

Approaching Sentient Building Performance Simulation Systems

Negendahl, Kristoffer; Perkov, Thomas; Heller, Alfred

Published in:
Proceedings of eCAADe 2014

Publication date:
2014

Document Version
Early version, also known as pre-print

[Link back to DTU Orbit](#)

Citation (APA):
Negendahl, K., Perkov, T., & Heller, A. (2014). Approaching Sentient Building Performance Simulation Systems. In E. M. Thompson (Ed.), Proceedings of eCAADe 2014 (Vol. 2, pp. 49-60)

DTU Library

Technical Information Center of Denmark

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Approaching Sentient Building Performance Simulation Systems

Kristoffer Negendahl¹, Thomas Perkov², Alfred Heller³

^{1,2,3} Technical University of Denmark, Department of Civil Engineering

^{1,2,3} <http://www.dtu.dk>

^{1,2} {krnj\alfh}@byg.dtu.dk ³ s103520@student.dtu.dk

Building designers make decisions in early design stages that have large impact on building performance, including those of energy-, daylight- and indoor environment performance. Building performance simulation (BPS) tools can support the designer, in making better decisions, by providing the performance consequences of design choices. However BPS tools often require deep technical knowledge and is too time consuming to use to effectively support the design exploration in the early design stages. To solve this challenge, the current paper proposes: Sentient building performance simulation systems, which combine one or more high precision BPS tools to provide near instantaneous performance feedback directly in the design tool. Sentient BPS systems are essentially combining: 1) design tools, 2) parametric tools, 3) BPS tools, 4) dynamic databases 5) interpolation techniques and 6) prediction techniques as a fast and valid simulation system for the early design stage.

Keywords: *Building Performance Simulation, Parametric modelling, Visual Programming Language, Database, Responsive system, Integrated Dynamic Model*

INTRODUCTION

Human intelligence is superior in developing abstraction, creativity and imagination while computers are superior in calculation, data analysis, and information retrieval.

Let computers handle the tactics, setting up simulations of multiple solutions, analyzing results and showing the consequences, while humans handle the strategy.

This is the fundamental idea behind the game of Advanced Chess [2]. Advanced chess is a human-computer symbiotic partnership that demonstrates a

human with a computer could be far superior to either a human alone or a computer alone.

While a computer is significantly more intelligent when it comes to chess tactics, a human is significantly more intelligent when it comes to strategy. This is the case for building design, yet we failed to employ the tactical skills of computers to support us in our strategy of designing buildings.

Based on this assumption, a computer supported *building performance prediction and decision making* system will be suggested. A prototype implementation of the system, focused on daylight per-

formance feedback, is employed to explain the proposed concept of *sentient building performance simulation systems*. Due to the narrow time frame affiliated with early design stages combined with various challenges of integrating BPS tools in building design, a sentient BPS system has to be optimized by different means, which is presented and discussed in this paper.

BACKGROUND

Building design is done on basis of geometrical representations in design (CAD) tools while performance evaluation is carried out aided by building performance simulation (BPS) tools. The actual stage is to get these tools to work together in an integral system. With the introduction Building Information Modelling (BIM) and visual programming languages (VPL), the integration of design tools and BPS tools, at model level, has improved significantly (Negendahl, 2013). This tendency is strongly implemented in the design tool Rhino [8] and belonging visual programming language (VPL) Grasshopper [3]. Grasshopper coupled to a BPS (e.g. DIVA [9] and/or Energy+ [4]) have the ability of strategic scripted parameter variations in user defined models. This reduces simulation time dramatically, as each new design proposal is automatically simulated by the runtime coupled BPS tool.

Additionally the introduction of VPLs has changed the way engineers and architects think of building design. The parametric capabilities have generated everything from architectural manifests to multi criteria optimization methodologies. Most importantly the concept of parametric models is giving building designers the "tool" that matches their continuously altering idea of a building in the early design stage. Combined with BPS, VPLs are capable of assisting and informing the building designer in every thinkable building related performance.

