DETAILED MODELLING AND SIMULATION OF A VAV AIR-CONDITIONING SYSTEM

Philip Haves
Department of Civil & Building Engineering
Loughborough University of Technology
Loughborough, LE11 3TU, UK

ABSTRACT

The paper describes a component-based dynamic simulation of a variable air volume (VAV) airconditioning system. The model is based closely on the design of one floor of a real commercial office building in London. The model includes an air handling unit and a duct system incorporating pressure-independent VAV boxes. The paper describes the simulation environment used to test control systems and to develop fault detection and diagnosis procedures and presents results of simulations that illustrate how the simulation can be used to study the interactions between control loops.

INTRODUCTION

The simulation environment and techniques described in this paper were developed to support the study of control algorithms and strategies [1, 2, 3] and fault detection and diagnosis methods [4]. The key features are:

- explicit treatment of the local loop controls (either in simulation or using real controllers connected to the simulation via a hardware interface)
- modelling of the principle dynamics of each control loop
- explicit modelling and simulation of the flow rates and pressure drops as well as heat flow rates and temperatures

A number of the forms of ideal behaviour assumed in most simulation programs break down in the presence of faults in the HVAC system. Detailed modelling of all parts of the system that interact with the part of the system being studied is therefore required.

At present, the only practical method of achieving the required level of detail is the use of a component-based simulation program such as HVACSIM+ [5] (the program used in the work presented here), TRNSYS [6], IDA [7] or SPARK [8]. This paper describes the simulation environment used and presents some results obtained from a model of the secondary HVAC system in a VAV office building. The requirement to commission the simulation model in the same way that a real building is commissioned is explained. Results illustrating the use of the simulation to study control loop iterations are presented.

SIMULATION ENVIRONMENT

A simulation environment for use in the study and development of control algorithms and strategies, automated commissioning techniques and fault detection and diagnosis methods has been developed, based on HVACSIM+. In addition to the explicit modelling described above, the main features are:

- interprocess communication via Unix sockets with:
  - control algorithms or fault detection and diagnosis procedures implemented as separate processes
  - a 'control panel' that allows control system parameters to be varied interactively during the simulation
- communication with real control systems via an analogue hardware interface
- on-line display of user-selected simulation variables as time series graphs and numerical values
- off-line utility program to display time-series graphs and save selected output data in formats suitable for MATLAB or for spreadsheet programs
- utility program to create, edit and merge boundary files, including the merging of boundary files with discontinuities at different times

Interprocess communication allows the convenience of developing control algorithms or fault detection procedures in a different software environment to that of the simulation, but requires a multi-tasking operating system, such as Unix. When testing real controllers and hardware implementations of fault detection procedures for use with real HVAC systems, it is necessary to run the simulation in real time with asynchronous communication between the simulation and the controllers or fault detection system. This involves the simulation reading the system clock and waiting until the physical time is equal to the simulation time. A hardware interface to real controllers using analogue to digital and digital to analogue converters can be implemented in a number of different ways, depending on the hardware and the operating system, as described in [9].

Interprocess communication, on-line graphics and communication with the hardware interface have each been implemented in the same way that component models of the building and HVAC systems are implemented, i.e. as TYPE subroutines. The only difference between these TYPEs and normal TYPEs is that they are not involved in the iterative solution process. The BLOCK/SUPERBLOCK structure of HVACSIM+, which allows a system to be partitioned in order to reduce the numerical problems associated with simulating large systems, provides a convenient mechanism for excluding these communication TYPEs from the iterative solution process. Placing these TYPEs in separate SUPERBLOCKs and inhibiting simultaneous (iterative) solution at the SUPERBLOCK level prevents their being called more than once per timestep.

IMPLEMENTATION OF SOCKET COMMUNICATIONS

'Socket' communication is a generalisation of the 'pipe' interprocess communication facility that has been a long-standing feature of Unix. It can also provide communication between different computers, via a serial line or Ethernet, although that facility has not been exploited in the work reported here. There are three aspects of socket communications as implemented here:

Initialisation Two uni-directional channels are opened, one for data transfer from the simulation to the process under test and the other for data transfer from the process under test to the simulation.

