Simulation of a Heated Room with Window Blind Control

Documentation
for the Simulation Program and
the Fuzzy Control Algorithm used

D.I. Peter Wurmsdobler
September 1994

o.Univ.Prof. Dr. H.P. Jörgl
Institute of Machine and Process Automation
Technical University of Vienna

1 Introduction

Topics of this document are the simulation of a thermal system -a heated room in an arbitrary building- and a control algorithm used for controlling the position of window blinds. The document should be a short description of what happens within the simulation program and how to use it. Additionally it should give a brief description of the control algorithm as the heart of the simulation program.

Since all programs are plain ASCII files, they can be easily edited, if some errors occur or if somebody wants to change system specific files for his site. This is true especially for the `startup`-file in the top level directory of this package. Additional changes can be made to all files representing data like extern data or data referring to the controllers. See the `readme` files in the appropriate directory how to make changes.

The control algorithm should run properly regarding the structure. Only the Fuzzy-Sets have to be tuned when using it in a real implementation, because at the time of writing no realistic values of outside data where available. The same is true for the Fuzzy database, where all rules for controlling the window blindes are placed.

2 Simulation program

The simulation program is written in the MATLAB high level simulation language and consists of several subprograms. The programs are ASCII-files with the extension `.m` and do not need to be compiled, but are interpreted by the MATLAB built-in interpreter. This, however, makes simulation a little bit slower then a compiled simulation program would run. On the other hand the development of a new program is straightforward and easy.

All files do not need to be in one directory and can be split in several subdirectories, depending on their topic. The top level directory `delta` contains five subdirectories for special purposes, the files `startup` and `readme`. Each subdirectory contains specific files and a separate `readme`-file. Additionally, all `.m`-files have a header to describe their function. So it should be possible to navigate through the whole package, since all files should be moreless self-explaining. The top level directory contains:

```
/delta:
CHANGES      database     fuzzy        readme       simulink     startup.m
INDEX        exdat        misc         simul.m      startctr.m   sysdat

/delta/database:
blindctrl1.m  fuzzydat1.m   hvlkdat.m
blindctrl2.m  fuzzydat2.m   readme

/delta/exdat:
readme        sommer.dat   testdat.dat  winter.dat

/delta/fuzzy:
compop.m     fuzzyfy1.m   fuzzyfy3.m   rule.m
defuzzy.m    fuzzyfy2.m   readme

/delta/misc:
artlight.m   hlkreg.m     iniprint.m   mauer.m      simend.m     sysmatz.m
datload.m    inicond.m    inisave.m    readme       simuloop.m
fenster.m    iniplot.m    lightpow.m   sclock.m     sysload.m
```

```
/delta/simulink:
artlight.m   bldctrl2.m   ctrldat.mat  ctrlini2.m   hvactrl.m    readme
bldctrl1.m   blindlib.m   ctrlini1.m   demo.m       lightpow.m

/delta/sysdat:
readme      sysdat.m    sysdat.tpl  sysdat1.m
```

The subdirectory `/delta/database` is used for controller data to be stored in it, and the Fuzzy blind control algorithm in `/delta/database/blindctrl*.m`. The structure of both heat and blind controller must not be changed, but all values can be edited and tuned for a specific application. The system data, representing all data of the heated room like wall thickness, thermal coefficients etc. are stored in `/delta/sysdat/*.m`. In the same directory there is a template file called `sysdat.tpl`, which can be used to generate a new system data file.

Extern data such as sun radiation or outside temperature are necessary for simulation. To provide this, data have to be layed down in data files in the directory `/delta/exdat`. How this ASCII-file has to look like, is described in the respective `readme`-file.

All functions, related to the fuzzy controller are placed in `/delta/fuzzy`. There are three different functions for Fuzzyfication, one for each rulebase treatment and rule composition, and one for Defuzzyfication.

The directory `/delta/misc` contains different files for loading data like `datload.m` and `sysload.m`, functions for generating the state representation of the thermal system like `fenster.m`, `mauer.m` and `sysmatz.m`. Other functions like `inicond.m`, `iniplot.m`, `iniprint.m` and `inisave.m` are necessary for initialisation of the initial conditions, the plot commands in the simulation window, the print commands and the save commands. The most important program is `simuloop.m`, the simulation loop, which uses the files `artlight.m` for calculation of the artificial light amount,`lightpow.m` for estimation of sun radiation power and `sclock.m` to calculate time.