While BPSs, VPLs and design tools are in a process of unification, the road to full integration is still far ahead. It is today highly improbable that a building designer can handle every aspect of building per-

formance evaluation in one go: Designing with structural optimization in mind, contemplating life cycle assessments, balancing building energy with indoor environment, etc. But why shouldn't the building designer do this? Why is the designer limited to the design tool when all this "knowledge of performance" is out there to just be simulated, evaluated and taken into consideration? This article describes a way to approach some of the primary obstacles in the merging of the design tool with the BPS by describing an implementation of a simple sentient BPS system.

EXISTING SYSTEMS LACKS RESPONSIVENESS OR PRECISION

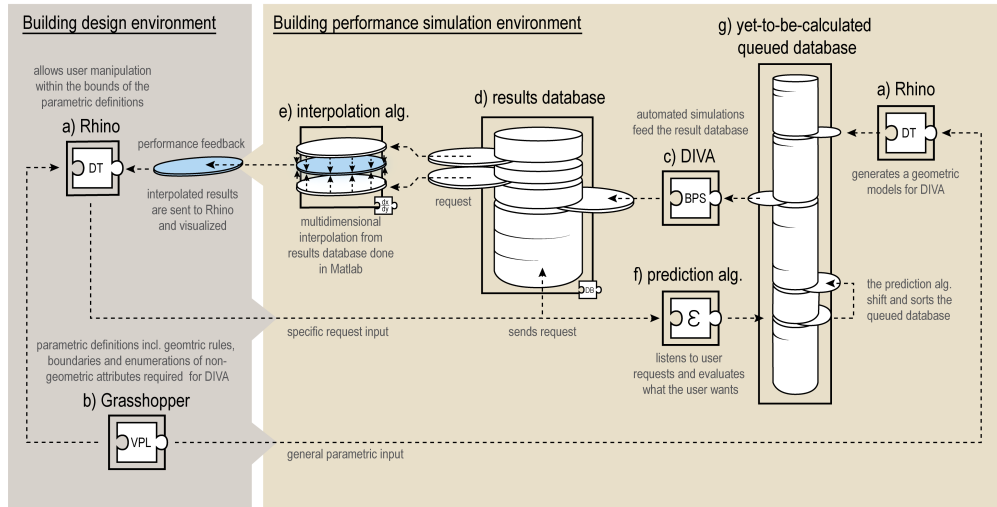
Two different approaches of linking BPS with design tools are dominating. The first approach is coupling highly detailed and complicated BPS environments to the design tools. These systems may be able to calculate the performance to a very precise degree, well beyond the information level of a building design in its conceptual stages. In general, these BPS tools needs large computing capacity and will take long time to simulate. The first approach is of this reason often much slower than the second approach and in some instances such system will block the dynamics of the design process.

The second most dominating approach seek to maximize responsiveness by either linking simplified BPS or implementing user defined scripts acting as BPS (Klitgaard et al. 2006). Ideally the right implementation and powerful computing power will allow super-responsive live performance feedback from the BPS. This approach lacks precision and may in worst case make performance evaluations on incorrect assumptions that again can lead to the very opposite of an improved building performance.

VALIDATED TOOLS, VALIDATED INPUT DATA, VALIDATED USERS

Souza (Bleil de Souza, 2012) argues that validity of modeling and calculation assumptions depends not only on the level of competency but also on the purpose of modeling. In this sense, a good model depends enormously on the experience of the modeler,

Figure 1
Implementation of a simple sentient BPS system. Rhino a) is dynamically coupled through Grasshopper b) to DIVA c) over an intermediate results database d) with interpolation features e). A prediction algorithm f) shifts and sorts yet-to-be-calculated queued data g) before the DIVA receives it.



which comes from practical knowledge and contextual understanding of the subject in order to solve similar problems. Valid operation and BPS tool input requires competent simulation experts or "simulationists" as Souza calls them.

INTRODUCING SENTIENT BUILDING PERFORMANCE SIMULATION SYSTEMS

Sentient, also meaning *conscious* and *responsive* expresses an almost - human-like behavior. However interesting (and frightening) an awakening consciousness in our computer companions are, the ability of a machine to respond to building designers demands is what is important to us in this article. Sentient BPS systems are suggested as a highly responsive alternative to building designers who are either using simple proximate BPS tools or complicated but slow BPS tools in early design exploration.