Writing A fixed number of reals are formed into a string of ASCII characters and sent from the socket in the originating process. In the current implementation, the data sent from the simulation are the time since the start of the simulation and sixteen control signals. These signals can either be sensor signals or, if part of the control scheme is implemented in the simulation, controller outputs. The data sent to the simulation are either outputs from a controller, manual inputs (set- points *etc*) from a control panel, or test signals from an automatic commissioning or fault detection procedure.

Reading The method employed for reading from a socket depends on whether the communication is synchronous or asynchronous. If the simulation and the process under test are not required to run in real time, it is necessary for them to be synchronised, and this is achieved implicitly by the exchange of data. Each process waits for incoming data from the other process before continuing executing. If both processes must run in real time, they can either run synchronously, with only one process reading a clock, or they can run asynchronously, with both processes reading their own clocks. If the communication is asynchronous, a separate signal handler process reads the socket whenever an interrupt indicates that data have been received. Once they have been read, these data are written into an array in memory that is global to both the signal handler and the main process that requires the data. The result is that the main process samples the data that arrived most recently, and any additional data arriving between sample times are lost. This allows communication between the simulation and the process under test when they have different sampling rates.

METHOD OF SIMULATION

The method of simulation is largely determined by the need to use a short time-step (typically 1-5 seconds) in order to treat discrete-time local loop controllers, either by modelling them explicitly in the simulation or by communicating with external devices or processes. The time delay introduced by the sampling action of the controllers allows the principal elements of each control loop (sensor, controller, actuator, plant) to be solved sequentially, reducing significantly the number of equations that must be solved simultaneously. The system model is partitioned using the HVACSIM+ BLOCK/SUPERBLOCK facility; Table 1 shows the structure used for the simulation described in Section 5. At each time-step, each SUPERBLOCK is called in turn and a simultaneous solution of the equations that are internal to that SUPERBLOCK is obtained. The controllers calculate their outputs on the basis of sensor signals produced at the previous time-step. The inputs to succeeding SUPERBLOCKS are the outputs of preceding SUBERBLOCKS produced at the current time-step.

Table 1: Simulation Structure

SUPERBLOCK 1	Read from Unix socket
SUPERBLOCK 2	Controllers
SUPERBLOCK 3	Actuators
SUPERBLOCK 4	Airflow components
SUPERBLOCK 5	(Water flow components)
SUPERBLOCK 6	Thermal components
SUPERBLOCK 7	(Central plant components)
SUPERBLOCK 8	Sensors
SUPERBLOCK 9	Write to Unix socket
SUPERBLOCK 10	Graphical output

The airflow components, water flow components and thermal components can be separated from each other because, in the systems considered here, the only interaction between them is via the controllers. The flow networks give rise to large numbers of non-linear algebraic equations and it is beneficial for numerical reasons to be able to separate them and hence reduce the number of equations to be solved simultaneously. This approach was used in the emulator developed at Tsinghua University [10].

COMPONENT MODELS

Most of the models of mechanical components developed as part of this work are described in [11]. This Section describes the modelling of the rooms and the controls and the extension of some of the models of mechanical components to include faults.

ROOM MODELS

Each enclosed space can be modelled either as a single zone or as a number of interconnecting zones. The airflow part of the zone model consists of a calculation of the flow through a leakage resistance to the outside and a mass balance on the room air volume. The flows included in the mass balance are the supply and extract flow for the HVAC system, interzone flow from up to two adjacent zones, local extract at a fixed rate (e.g. kitchen or toilet extract) and the leakage flow. The model assumes perfect mixing and zero pressure variation within each zone.

