

ColSim - A NEW SIMULATION ENVIRONMENT FOR COMPLEX SYSTEM ANALYSIS AND CONTROLLERS

*Christof Wittwer, Werner Hube, Peter Schossig
Fraunhofer Institute for Solar Energy Systems ISE
Oltmannsstrasse 5, D-79100 Freiburg, Germany
e-mail: christof.wittwer@ise.fhg.de*

*Andreas Wagner, Christiane Kettner, Max Mertins, Klaus Rittenhofer
University of Karlsruhe, Section of Building Physics and Building Services
Englerstr. 7, D-76128 Karlsruhe, Germany
e-mail: wagner@fbta.uni-karlsruhe.de*

Abstract

The paper describes the main features of the new simulation environment ColSim [15] and a special application for investigating ventilation strategies in buildings. ColSim was developed with the intention of simulating and testing complex control strategies in different HVAC and thermal solar systems, coupled to a building model. It is based on a modular structure and a modified Euler algorithm was chosen for the numerical integration of the nonlinear differential equations. In contrast to existing programs, time steps ranging down to one second are used in order to receive an accurate response of the dynamic systems. An energy and mass flow balance is automatically executed within every time step. Current activities focus on the development of controllers to drive pumps and fans in various applications. All control routines and models within ColSim are implemented in ANSI-C code and can be operated on LINUX/UNIX platforms as well as in embedded system controllers. Different public domain tools are integrated in ColSim e.g. for graphical input or online plotting.

Introduction and Features of ColSim

The simulation of controller strategies with the simulation code TRNSYS¹ [1] and other commercially available programs is not possible without modifying numeric algorithms of the programs' solvers in order to achieve convergence with small time steps. Furthermore, the implementation of controller routines in Fortran within the programs is absolutely unsuitable for the code transfer to modern micro-controller hardware. Experiencing these (and other) restrictions of existing simulation programs was the original motivation to create a new simulation environment. ColSim is based on the UNIX multitasking philosophy² and the idea of public domain software. The current platform is a LINUX system (default installation SuSE

LINUX) with different public domain tools included to setup the simulation environment. Most important is the GCC C-Compiler which represents a solid base for highly reliable software and development environments. ColSim is primarily a command line driven tool, which can be controlled interactively by a simple TCL-menu. The visualization of online results is realized with the interactive function plotting program GNUPLOT [14]. The design of systems to be simulated can be done with the vector graphic tool XFIG [13], which allows to convert a simulation script with all necessary object relations and parameter information.

The initial intention for creating ColSim was not the simulation of a building's thermal performance, but

¹ColSim is based on a similar structure to the modular simulation code TRNSYS

²ColSim can be compiled as well on 32-bit-MS-windows systems

primarily the simulation of HVAC and solar thermal systems with its controllers. But the modularity of ColSim with its objects and the graphical representation allowed an extension of a building model which is described below in more detail. The high resolution of time of a simulation in ColSim now enables to investigate particularly non linear physical effects, which often occur in energy saving technologies for buildings (e.g. phase change materials in a wall system, transparent insulation etc.).

Kernel and Numerics of ColSim

The differential equations of the dynamic system are solved by a finite difference method. ColSim uses a spreaded working³ modified Euler solver to calculate system state at run time of each object. The Euler algorithm requires small time steps to ensure stability but in return it's simple to realize, fast to execute, and has further advantages in comparison to other classical integration methods (e.g. Runge-Kutta):

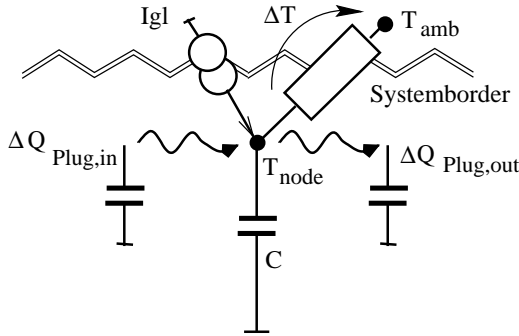


Fig1: Node and Plug flow modeling

In a typical thermal RC-network with more than one path, the solver simply returns the following state temperature T_{k+1} based on the discrete time axis, which means that the single energy flow through each path is undetermined. The state function of the node is usually described as a first order differential equation $mc_{node} \frac{dT_{node}}{dt} = \sum \dot{Q}_{path}$, where the energy flows on the right side depends on the state T_{node} of the actual state. Euler approximation uses a simple transfer into a difference equation $mc_{node} \frac{\Delta T_{node}}{\Delta t} = \sum \Delta Q_{path}$, where one type of heat flow is defined by resistances