The sentient BPS system is based on parametric modeling procedures, which decreases the decision space into a finite size. The system utilizes a database structure combined with a multivariate interpolation algorithm that makes it feasible to sim-

ulate less solutions and still provide the building designer with fast and precise results (from one or more building BPS tools). The idea is essentially to construct a **result database** containing building performance feedback data needed to accompany the designer's own solutions. The system further reduces the number of solutions needed to be simulated, as it observes user activity and adjusts the BPS tool to simulate and improve interpolation precision. To effectively do this, the system attempts to **predict the space of interest** of the building designer while utilizing multivariate **interpolation capabilities of the system**. Essentially the system presents building performance feedback of solutions that is of interest to the designer for decision making in the early design stages, in a very efficient way.

Sentient BPS systems are built by recognizing the need to separate the building designer and the simulationist in the early design stages. The system detaches the complexity of the BPS environment from the building designer. The building designers requests (design solutions) for performance evaluation are sent to a separated (web) performance simulation environment containing a results database, see

Figure 1. The feedback is visualized, or otherwise handled directly in the building designers' own environment, Rhino [8], in a way that fits the building designer. The simulationist will take part in the system by creating the object relations and (parametric) variables necessary to get meaningful results from the BPS. During the design process, the simulationist only job is to maintain the building performance environment (see Figure 1). The sentient BPS system can therefore support the validity of performance feedback required for any building project.

Sentient BPS systems can combine one or more high precision BPS tools and provide near instantaneous performance feedback directly in the design tool, hence providing the best of both worlds; speed and precision. The concept of sentient BPS systems is based on a further development of a student project performed by Perkov (Perkov, 2014) at the Technical University of Denmark.

THE RESULTS DATABASE

It was first suggested by Sullivan (Sullivan et al., 1988) that large number of building energy simulations saved systematically in database could provide fast feedback of energy performance. Such database is capable of giving responsive answers to multiple criteria but required either very large databases or very simple buildings to get meaningful answers. Caldas notes that these kinds of approaches generates data that do "only apply to solutions that are close to those simulated, what's makes them of limited use in an architectural design domain" (Caldas, 2001). Nonetheless, at the time of Sullivan's and Caldas' considerations were written, much development have been done in the field of databases and computing in general. It may still not be feasible to construct universal databases, comprehending every thinkable combination of variables. But it can be feasible to make a finite subset of solutions as a database lookup that takes a very specific design concept into consideration.

When constructing a database of solutions, size matters. To grasp the scale of multidimensional re-

sult databases, consider a room with three variables: its height h , its width w , and its depth d . The variables each have *variable resolution* of 10, meaning the variables may be defined in 10 different unique states. The total amount of combinations, c of solutions that need to be simulated adds up to:

$$c(r_w, r_h, r_d) = 10^3 = 1000 \text{ combinations} \quad (1)$$

If each simulation takes 5 minutes in average to simulate with a BPS tool, it will take 3.5 days to construct a result database. Now imagine we add two more variables to the equation, again each with a variable resolution of 10. We end up with almost a year of simulation time to construct the results database, which is not feasible.

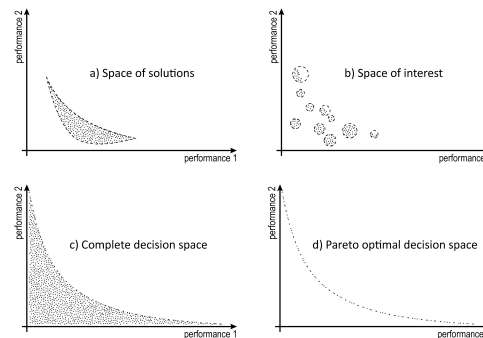


Figure 2
Decision space:
Here illustrated in relation to two performance metrics, each dot represents a specific solution. Space of solutions a) is defined by all solutions that conform to the requirement of a certain performance criteria, here performance 1 and 2. Space of interest b) is defined by the building designers' interest in certain solutions related or unrelated to the performance criteria 1 and 2. The complete and pareto optimal decision spaces c), d) are shown for comparative purposes.

The solution is to limit the simulations to the problems that actually are worth investigating. Seen from a design point of view the *decision space is unlimited*. But since computer power and time are limited factors (particularly in early design stages), how and where does the designer limit the decision space?