The thermal part of the zonel model is an extension of the two time constant, lumped parameter model developed by Laret [12]. In the example building and system model described below, the zone model consists of an occupied space and a ceiling plenum containing supply and extract ducts. Each space is represented by an 'air' node (radiation and convection are combined and the difference between the dry bulb temperature and the mean radiant temperature is ignored) and a structure node, which are linked to each other and to the ambient. The heat balance on the 'air' node in the occupied space includes terms for the supply air and any inward flowing interzone air, as well as gains from occupants, lighting and equipment. An additional heat gain can be specified interactively via a file that is read at each step time. Editing the contents of the file whilst the simulation is running provides a simple method of varying the load on the zone during a run. (Note that the simulation typically runs only one order of magnitude faster than real time even when it is not synchronised to real time.)

The heat balance on the 'air' node in the plenum includes heat gains/losses from the ducts and the lights. A thermal capacity is associated with each node, which, in the case of the 'air' node represents the air itself, furniture and lightweight partitions. The model approximates the response of a zone on time-scales of minutes to hours, making it suitable for generic studies of room temperature control loops and their interaction with other control loops in the HVAC system. A moisture balance is also performed at each 'air' node.

CONTROLLER MODELS

The controller models implemented to date in simulation include:

- PI controller with sequenced output for air handling unit supply air temperature control
- Mixing box (dry bulb economiser) control
- PI control of VAV box with reheat coil
- Supply air temperature reset control
- PI control of fan speed

A number of these controllers are based on strategies used by a major controls manufacturer and have been implemented using a library of primitive functions, paralleling the way strategies are implemented in a number of commer-

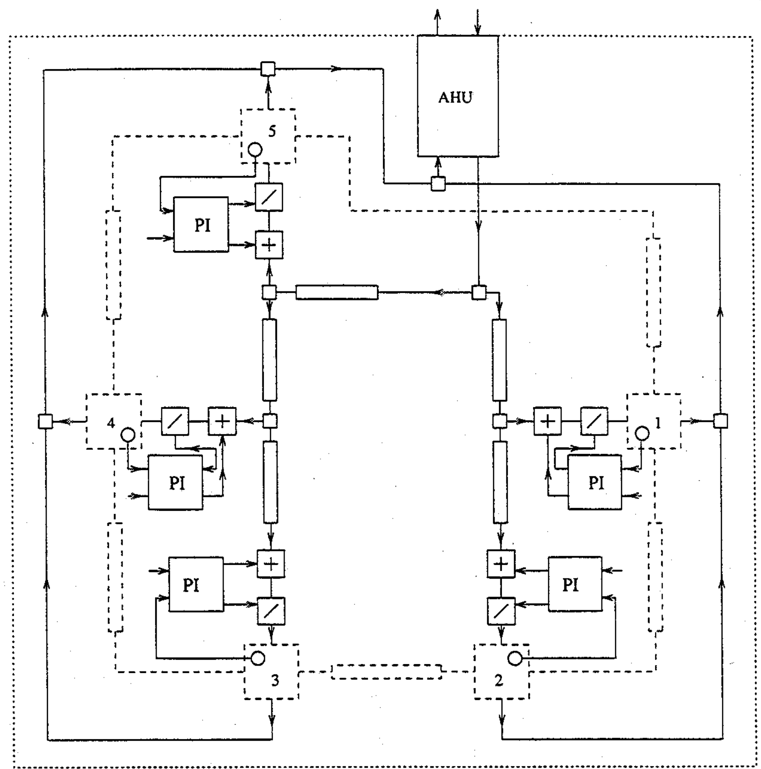

Figure 1: Simplified zoning and duct layout for simulated building. The mechanical components and the controls are indicated by solid lines and the components that treat the interzone flows and zonal mass balances are indicated by broken lines.

cial microprocessor-based control systems. These functions vary in complexity from arithmetic and logic operations to PI control with anti-windup. The implementation in the model consists of a sequence of function calls that are executed at each sample instant. A manual over-ride is included in each controller, allowing open loop commissioning and tuning tests to be performed interactively via the control panel.

DESCRIPTION OF THE SIMULATED BUILDING AND HVAC SYSTEM

The simulation used to illustrate a number of the features described above is based on one floor of a commercial office building in London that has been documented as part of a research project to develop techniques to automate the commissioning of building control systems [13]. Detailed information from equipment manufacturers' specifications and measurements made during the commissioning of the building have been used to calculate the values of the parameters of the component models. The building has a variable-air-volume (VAV) air-conditioning system and each floor has its own air handling unit. It was considered impractical to model explicitly each of the twenty five terminal boxes on the floor in question, and so the open plan office space has been divided into five zones, mainly on the basis of orientation and the routing of the main ducts.