and temperature differences $\Delta Q_{res} \approx kA(T_{node} - T_{amb})\Delta t$. ColSim also uses finite energy plugs (plug flow technique) for modeling mass flow processes: $\Delta Q_{plug} = \dot{m}c_P\Delta t(T_{plug} - T_{ref})$. The difference equation (state function) of the thermal capacity is calculated via the energy of node E_{node} by the sum of path energies⁴ in order to minimize rounding errors.

$$E_{node}(t) = E_{node,init} + \sum(\sum \Delta Q_i \Delta t)$$

$$T_{node} - T_{ref} = E_{node} / (m_{node} c_P)$$

With a freely defined reference temperature of $T_{ref} = 0$, the current state is calculated with the energy of the node E_{node} . The energy flow across the system border is registered by a globally working routine checking the energy balance by utilizing the path energy flows.

Stability: Considering the time step, the time constants of each path $\tau = RC = mc/kA$ have to be checked for the models at run time to ensure stability of the solver. The Euler approximation \dot{Q}_{res} uses the tangent of the state function over the time and gets the state of the following time step, using a fixed time step $h = \Delta t$. For the stabilization the observation of the ambient temperature limits the losses of the node to the ambient level.

Critical mass flows with respect to the time constant are controlled with the so called Mixed Plug model, where an incoming energy plug $\Delta Q_{plug,in}$ is first mixed with the energy of the node E_{node} which results in the node temperature $T_{node,mix}$. In a second step $\Delta Q_{plug,out}$ is calculated with the current mixed state $\dot{m}c_P(T_{node,mixed} - T_{ref})\Delta t$. Further energy paths are calculated separately in a second calculation step of the state function.

Plug Flow Model: The plug flow model allows the tracing of the mass flow through a closed hydraulic network, which guarantees a fast calculation of the object chain. The plug is initiated at a pump object, is shifted through the branched network and is returned to the origin (pump). This mechanism allows simple modeling of flow delay effects and a mass balance at the second call of the pump in one time step.

³System equations are solved inside the object calculation

⁴integral of heatflow from simulation start to current point of time

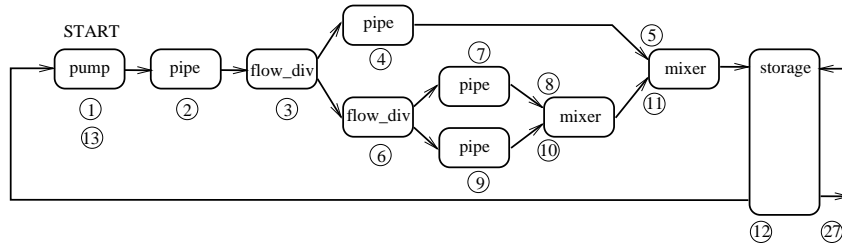


Fig 2: Plug flow modeling in a multi-path closed hydraulic cycle (calculation order)

Error detection (e.g. missing connection between two objects) is one of the most important tasks of a simulation code, since the developer is usually spending most of the time for debugging a new system design. Checking the mass flow and system energy balance are the main features of ColSim's error detection mechanism.

Structure of ColSim and Object Linking:

ColSim is a development environment with an open structure where everything can be modified. The central document is the system graph 'system1.fig'⁵ which is created using the vector graphic program XFIG [13] and a graphical model library. The parser *cnv.exe* in the subdirectory *./cnv* converts the system description into the simulation script 'system1.dek' in the ColSim home directory.