The process of limiting the decision space can be separated into two different system approaches on designing with performance as seen in Figure 2:

- A system that supports a *space of solutions*, Figure 2a)
- A system that supports a *space of interest*, Figure 2b)

A system that finds the "space of solutions" is a sys-

tem which seek to aid the designer to find solutions that complies with predefined performance criteria e.g. annual building energy consumption (Petersen, 2011). The designer usually in one way or another "pick out" a specific solution from an enumerated list of permitted solutions within the space of solutions. The system is always limited of predefined performance criteria.

A system supporting the deductive search towards solutions: the "space of interest", allows the designer search through the solutions that may or may not comply with certain performance criteria. Typical design tools (e.g. Rhino) as well as parametric tools (e.g. Grasshopper) support deductive search of interest, usually with focus on the geometrical representation of layout, functions, visual appearances etc. Criteria of these types of qualitative objectives may be unknown to the designer until the designer suddenly *uncover* a solution that fits in a greater holistic whole. *Through a web of moves, designers discover the consequences, implications, appreciations and further moves. Within these moves, phenomena are understood, problems are solved and opportunities are exploited.* (Souza. 2012) A system based on the space of interest is therefore limited by predefined interest criteria. The real question is, when adding a BPS tool to any of the two system approaches, how does the designer use the BPS tools, or with a simulationist involved, how does the designer and the simulationist use the BPS tools in the system?

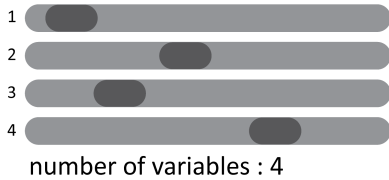
It surely should not be the BPS tool, or the assisting simulationist, that defines the design direction, but the designers own choices in *what is worth investigating*. In this regard, the sentient BPS system aligns itself with the approach of space of interest. However, sentient BPS systems may be used to narrow down the decision space by utilizing predefined performance criteria as required by the space of solutions framework. The sentient BPS system employs the parametric capabilities of a VPL to define the *space of interest*, thus narrowing down the open design problem into a smaller finite decision space. Aided by a VPL, this can be done in numerous ways:

- The building designer (and simulationist) may focus purely on an (optimization of) expected performance.
- The building designer and the simulationist may define a coordinated reduction of the decision space, seeking to advance various performance related and unrelated objectives.
- The building designer (and simulationist) may choose to setup a parametric model on the sole purpose of finding a particular desired geometric form and use the BPS results to validate the geometry as "good enough".

In the prototype sentient system discussed in this article, the simulationist and the building designer are collectively reducing the decision space by employing parametric model scripted in Grasshopper [3]. The only clearly defined objective is to improve daylight factor conditions in a room model. Other objectives such as aesthetics, layout and qualitative use of daylight is unknown to the users in the beginning of the modeling process, however, aided by the sentient BPS system the objectives becomes apparent during the design exploration with the parametric model. There are no criteria or predefined rules, others than the limitations based on the implemented parametric definitions and the parametric boundaries in the variables used in the model.

PREDICTING THE SPACE OF INTEREST

Predictions of building designer interest is a rather unexplored subject while predictions of the (space of) solutions has been thoroughly investigated e.g. by (Pedersen, 2006; Shi & Yang, 2013). Framing the space of solutions is defined by very accurately defined objectives, and in terms of building performance, the objectives have to be defined in a way that BPS tools can understand. Predicting the user interest is very different, simply because the user often does not know what he or she is interested in to begin with. The objective is an exploration in itself why



weight function s
sum of weights on every
state in a sequence based
on primes

weight function t
variable listener function

weight function w
variable state overrule function

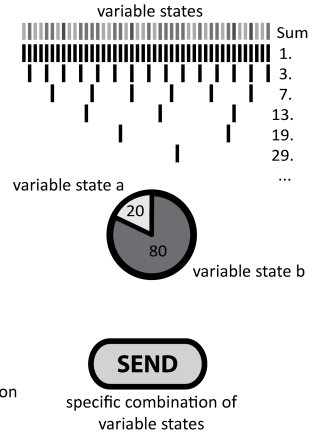


Figure 3
The prediction algorithm uses variable states (here represented on sliders). The slider in the bottom left shows a variable resolution of 7 and its current state is 5. The three weight functions are illustrated to the right.

objectives are likely to be unclear and fuzzy. The concept is to utilize embedded information of the parametric variables present in the model. The amount of variables and their *resolutions* will define the amount of unique combinations in the model as it was mentioned earlier.