Figure 1 shows the configuration of the simplified duct system. Zone temperature is controlled by modulating the demanded flow through the pressure-independent terminal box and the position of the reheat coil valve in sequence. The number of components in the airflow part of the model has been minimised by including the resistance of each branch of the duct system in the flow split and flow merge components and by combining the resistances of the various components of air handling unit. The resistance of each branch of the duct system was estimated from measurements of the corresponding flow rate and pressure drop in the real building made during commissioning. The pressure drop across each resistive element is assumed to vary as the square of the flow rate if the flow rate is significant and a linear relationship is used at low flow rates in order to avoid numerical difficulties. Further details of the resistance model and the models used to represent the fans, mixing

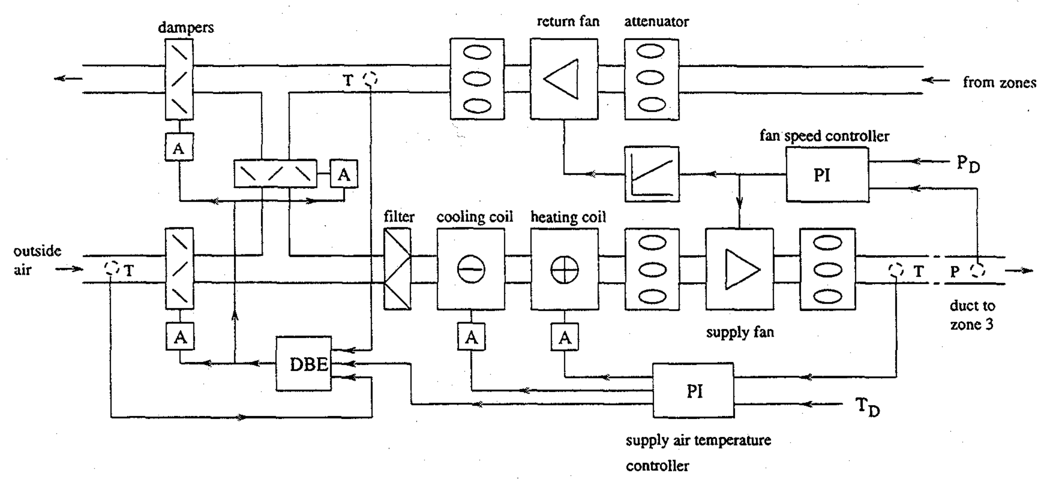

Figure 2: Diagram of the air handling unit in simulated building. *DBE* indicates the dry bulb economiser controller, A indicates a motor-driven actuator, T a temperature sensor and P a static pressure sensor.

box, pressure-independent VAV boxes, flow splits and flow merges are given in [11]. It should be noted that the pressures that are calculated are *total* pressures. However, the pressure measurement that is used in the control of the supply fan is *static* pressure, and so the pressure sensor model includes the cross sectional area of the duct as a parameter so that the velocity pressure can be calculated from the mass flow rate. Figure 2 shows the configuration of the air handling unit and its controls. The rotation speed of the supply fan is varied in order to control the static pressure in the duct supplying zone 3. The speed of the return fan depends on the speed of the supply fan, using a relationship determined during commissioning, as described below. The temperature of the supply air is controlled by modulating the cooling coil valve, mixing box dampers and heating coil in sequence. The required direction of movement of the dampers is determined by comparing the outside air temperature to the return air temperature using a dry bulb economiser controller. The set-point for the supply air temperature is determined by a supervisory controller (not shown) that seeks to keep the set-point as high as possible when in cooling mode, whilst ensuring that the cooling load is satisfied in each zone.