2.1 How to get started

After having started MATLAB, change to the directory where the simulation package is installed. this should be `/delta`. Run the startup procedure by typing `startup` to set the Matlab environment for your site. To finally start a simulation, just enter `simul` at the Matlab prompt, and follow the instructions.

The main program `simul.m` defines the integration interval called `dtint` as first step (Figure 1). Afterwards it invokes the programs `datload` and `sysload`. Both come up with a menu, where the user can select a file. In the latter one the system data stored in a selected file of `/sysdat/*.m` are loaded and used for the calculation of system matrices to describe the dynamical behaviour of the thermal system.

The program `datload` searches in the directory `/exdat` for files with the extension `.dat` and loads the selected datafile to the MATLAB workspace. Further all data are interpolated here for each integration step to save time during simulation. After this task has been completed successfully, more data are loaded from the directory `/database` both for the HVAC controller and the Fuzzy controller.

In the following file `inicond` all initial conditions are set up as well as vectors for saving the calculated data are prepared, again to save time during simulation. When all these tasks are done, the graphical user interface (GUI) is built with all actions which can be done during simulation like start and stop simulation, or loading new data to the MATLAB workspace. What has to be done when starting an action is defined in the so called 'CallBacks'. The GUI presents following menu and submenus:

- `Workspace`
  - `Print Window`: Print Window content after simulation in selected PostScript-file.
  - `Save Data`: Save calculated simulation data in selected file at the end.

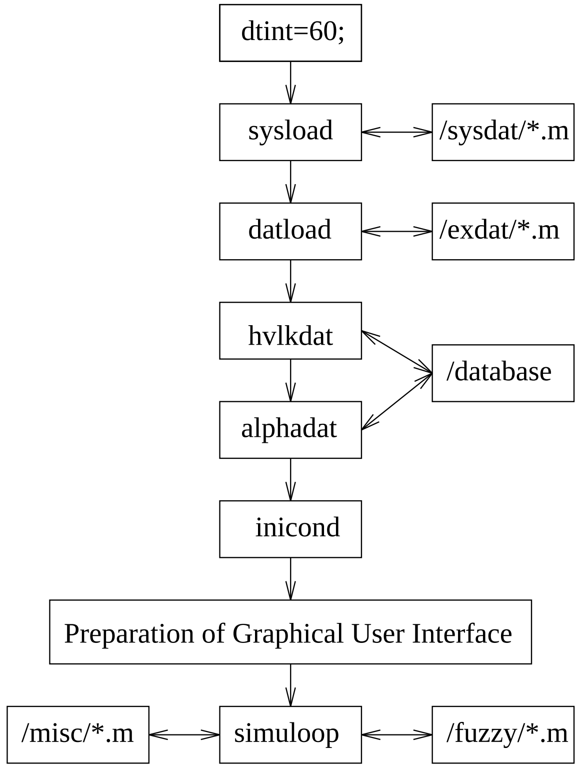

Figure 1: Structogram of simulation program.

  - `Close Window`: Close the simulation window and clear workspace.

- `New_Data`
  - `New Data`: Load new extern data and interpolation for simulation.
  - `New System`: Load new system data and recalculation of system matrices.
- `Simulation`
  - `Start` or `Restart`: Start simulation at the beginning, else restart.
  - `Stop`: Stop Simulation.
  - `Pause`: Pause Simulation.
  - `Continue`: Continue Simulation.

If one process is started, not all other processes are enabled, depending on the process type, e.g.: if `Simulation` is started, `Printing` or `Loading` new data is not possible. Generally, it should be easy to follow the possible actions.

2.2 What happens during simulation

The most important program here is `/misc/simuloop.m`. Here all updates of time dependent values and graphical output is done. The main part in this file is the control algorithm (see Figure 2). A detailed description of the control algorithm for the window blindes is topic of the next section.

As long as `t` is less equal `tfinal`, the whole loop is run through. The first point is to pick up outside data values from prepared vectors. Then the HVAC-controller calculates the heating or cooling power

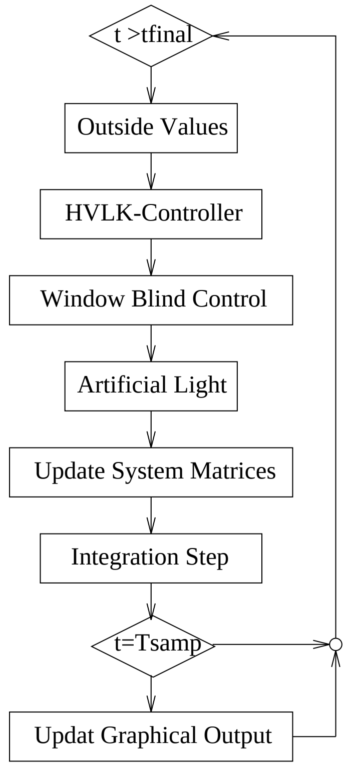

Figure 2: Structogram of simulation loop.

