Arroyo, I. C. Figueroa, D. Blum, K. Arendt, et al. All you need to know about model predictive control for buildings. *Annual Reviews in Control 50, 190 - 232*, 2020.

Eichholtz, P., Kok, N. and Quigley, J.M. The Economics of Green Building. *Review of Economics and Statistics, 95(1), 50-63*, 2013.

Gentile, N., Osterhaus, W., et al. Integrating daylighting and lighting in practice: Lessons learned from international case studies. *Technical Report of Subtask D, IEA SHC Task 61 / EBC Annex 77, online: https://task61.iea-shc.org/publications*, accessed October 31, 2021.

IEA. World Energy Outlook 2011. *International Energy Agency, Paris*, 2011.

Lee, E.S., Selkowitz, S.E., et al. High performance building facade solutions. *PIER Final Project Report, CEC-500-06-041*, 2009.

Tan, Z., Zheng, S., Palacios, J. and Hooks, C. Market Adoption of Healthy Buildings in the Office Sector: A Global Study from the Owner's Perspective. *MIT Center for Real Estate Research Paper, (21/07)*, 2021.

Turan, I., Chegut, A., Fink, D. and Reinhart, C. The value of daylight in office spaces. *Building and Environment 168: 106503*, 2020.

Wetter, M. et al. Modelica Buildings Library. *http://simulationresearch.lbl.gov/modelica/ , accessed February 7*, 2018.

Veitch, J.A., Christoffersen, J. and Galasiu, A.D. Daylight and view through residential windows: effects on well-being. *Proceedings of Lux Europa, 12th European Lighting Conference, 2013, Krakow, Poland, pp.1-6*, 2013.

Wienold, J. Dynamic daylight glare evaluation. *Building Simulation, Eleventh International IBPSA Conference, Glasgow*, 2009.