The partitioning of the simulation so as to separate the air flow and thermal networks has the consequence that, in a number of cases, the airflow behaviour of a particular mechanical component is treated separately from the thermal behaviour. For example, the relationship between pressure rise and flow rate for each fan is treated in the airflow SUPERBLOCK, whereas the temperature rise across each fan is treated in the thermal SUPERBLOCK. At present, two alternative versions of the fan model are used: one calculates the pressure rise from the flow rate and the rotation speed, and the other calculates the temperature rise from the pressure rise and the flow rate. This situation, which arises from the need, in programs such as HVACSIM+, to partition the simulation at the component level, is clearly unwieldy and mitigates against wider use of component-based simulation of larger systems. Programs such as IDA and SPARK, which support partitioning at the equation level, have a clear advantage in this respect.

COMMISSIONING THE SIMULATION

A consequence of the detailed level of modelling of the HVAC system is that there is a very large number (773) of parameters whose values must be determined. As noted above, most of these values were determined from manufacturers' data and measurements made during the commissioning of the real building. However, partly as a result of the approximations and simplifications in the system model, there are a number of parameters whose values cannot be taken directly from the real building. One important category of these parameters is the tuning parameters for the controllers, which must be determined by tests on the simulation model.

interzone resistance ($kg^{-1}.m^{-1}$)	interzone flow rates ($kg.s^{-1}$)					execution time (s)
100	-0.1780	0.0873	-0.1500	0.0950	0.1873	58.6
10	-0.1884	0.0897	-0.1615	0.1037	0.2023	59.0
1	-0.1896	0.0890	-0.1629	0.1038	0.2046	108.9

Table 2: Interzone Flow Rates and Execution Times for Different Values of the Interzone Resistance

COMMISSIONING THE AIRFLOW NETWORK

1. The VAV boxes were set to fully open and the fan control mode set to open loop. With the fans set to full speed, a check was made to verify that the flow rates and pressures were approximately as expected. This step is intended to trap major errors in the configuration of the system model and to confirm that the design flows could be obtained without the VAV box dampers being so far closed as to to make for poor control when minimum flow rate is demanded.

2. The interzone resistances were set to a relatively high value and progressively reduced to confirm that interzone flows that were essentially independent of the interzone resistance could be obtained before the onset of numerical difficulties (see above). The results are shown in Table . There is a significant change in the flows as the resistance is decreased from 100 $kg^{-1}.m^{-1}$ to 10 $kg^{-1}.m^{-1}$ and a much smaller change as the resistance is further reduced to 1 $kg^{-1}.m^{-1}$. The last column of the table is the execution time required for the first 200 seconds of the simulation, which involves finding a solution from the very poor initial guess of zero flow and zero pressure drop in each branch. The execution time increased by almost a factor of two on going from 10 $kg^{-1}.m^{-1}$ to 1 $kg^{-1}.m^{-1}$ and a significant number of warning messages indicating numerical difficulties were produced. 10 $kg^{-1}.m^{-1}$, which gives rise to pressure differences between zones of less than 1 Pa, was therefore selected as the interzone resistance. (The supply mass flow rates to the five zones are in the range 1 - 2.4 $kg.s^{-1}$.)
   This procedure is the only one that does not parallel the commissioning activities in a real building.

3. The relationship between the supply fan speed and the return fan speed required to maintain modest positive pressure (10-20 Pa) in the occupied spaces was determined by:
   (a) Setting the demanded flow rates through the VAV boxes to the design flows
   (b) Setting the mixing box dampers to give 100 % outside air
   (c) Adjusting the speeds of the supply fan and the return fan until the pressure rise across the fans was high enough to produce the design flows in each zone and the zone pressures were ~20 Pa
   (d) Varying the position of the mixing box dampers and verifying that the zone pressures did not vary outside the range 10-20 Pa
   (e) Setting the demanded flow rates through the VAV boxes to their minimum values (40 % of the design flow rates) and repeating steps (3) and (4)

   A linear relationship is used in the fan speed controller to determine the speed of the return fan N_R from the speed of the supply fan N_S:

$$N_R = A + BN_S$$

   The coefficients A and B are determined from the values of N_S and N_R obtained in steps (3) and (5).