corresponding to his algorithm when sampled at time `Tsamp`. The amount of energy for artificial illumination must be calculated, because it results in an additional heat source. After the the blind position is calculated when sampled with `Talpha` or when interrupted by a user, all system matrices are updated and the integration can take place. If `t=Tstore`, graphical output and storing in prepared vectors is done.

3 Control Algorithm

Conventional window blind controllers mainly can't achieve a high user acceptance, and don't care about the influence on the energy consumption for heating or cooling a building. This new control concept should consider all influences of window blinds on the thermal behaviour of a building as well as on the light situation within a room of the respective building, which leads to acceptance for a user. To cope with these requirements, the entire control algorithm has several inputs, namely:

1. Values provided by the HVAC-system or alternively by building control
   (a) Setpoint for ambient temperature depending on the mode of the HVAC-controller (economic or comfortable),
   (b) Ambient temperature,
   (c) External temperature,
   (d) Heating Power, either values in Watts or flags for heating 'on', 'off',
   (e) Nobody bit determines whether somebody is inside
   (f) Wind velocity
2. Values provided by the light management

(a) Measured sun light values in four directions horizontally and vertically and one additional global value for thermal radiation,
(b) Actual blind position
(c) User defined blind position
(d) User defined weight
(e) Flag determines user interrupt

The control algorithm basically consists of four parts. The first part calculates estimates for both sun radiation in Watts and sunlight in Lux depending on nine measured light values. The second one determines a blind position and the corresponding weight of the control action from the HVAC point of view. The third part operates in terms of light, including the user wish depending on stored wishes and spontaneous user interrupts. Finally, the fourth part makes a synthesis of the upper requests. This is shown in Figure 3.

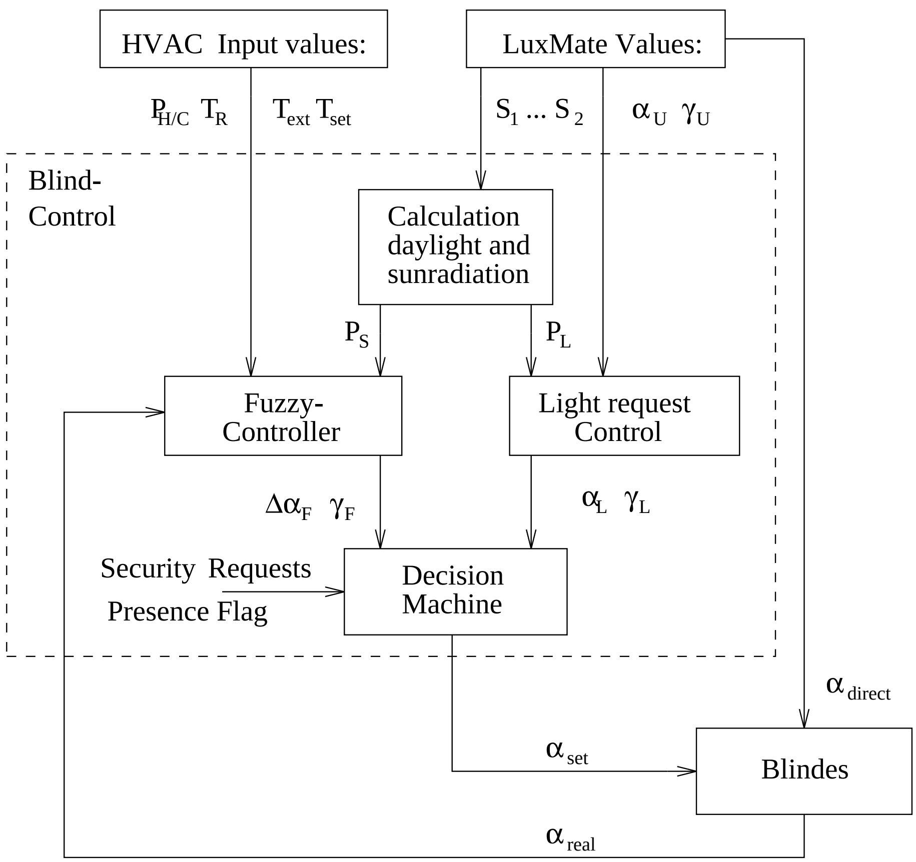

Figure 3: Structure of control algorithm

3.1 Fuzzy Controller

The most comprehensive part is the first mentioned above. A Fuzzy Controller is used for this special control task. Since the rules how to manage the window blinds for a room seem to be moreless heuristic, this control strategy should fit best.