A variable resolution is the amount of unique states a given (parametric) variable has. The variable is an enumeration of numbers, which does not need to be sequential or based on integers. An example is shown in Figure 3; let a variable be represented by a slider that can be set in state [1..7], the variable resolution of the slider is 7. A system may have more than one slider or any other collection function of variables (navigational controllers, lists, arrays, etc.). Each individual variable has a individual variable resolution. We now introduce a concept of *variable resolution levels* to further reduce the amount of solutions needed to be simulated. The idea is to make precise performance simulations on strategically selected solutions within the space of interest, then estimate the rest of space of interest with minimum amount of errors. The *variable resolution level* of any given sentient BPS system is basically all the unique combinations of every variable states divided by the number of fin-

ished simulations (per coupled BPS tool), defined as follows:

Variable resolution level; has a number of variables $v > 1$ where each variable resolution $r > 0$, for every resolution r in the sequence i of variables v_i :

$$\frac{r_{v1} \cdot r_{v2} \cdot \dots \cdot r_{vi}}{\sum (\text{number of simulations completed})} \quad (2)$$

Essentially the variable resolution level indicate how much of the space of interest have been covered by simulated results. A high variable resolution level means few simulations is completed by the coupled BPS tool (in relation to the total number of potential solutions), while a variable resolution level = 1 means every possible variable state combination has been simulated. As it follows, the number of variables and their resolution will affect the variable resolution level quite substantially. An ideal model will have a minimum required number of variables each with lowest possible variable resolutions to cover the space of interest quickly in the design process. Minimizing variables and resolutions, however can be rather difficult when the building designer have not yet decided all the design objectives. Of this reason an *interest prediction algorithm* has been imple-

mented, hence to further reduce the needed simulations, to cover the actual interest space within the boundaries of the defined variables and their resolutions.

An interest prediction algorithm is implemented on the basis of a continuous weight factorization of the yet-to-be-simulated unique data combinations. There are basically three weight functions in the prediction algorithm; s , t , w (shown in the right side of Figure 3).

Weight functions s , t and w

Weight-function, s is given to all variable states but distributed flat out semi-random by utilizing a sequence of primes. Imagine all states is distributed in a sequence, where every third state is weighted less than every fifth, every fifth is weighted less than the seventh etc. In this way the system is set in a progressive loading-state that helps to get "rough" and faster interpolations "evenly" distributed over the entire the space of interest. The idea with this function is to gradually improve the overall distribution of simulations, in the space of interest, by an incremental expansion variable resolutions.

Weight-function, t is a *variable listener function*, which essentially is a timer function that reads the particular variable states of navigational controllers (aka. sliders) embedded in Grasshopper. Basically the listener function identifies the state of every variable and how long time it remains in that state. The highest weight is given to the variable with the fewest alterations, which arguably must be the preferred state of interest of that particular variable.

Weight-function, w is a simple overwrite-function that favorably alters the weight of a given state on a given variable. It gives the building designer an option to alter a specific request (design solution) to become more important than all other requests queued for simulation. The exact combinations of variable states are sent directly to the BPS to perform an analysis based on that specific request.

weight s_j , t_j , w_j $[0..1]$, as follows: For every

variable state in the sequence j :