4. The fan speed controller was then tuned using the closed loop Ziegler-Nichols method [15], which involves determining the critical proportional gain for the loop to oscillate, together with the period of this oscillation. The proportional gain and integral time for the PI controller are then determined from the critical gain and the period. The critical gain and the period were each found to vary by a factor of two when going from the maximum (design) air flow rate to the minimum, reflecting the non-linear relationship between fan speed and supply duct static pressure. The tuning values adopted were obtained from the smaller critical gain.

RESULTS

This Section illustrates how the simulation can be used to study the interaction of HVAC control loops. Figure 3 shows the effect of changing the gain of the fan speed controller on the response to step changes in the demanded flow rate through one of the VAV boxes. The fan is switched on

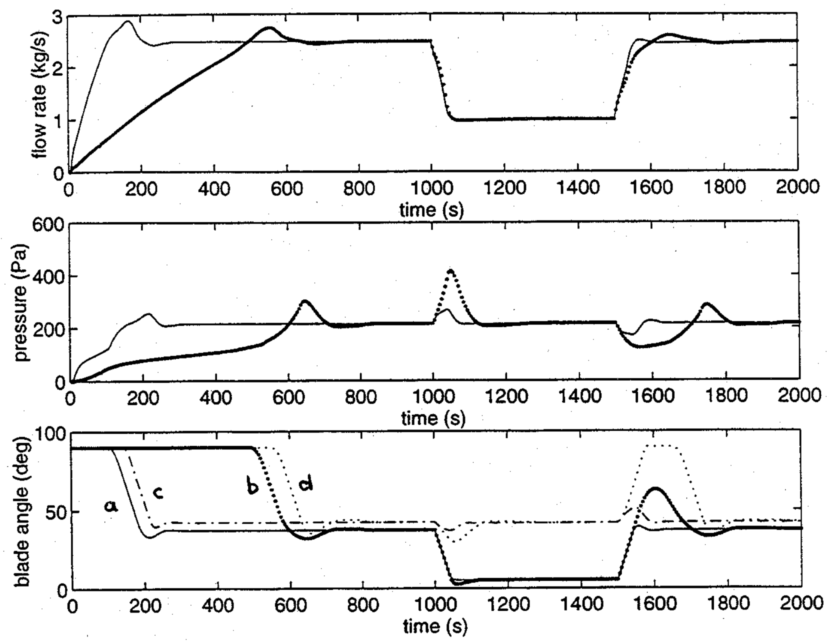

Figure 3: The effect of changing the gain of the fan speed controller

at the start of the run and the demanded flow to zone 1 is reduced to 40 % after 1000 s and increased back to 100 % at 1500 s. The top plot shows the flow rate to zone 1 and the middle plot shows the pressure at the static pressure sensor. In each plot, the continuous line is for the correct tuning and the heavy dotted line is for the reduced controller gain. The bottom plot shows the position of the damper blades in the VAV boxes for zone 1 (*a* – correct tuning, *b* – reduced gain) and zone 3 (*c* – correct tuning, *d* – reduced gain).

In the case of the reduced gain, the fan speed changes more slowly, taking nearly ten minutes to attain full speed. (The dynamics of the loop are determined primarily by the rate limit on the fan speed, so that the minimum time for the fan to attain full speed is 40 s, and the gain of the controller is limited by the need to achieve stability when there is minimum flow rate to each zone, as discussed in Section 6.) When the demanded flow to zone 1 changes, there is a deviation in the pressure that is quite significant in the case of the reduced gain. This change in pressure is compensated for by the flow controller in each VAV box. The bottom plot shows that, in the case of the reduced gain, the damper for zone 3 goes fully open for approximately two minutes in order to compensate for the reduced pressure in the supply duct.

CONCLUSION

A simulation-based development environment suitable for the development and testing of control algorithms and strategies and fault detection and diagnosis procedures has been developed. A detailed model of a VAV system has been developed that includes explicit modelling of the duct system and the controls. Such simulations are very complicated (the system model described here has 98 components, 324 system variables and 773 parameters) and there is a need for a systematic procedure to check that the system model has been correctly assembled. The problem is analogous to that of commissioning a real HVAC system and similar procedures can be adopted.