Input variables for the controller are:

P_S, the power of sun radiation, with three Fuzzy-Sets *low, little* and *high.*

$P_{H/C}$, the power for heating or cooling given by an HVAC controller, with three Fuzzy-Sets *negative, zero* and *positive.*

T_R, the room temperature, with three Fuzzy-Sets *cold, comfortable* and *warm.*

T_R, the outside temperature, with three Fuzzy-Sets *cold, comfortable* and *warm.*

α, the actual window blind position, with three Fuzzy-Sets *closed, medium* and *open.*

For all input variables three triangular Membershipfunctions (MSFs) or Fuzzy Sets are used. This is done for reasons of simplification of the rulebase. Since there are 5 input variables, the total number of possible rules would be $3^5 = 343$, and thats quite a lot. One could say, that not all rules are necessary, but fore sake of Matrix computation all possible rules have to be considered. In this special case a combination of 3 or 2 input variables is used for a rulebase.

Output of this Fuzzy controller is the differential blind position, i.e. how much they should be moved, and the weight of this control output denoting the importance of the control action. For reasons of computation speed, Singletons are used for output MSFs and are defined as:

$\Delta\alpha$, the differential blind position, with five Fuzzy-Sets *down, bitdown, equal, bitup* and *up.*

γ, the weight, with three Fuzzy-Sets *unimportant, nocare* and *important.*

The Fuzzy controller is trained to find an optimum regarding the energy consumption for the special room, but does not consider any user wishes. For there are two output variables, two different rulebases are used. The first is for the window blind position:

```
%        RULEBASES:
%
%        Rulebase T_room, and P_sun, and Blinds
%
%         IF             AND            AND              THEN BLINDS
RB_TS= [ T_room cold,  P_sun low     Blinds closed,  equal   ;
         T_room comf,  P_sun low     Blinds closed,  bitup   ;
         T_room warm,  P_sun low     Blinds closed,  up      ;
         T_room cold,  P_sun middle  Blinds closed,  bitup   ;
         T_room comf,  P_sun middle  Blinds closed,  bitup   ;
         T_room warm,  P_sun middle  Blinds closed,  bitup   ;
         T_room cold,  P_sun high    Blinds closed,  up      ;
         T_room comf,  P_sun high    Blinds closed,  bitup   ;
         T_room warm,  P_sun high    Blinds closed,  equal   ;
         T_room cold,  P_sun low     Blinds medium,  down    ;
         T_room comf,  P_sun low     Blinds medium,  bitdown ;
         T_room warm,  P_sun low     Blinds medium,  up      ;
         T_room cold,  P_sun middle  Blinds medium,  bitup   ;
         T_room comf,  P_sun middle  Blinds medium,  equal   ;
         T_room warm,  P_sun middle  Blinds medium,  bitup   ;
         T_room cold,  P_sun high    Blinds medium,  up      ;
         T_room comf,  P_sun high    Blinds medium,  bitdown ;
         T_room warm,  P_sun high    Blinds medium,  down    ;
         T_room cold,  P_sun low     Blinds open,    down    ;
         T_room comf,  P_sun low     Blinds open,    bitdown ;
         T_room warm,  P_sun low     Blinds open,    equal   ;
```

```
         T_room cold,  P_sun middle  Blinds open,    bitdown ;
         T_room comf,  P_sun middle  Blinds open,    bitdown ;
         T_room warm,  P_sun middle  Blinds open,    bitdown ;
         T_room cold,  P_sun high    Blinds open,    equal   ;
         T_room comf,  P_sun high    Blinds open,    bitdown ;
         T_room warm,  P_sun high    Blinds open,    down    ];
%
%
%        Rulebase P_heat, and T_ext
%
%        IF            AND           THEN BLINDS
RB_HT= [ P_heat neg,   T_ext cold,   up      ;
         P_heat zero,  T_ext cold,   equal   ;
         P_heat pos,   T_ext cold,   down    ;
         P_heat neg,   T_ext comf,   bitup   ;
         P_heat zero,  T_ext comf,   equal   ;
         P_heat pos,   T_ext comf,   bitdown ;
         P_heat neg,   T_ext warm,   down    ;
         P_heat zero,  T_ext warm,   equal   ;
         P_heat pos,   T_ext warm,   up      ];
```