$$\text{weight}(j) = s_j + t_j + w_j \quad (3)$$

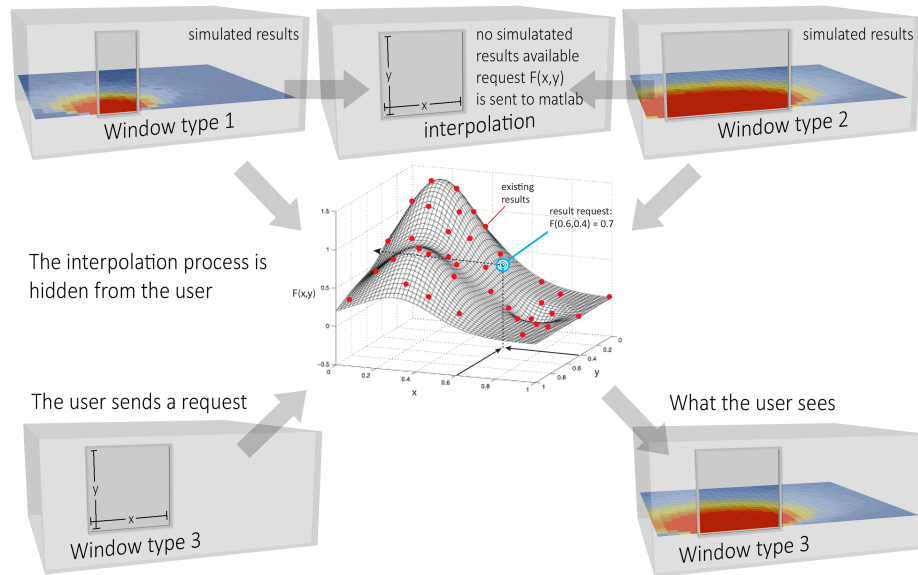
The sum of the weights of each parameter is continuously updated while the building designer uses the system. As seen in Figure 1 g) the effect of the prediction algorithm is a reordering of the yet-to-be-calculated queued database. The consequence of using the prediction algorithm is a more efficient use of simulation power, as the requests from the building designer is automatically taken into account.

INTERPOLATION CAPABILITIES OF THE SYSTEM

Well before the results database is complete, it is possible to interpolate results by using multivariate interpolation techniques. In the prototype implementation seen in Figure 1, the choice was to use the GridDataN from the native Matlab library. GridDataN fits a hyper-surface of the form $y = F(x)$ to the data in non-uniformly-spaced vectors. When a request is sent to the interpolation algorithm it will automatically perform a linear interpolation between the continuous and sequential simulations in the results database.

To ensure that interpolation is not performed over discontinuities the user has to state which of the input variables are continuous and discrete, thus sending compliant data to the interpolation algorithm. The interpolation feature is simply a method to give fast feedback of the results "yet-to-be-simulated". While it is strictly not required in the sentient BPS system, the interpolation algorithm can give the building designer faster and more detailed feedback with fewer simulations, but it also introduces an element of uncertainty into the system.

As seen in Figure 4 the user may get fast feedback from an incomplete results database. In the prototype implementation the performance metrics is daylight factors, where interpolation was calculated over each measurement-point in a grid (648 individual points corresponds to 648 individual results). This means every time the user sends a request with a new set of variable states (in Figure 4, two dimensions, x



The interpolation process is hidden from the user

The user sends a request

What the user sees

and y , representing width and height of a window), the user asks the system to look for 648 individual results from the database. If the corresponding combination of x and y is absent in the database, the system sends the request over to Matlab. GridDataN then constructs 648 individual hyper-surfaces corresponding to the number of measurement points (and not the amount of simulations already performed). Each individual red dots in Figure 3, however, are representing the separate simulations already performed of the *same measurement point in space*. The interpolations are performed 648 times from on each of these hyper-surfaces. In theory the 648 individual measurement points could be a product of 648 different BPS tools. While the system is utterly scalable, the 648 points are just used as an example to represent the daylight factor distribution in a room, the mind blowing flexibility and scalability of the sys-

tem is hard to describe. However useful and interesting multivariate interpolations are, the very fact humans cannot comprehend higher levels of multidimensional operations, makes interpolations risky to use in practice.

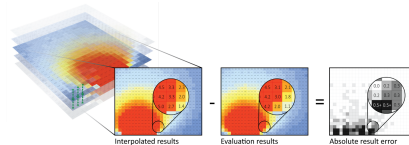
8. DISCUSSION

When focusing on the multivariate interpolation included in the prototype sentient BPS system, the real challenge is to minimize the errors of interpolation. As with any approximate method, the utility of multivariate interpolation cannot be overextended. While various techniques exist in error minimization (e.g. Lagrange multipliers for Kriging (Vapnyarskii, 2010)), we have simply attempted to quantify the errors that can occur in multivariate interpolation. The reason of this, as it follows, is to show where to expect large errors and accordingly seek to avoid them.