Each object is identified with a UNIT number and uses a TYPE identifier which selects the associated source codes in the *src*-directory. The configuration script in the users projects folder allows to link the source code in a local project directory where user defined modifications can take place. This feature is very useful, since some projects require very specific models. The main routine reads the simulation script, load profiles, weather data and writes simulation data to an ASCII output file. It calls the online plotter which works as a subtask based on GNUPLOT [14]. ColSim is completely implemented in ANSI C code with the following advantages:

- portable code, (32Bit MS Windows, UNIX, Linux, Embedded Systems for model based controller development)
- acceptable hardware resources
- fast execution time
- completely free development environment based on GPL license

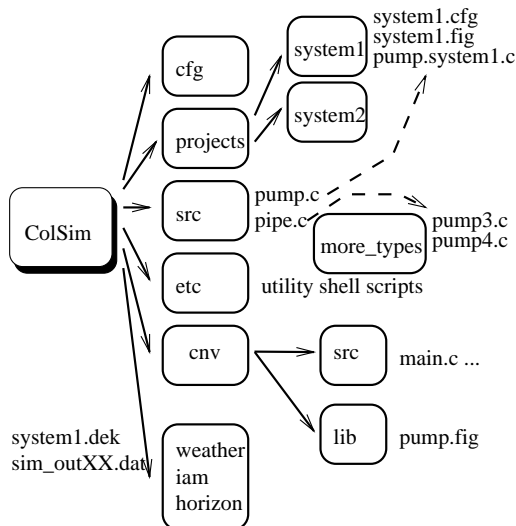


Fig 3: Structure of ColSim

It's a traditional ASCII file, which allows fast modifi-

⁵parameters are bound in the 'hidden' mode to the model object (ASCII object)

Example System for Testing Control Strategies in a HVAC system

A ventilation system with a PI controller or alternatively with a fuzzy controller was set up to control the ventilation rate and the heating supply for a single zone arrangement. The controller receives the signals of three sensors: CO_2 , humidity and temperature. A thermodynamic model of the air flow was created the plug flow idea: Initiated by the fan, the plug flow is described by the vector $\vec{v} = (\Delta Q, \dot{m}_{dry}, \dot{m}_{steam}, \dot{m}_{water}, \dot{m}_{CO_2})$ where ΔQ is the enthalpy plug, which is shifted through the closed network of the ventilation system. The mass flow is intentionally not described by the volume rate to strictly realize a global mass balance in the system. The object structure is assembled with following elements: *pipe_air*, *fan* (*pump_air*), *mixer_air*, *flow_diverter_air*, *heater_air*. A conditioning to any state is realized via the '*heater_air*' model, which allows temperature setting, humidity setting and CO_2 conditioning.

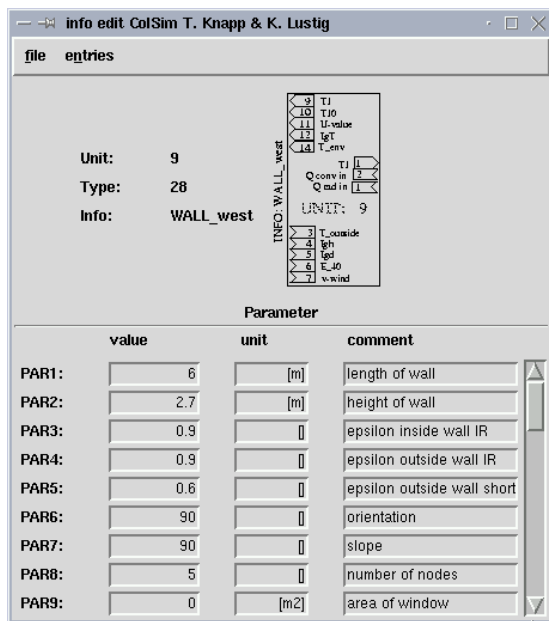


Fig. 5: Parameterization of the objects (TCL)