The second is for the weight of the HVAC-output:

```
%        Rulebase P_heat, and T_ext for weighting
%
%        IF             AND              THEN HVAC
RB_GW= [ P_heat neg,   P_sun low,        nocare  ;
         P_heat zero,  P_sun low,        unimptd ;
         P_heat pos,   P_sun low,        import  ;
         P_heat neg,   P_sun middle,     nocare  ;
         P_heat zero,  P_sun middle,     unimptd ;
         P_heat pos,   P_sun middle,     import  ;
         P_heat neg,   P_sun high,       import  ;
         P_heat zero,  P_sun high,       nocare  ;
         P_heat pos,   P_sun high,       import  ];
```

The structure of this controller can be seen in Figure 4Ånother possibility is a seperate rulebase for summer and for winter. Which one is chosen will be decided by a independend variable like an estimate of the whole-day mean temperature.

3.2 User wishes and union of blind position

The third part of the algorithm computes the weight of the user wish depending on whether the wish is spontaneous or programmed. If it there is an user interrupt, the weight is a decreasing, time dependent function, otherwise the value stored by a user is considered. If there is nobody in the respective room, i.e. the "presence flag" is set 1, the user weight is zero. Finally, the output weight is the maximum of both weights.

If the first parts are ready, the fourth part compares the weight of the user wish and the weight from the HVAC point of view,i.e. the importance of the Fuzzy control action. The blind position with higher weight is considered then (see Figure 3). Of course, security requests such as close or open the blindes in

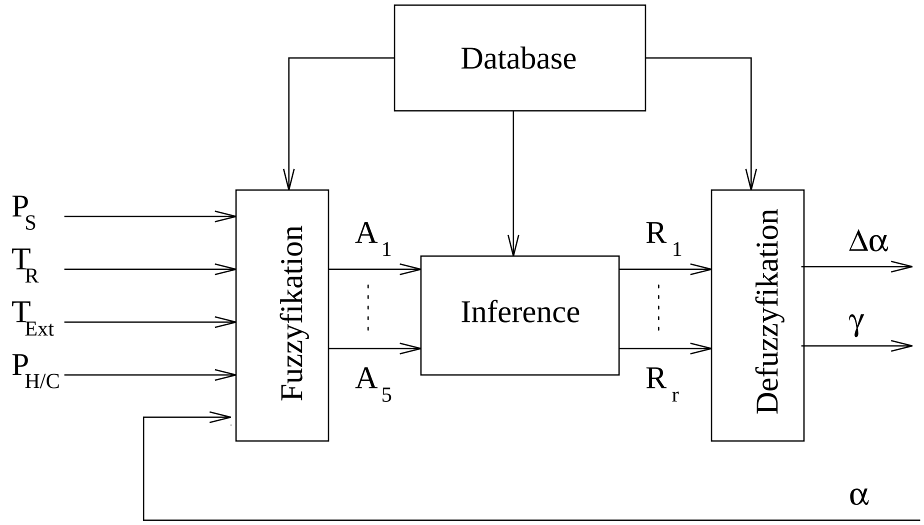

Figure 4: Structure of Fuzzy controller

case of too strong wind can override the precalculated value. The final value of the blind position is then passed to the simulation program.

4 Conclusion

The proposed simulation program should work properly and results can be easily achieved. As mentioned above, the structure of both simulation program and the control algorithm do not need to be changed, but all values of controllers have to be adjusted for a real time application. The simulation should be a powerful tool to predefine the Rulebase and all Fuzzy data. The results of a simulation can either be saved in a `log`-file, or printed in a PostScript-File. The simulation result then might look like Figure 5.