Figure 4
The interpolation process of using GridDataN, here showed with a two-dimensional interpolation $F(x,y)$, where x is the width and y the height of a window. F represents the performance metric daylight factor [%] in 648 measurement-points (seen as colors on a plane). The red dots represent a unique product of simulation from DIVA. The blue dot represents the "lookup" procedure handled by the interpolation algorithm.

In Figure 5 is shown the process of estimating the error from GridDataN in the system. The errors are calculated by subtracting the interpolated results with the actual simulated results. To simplify it further the numerical value of this difference is used to estimate the effect of simulation variable resolution levels, equation (2).

Errors of the simple case of the two-dimensional interpolation are showed in figure 6. It is seen in the upper left corner that even two exact simulations vary due to stochastic variations in simulation tool DIVA. Based on two previously discussed variables x, y , each variable has a resolution of 9, which follows a variable resolution level of: $(9 \cdot 9)80 \approx 1$. The interpolation performs reasonably well, however errors tend to accumulate near the window, thus affecting the highest result values. The reason is associated with the inter-dimensional increasing numeric variations in the larger values, which means more simulations may be needed where changes vary much from one solution to the other. Nonetheless estimating daylight factors of 2% or below was reasonable precise even with a fraction of the simulations done.



When a higher resolution level is analyzed $(9 \cdot 9)35 = 2.25$, the errors rises accordingly. From the various test performed with the system, it was found that variable resolution levels should not exceed the level 10, since it generated too great errors to be a useful guide to the designer. Nevertheless much work still need to be done in quantifying a general assumption of error levels, number of parametric variables and variable resolutions. Generalization is further problematic when relationships between variables have strong oscillations or discontinuities. This suggests that, it may be advantageous to attempt to script additional relations between variables, particularly in the case of abstract problems for

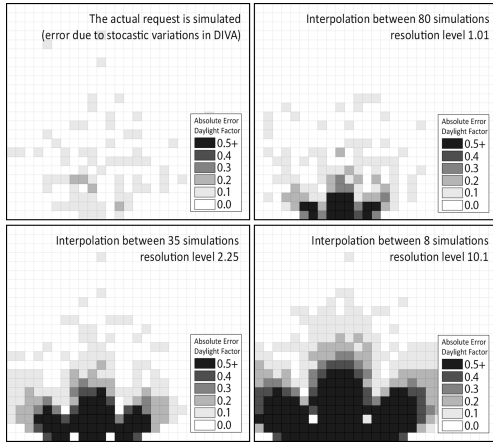
which the topology of the input and output spaces may not be clear a priori, as to produce a relationship which is as smooth as possible.

In the prototype system, the discontinuities of variables have to be identified by the user. In most cases, e.g. variables such as number of windows, type of glazing, and enumerations in general, the user can easily identify the discontinuities, however, in some cases e.g. where geometry "jumps" from a state to another, the discreteness can be difficult to identify.

While only weight-function t autodidact seeks to predict the user interest, weight-function w is more of a service for the designer to validate, or refine his or her own intuition of the building performance. The inclusion of weight-function w was found necessary in providing the user a feel of control over the model. Weight function s was found necessary to include into the model because it helped the interpolation algorithm to get enough data to make the multivariate interpolations distributed more uniformly in the vast space of interest. The function also acted as a continuous generalist refiner for the result database, thus counteracting the weight function t . If the model was left completely unattended the weight function s makes sure every combination of every variable-state is pulled through the coupled BPS tool.

The actual interesting weight-function t is incredibly simple as a concept and in its implementation. The impression that a preferred state is likely to stay the same throughout the design process is quite reasonable, however in many real situations this might not be the case.

Figure 5
The absolute error of interpolation is simply the absolute value of interpolated result subtracted from the actual simulated result.



A user might not alter a variable simply because the variable is of less importance to the user. This could be the color-tone of a window pane or something else of minor interest, in terms of larger building design perspective. While this preferred state of this particular variable does not matter for the user, the system does not know the difference of inter-variable importance. This results in significantly more simulations with e.g. greenish glass, and thus giving the results of the combinations with all other variables based on this color. Greenish glass poses no physical major significance for e.g. thermal performance and building energy consumption. However, if the color green was somehow associated to the window pane g-value, the thermal performance and energy consumption will be very much affected by the choice of state. Of this reason it is suggested in future research to implement a variable importance function.