The *pipe_air* element is a very versatile element, which can be used as an earth heat exchanger model [16][12] with coupled mass elements or as a non-dynamic zone model which can be coupled to the complex building model as shown in figure 4. It has to be pointed out, that the plug flow modeling is specially designed for control application, since the time resolution is very high. In the shown example with 21 units a time step of 10 seconds is used and leads to typical simulation times of about 20 minutes on a PIII 850MHz machine (simulating one year). The building model is characterized by a fully detailed representation of each wall object and a type which is called '*AirRadNode*'. The input and output structure allows to monitor any internal state, which is useful for binding new sub models into this type. The energy exchange is realized by these inputs and they are documented in the simulation script. The wall element is realized with finite difference technology and is classical structured in layers and nodes. It includes a window element with the option of processing weather data though a radiation processor. Internal heating of single nodes to model transparent insolation or wall heating systems is easy to realize, also a substitution of nodes with phase change materials which is characterized by temperature depended heat capacity $c_{P,pcm} = f(T)$. The object '*AirRadNode*' is realized as a simple two-star network [7] with n walls, which allows to calculate independent infrared radiation (with T_{star}) and convection heat flow between the inner surfaces and the air node. Here the developer can include a heating system, where radiation and convection paths are calculated separately.

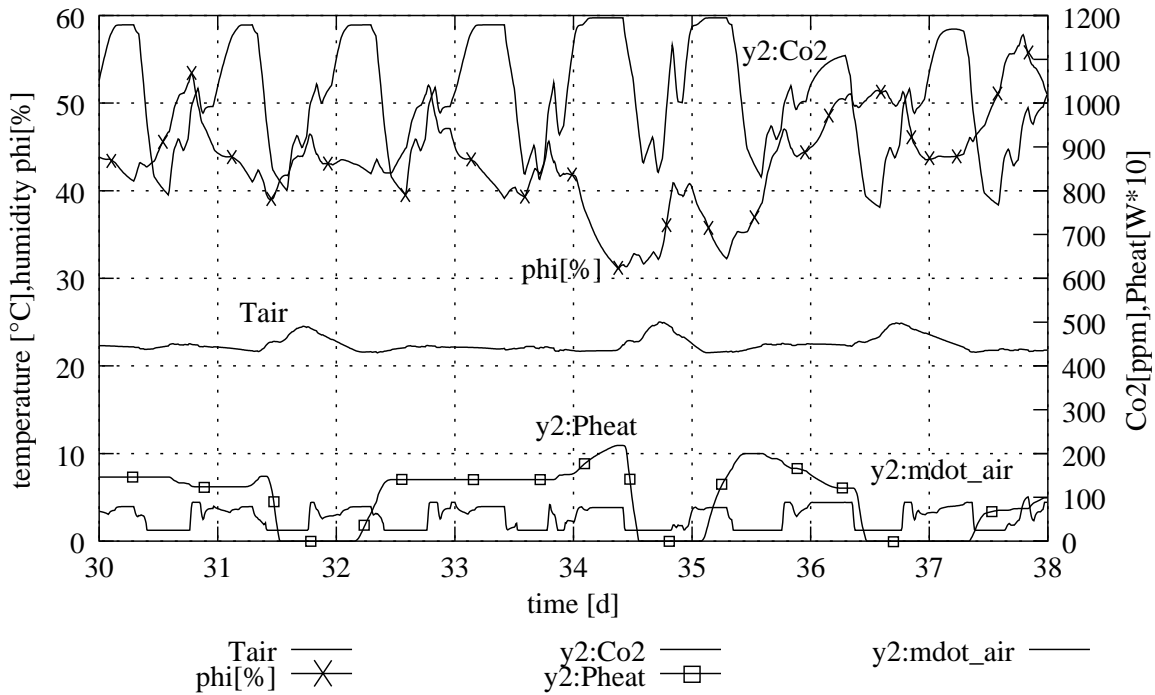


Fig 6: 10 days sequence of the described HVAC system with a fuzzy control concept optimizing ventilation rate, Co2- and humidity conditions

A cross validation with other well-known simulations codes (e.g. TRNSYS and ESP-r [2], see Fig. 7) has been carried out using the simulation test method BESTEST [17] [7].