The models treat the main dynamics and nonlinearities of each control loop and hence the simulation can be used to study the interactions between such loops. The explicit modelling of the controls allows the simulation to be used to test automatic commissioning and fault detection procedures that need to interact with the control system [14]. The facility to implement the controls either in the simulation or in external hardware provides additional flexibility.

Further work is required to develop and test subsystem models for the water networks. There is a need to develop dynamic models of central plant components such as boilers and chillers in order to be able to model a complete HVAC system and its controls. There is also a need for validation, both at the component level and at the system level.

ACKNOWLEDGEMENTS

The author wishes to thank A L Dexter for useful discussions on the modelling of control systems and J A Wright for calculating the parameters of the duct system model. The work was supported in part by the UK Building Research Establishment and the UK Science and Engineering Research Council.

REFERENCES

[1] A L Dexter and P Haves. A robust self-tuning controller for HVAC applications. *Transactions of the American Society of Heating, Refrigerating and Air-Conditioning Engineers*, 95, Pt. 2, 1989.

[2] P Haves, A L Dexter, D R Jørgensen, K V Ling, and G Geng. Use of a building emulator to develop techniques for improved commissioning and control of HVAC systems. *Transactions of the American Society of Heating, Refrigerating and Air-Conditioning Engineers*, 97, Pt. 1, 1991.

[3] A L Dexter and P Haves. The influence of tuning on the performance of a building control system. In *Proceedings of System Simulation in Buildings '90*, Liège, Belgium, December 1990.

[4] M Benouarets, A L Dexter, R S Fargus, P Haves, T I Salsbury, and J A Wright. Model-based approaches to fault detection and diagnosis in HVAC systems. In *Proceedings of System Simulation in Buildings '94*, Liège, Belgium, December 1994.

[5] C Park, D R Clarke, and G E Kelly. An overview of HVACSIM+, a dynamic building/HVAC/control systems simulation program. In *Proceedings 1st. Annual Building Energy Simulation Conference*, Seattle, WA, 1985.

[6] S A Klein, W A Beckman, and J A Duffie. TRNSYS - a transient simulation program. *Transactions of the American Society of Heating, Refrigerating and Air-Conditioning Engineers*, 82, Pt. 2, 1976.

[7] P Sahlin and A Bring. IDA solver - a tool for building and energy systems simulation. In *Proceedings of Building Simulation '91*, Nice, France, August 1991. IBPSA.

[8] E F Sowell, W F Buhl, A E Erdem, and F C Winkelmann. A prototype object-based system for HVAC simulation. In *Proceedings of System Simulation in Buildings '86*, Liège, Belgium, December 1986.

[9] S W Wang, P Haves, and P Nusgens. Design, construction and commissioning of building emulators for EMCS applications. *Transactions of the American Society of Heating, Refrigerating and Air-Conditioning Engineers*, 100, Pt. 1, 1994.

[10] Q S Yan, Y Jiang, and Y X Zhu. A building and plant real time simulation. In *Proceedings of Building Simulation '89*, Vancouver, BC, June 1989. IBPSA.

[11] P Haves. Component-based modelling of VAV systems. In *Proceedings of System Simulation in Buildings '94*, Liège, Belgium, December 1994.

[12] L. Laret. Development of building dynamic multizone model for large scale simulations. In *Proceedings of the USER-1 Building Simulation Conference*, Ostend, Belgium, September 1988.

[13] A L Dexter, P Haves, and D R Jørgensen. Development of techniques to assist in the commissioning of HVAC control systems. In *Proceedings of CIBSE National Conference*, Manchester, UK, May 1993.

[14] A L Dexter, P Haves, and D R Jørgensen. Automatic commissioning of HVAC control systems. In *Proceedings of CLIMA 2000*, London, UK, November 1993.

[15] J G Ziegler and N B Nichols. Optimum settings for automatic controllers. *Transactions of the American Society of Mechanical Engineers*, 64:759–768, 1942.