FURTHER STEPS TOWARDS SENTIENT BPS SYSTEMS

One of the most promising directions towards sentient BPS systems may be found in the fast growing field of machine learning. Of the various directions in research some popular methods are mentioned; neural networks, Gaussian processes, support vector ma-

chines, nearest neighbors, however there are many others. The way the prototype sentient system interpolates between the multidimensional results, and gradually becomes more certain over time, can be compared to many implemented machine learning concepts. However, the proposed prototype system cannot be classified as a machine learning system, of the simple reason that a hypothesis set (training data) is not a necessity for the system to work. Machine learning may very well be used to further improve the prediction of user preferred solutions, thus further narrow the the amount of simulations needed. Nonetheless, the problem with many machine learning algorithms are they often need vast amounts of data to train the system effectively (Mitchell, 2014). This needs to be addressed in future sentient BPS system builds.

Regression is concerned with modelling the relationship between variables, but unlike interpolation methods, regression does not need a continuous stream of results data to function (although, regressions are often constructed on vast amounts of empiric data). Regression can iteratively be refined by using a measure of error in the predictions made by the model. Regression methods are a work horse of statistics and have been cooped into statistical machine learning and in future sentient systems regression may be a natural next step for improved inter-dimensional estimations of building performance simulation results.

CONCLUSION

Sentient BPS systems are yet to be seen as a stable and agile implementation. Much work is needed in the area of predicting building designer requests. Better, more adaptable multivariate interpolation methods needs to be utilized. Additional features e.g. feedback of suboptimal directions will be highly beneficial for the sentience of the system. Nevertheless, the implementation of "sentient", also meaning "responsive", BPS systems promises the building designer a fast feedback with valid results.

Figure 6
The absolute error of interpolation related to the resolution level (higher level means fewer simulation s available for the interpolation algorithm).

REFERENCES

Bleil de Souza, C, BDS 2012, 'Contrasting paradigms of design thinking: The building thermal simulation tool user vs. the building designer', *Automation in Construction*, 22, pp. 112-122

Klitgaard, Jens, JK, Kirkegaard, Poul Henning, PHK and Mullins, Michael, MM 2006 'On the Integration of Digital Design and Analysis Tools', *Digital Architecture and Construction*, South Korea, pp. 187-196

Negendahl, K., KN 2013, 'Building Performance Simulation in the early design stage: An introduction to Integrated Dynamic Models', *Submitted to Automation in Construction*, ., p. .

Caldas, L., LC 2001, *An evolution-based generative design system: using adaptation to shape architectural form*, Ph.D. Thesis, Massachusetts Institute of Technology

Burly, M., MB 2013, *Scripting cultures: Architectural design and programming*, AD Primers

Sullivan, R, RS, Arasteh, D, DA, Papamichael, K., KP and McCluney, R., RMC 1988, 'An indices approach for evaluating the performance of fenestration systems in nonresidential buildings', *ASHRAE Transactions*, 94, p. .

Petersen, S., SP 2011, *Simulation-based support for integrated design of new low-energy office buildings*, Ph.D. Thesis, Technical University of Denmark.

Perkov, TP, THP 2014, *Metode til kontinuerlig præcisering af simuleringsresultater*, Master's Thesis, Technical University of Denmark.

Shi, X., XS and Yang, W., WY 2013, 'Performance-driven architectural design and optimization technique from a perspective of architects', *Automation in Construction*, 32, pp. 125-135

Pedersen, F. Ø., FØP 2006, *A method for optimizing the performance of buildings*, Ph.D. Thesis, Technical University of Denmark.

[1] www.autodesk.com/products/autodesk-simulation-family/features/robot-structural-analysis

[2] www.en.wikipedia.org/wiki/Advanced_Chess

[3] www.grasshopper3d.com

[4] www.apps1.eere.energy.gov/buildings/energyplus/energyplus/about.cfm

[5] www.encyclopediaofmath.org/Lagrange_multipliers

[6] www.bentley.com/GetGC

[7] www.rtw.ml.cmu.edu

[8] <http://www.rhino3d.com>

[9] <http://diva-for-rhino.com>