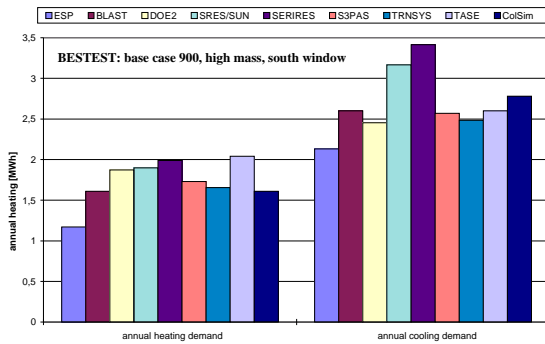


Fig. 7: annual heating and cooling demand for BESTEST case 900, ColSim vs. others

Conclusion

ColSim has been designed to develop controller concepts (e.g. predictive energy management systems) with a special model library that allows investigation with the necessary time resolution. Models and con-

troller algorithms are suitable for embedded systems (ANSI-C) to realize a combined development of system optimization and controller realization. We are working on several research projects at the moment to realize model based HVAC controller systems. The model library covers solar thermal systems, ventilation systems (active and passive cooling), earth heat exchanger [12], desiccant cooling systems [6] and fuel cell power station as well as different building models. A model for coupling thermal and hydrodynamic effects in heating and ventilation systems will be realized soon to optimize complex heating and ventilation networks, where a discontinuous operation is used to minimize parasitic energy consumption of pumps and fans.

References

- [1] TRNSYS: Klein, S.A.; Beckmann, W.A.: A Transient System Simulation Program Solar Energy Lab, Madison Wisconsin, 1988
- [2] ESP-r: Clarke, J. A.: Energy Simulation in Building Design, 1985
- [3] Duffie, J. A.; Beckmann, W.A. Solar engineering of thermal processes.
- [4] Mitalas G. P.; Stephenson, D.G.: Calculation of Heat Conduction Transfer Functions for Multi-layer Slabs. ASHRAE Annual Meeting Washington 1971 New York : Wiley, 1980
- [5] Hube, W.: Systemmodellierung and Validierung einer hybriden Heizungsanlage für ein Niedrigenergiehaus und Entwicklung einer lastabhängigen Regelungsstrategie, Diplomarbeit Uni Kaiserslautern 1999
- [6] Mertins, M.: Entwicklung eines Modells für Dessicant and Evaporative Cooling (DEC) und Implementation in eine Simulationsumgebung, Diplomarbeit Uni Karlsruhe 2000
- [7] Chen T.: Entwicklung und Implementierung eines Raummodells in der Simulationsumgebung ColSim, (BESTEST) Studienarbeit Uni Karlsruhe 1999
- [8] Wittwer, C.: Durchflußregelung einer solaren Kollektoranlage mit Schichtspeicher. Diplomarbeit Universität Kaiserslautern 1993
- [9] Wittwer, C.: ColSim - Simulation von Regelungssystemen in aktiven thermischen Anlagen; PhD thesis <http://www.ubka.uni-karlsruhe.de> 1999
- [10] Wittwer, C.: A Flexible Simulation Program for Multi Tasking Operating Systems to Investigate Controllers in Thermal Solar Systems. Proceedings EuroSun Freiburg 1996
- [11] Wittwer, C.: Simulation eines wissensbasierten Regelungskonzeptes mit ColSim. 8. Symp. Thermische Solarenergie OTTI 1998
- [12] Rittenhofer, K.: Implementierung eines luftgeführten Systems in die Simulationsumgebung ColSim und Validierung anhand eines Erdwärmetauschers, 1999
- [13] Smith, B.: XFIG - Facility for Interactive Generation of figures, bvsmith@lbl.gov
- [14] Williams, T.: GNU PLOT: An Interactive Plotting Program. info-gnuplot@dartmouth.edu, 1997
- [15] ColSim www.colsim.de
- [16] Källblad K.: Calculation methods to predict energy savings in residential buildings, Swedish Council for Building Research, 1983
- [17] BESTEST: International Energy Agency Building Energy Simulation Test (BESTEST) and Diagnostic Method, NREL, National Renewable Energy Lab, 1995