5 Literature

1. **Neirac, F.P.**, *Approche Theoretique et Experimentale des Modeles Reduit du Compartiment Thermiquedes Batiments*, Ecole Nationale Superioreure des Mines de Paris, Paris 1989.
2. **Tantot, M.**, *Contribution a l'Etude de la Gestion des Gain Direct dans un Batiment*, Ecole Nationale Superioreure des Mines de Paris, Paris 1990.
3. **W.Hegetschweiler, T.Furtiger, R. Rainmann, N.Degunda**, *Ein Beitrag zur Beurteilung des instationären Verhaltens wärmespeichernder Bauteile*, Landis & Gyr pp 1-23, 1991.
4. **Bothe**, H.H.: *Fuzzy Logic, Einf"uhrung in Theorie und Anwendungen*, Springer Verlag, Berlin-Heidelberg 1990
5. **Zadeh, L. A**, *Fuzzy Sets*, Information and Control, 1965.
6. **Pedrycz, W.**, *Fuzzy Control and Fuzzy Systems*, John Wiley and Sons, New York 1989.
7. **Zimmerman, H.-J.**, *Fuzzy Set Theory and its Application*, Kluver, Dortrecht 1991.
8. **Geering, H.P.**, *Introduction to Fuzzy Control* , IMRT-report, ETH Z"urich 1992.

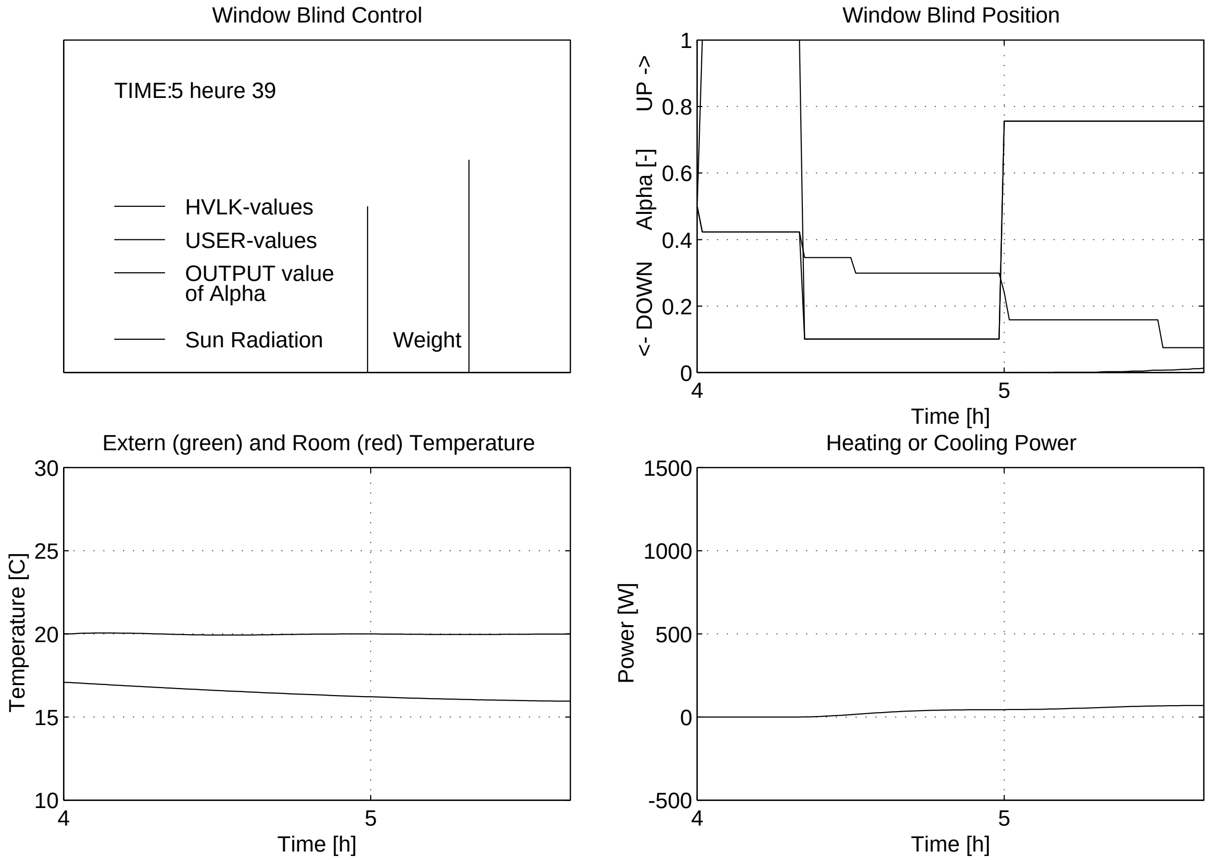

Figure 5: Demonstration of a Simulation

9. **Kosko, B.**, *Neural Networks and Fuzzy Systems*, Prentice Hall, London 1992. Comptes Rendus Soc. Sci. et Lettr. Varsovie, 1932.

10. *MATLAB Users Guide*, The Mathworks, Inc. August 1992.