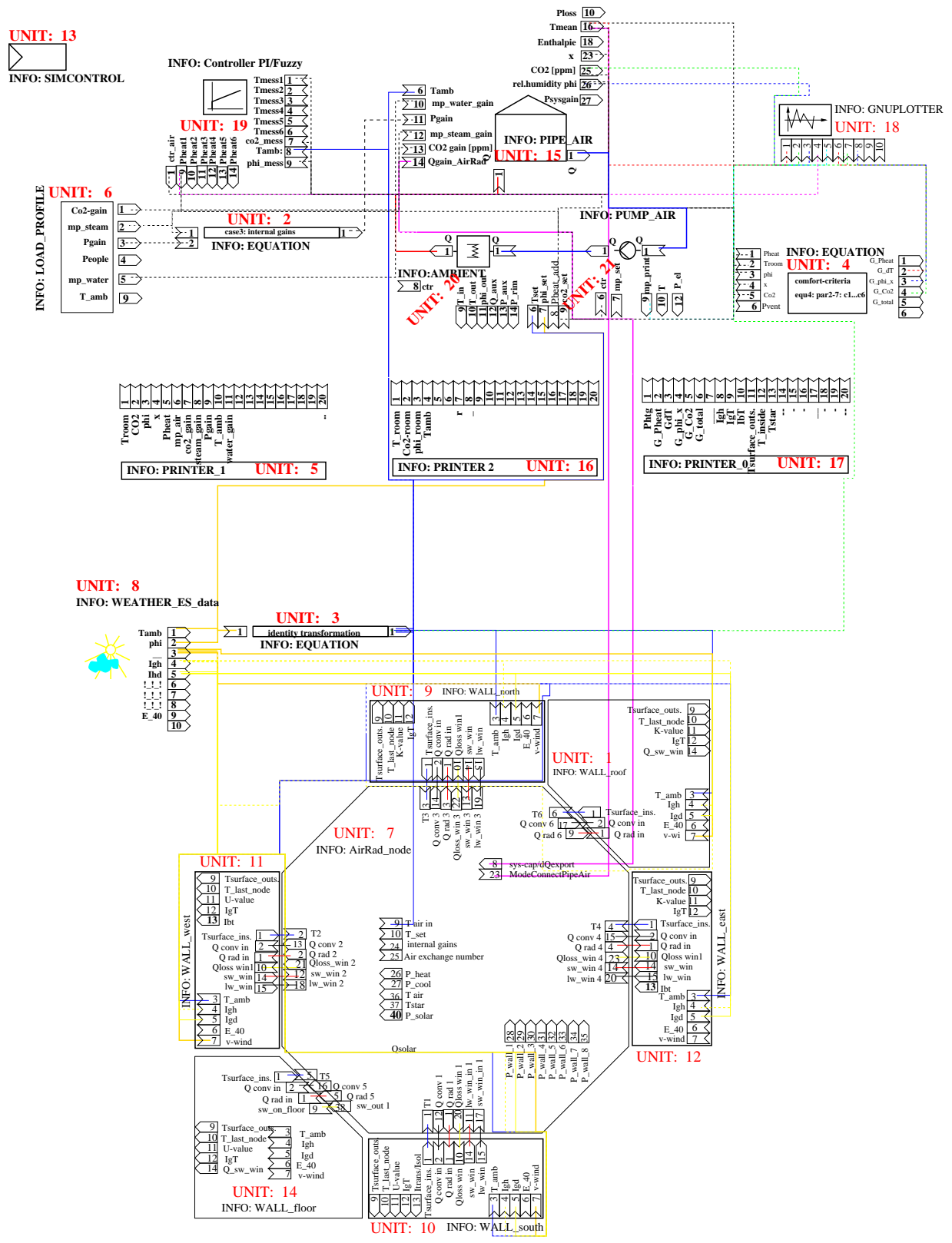


Fig 4: HVAC System with a coupled single zone, modeled as XFIG objects