CHAPTER 5 INTEGRATED SIMULATION BACKGROUND

In this chapter the simulation techniques used for construction of the human energy simulation system are shortly discussed. A brief overview is given and the need and motivation behind simulation as well as a discussion on the implementation of the techniques for construction of the model is shown. Furthermore, the procedure for solving the model is also outlined.

5.1 Introduction

In this chapter a brief overview of the general simulation techniques used for construction of the human energy system model is given. The techniques are widely used and well-researched. The purpose of this chapter is however not to present new techniques, but to provide an introduction into integrated simulation in order to clarify the discussion on the simulation model provided in the following chapters.

In 1994 LeBrun gave a good explanation of why simulation is important for general engineering applications [1]. His expertise was thermal building simulation but the same principles apply for any discipline. He noted that the dream of engineers is to find simulation techniques that will allow them to go, without any discontinuity, through all steps of a technical solution.

The main challenges he highlighted were:

- to propose models that are easy to understand and easy to improve, whenever necessary;
- to produce reference data allowing the user easy parameter identification;
- to introduce the simulation models in the earliest stages of development and to implement them in such a way that they can be used along the entire life of the system.

According to Cellier simulation is nothing more than a way of performing an experiment [2]. But, the experiment is with a model of the actual system, not with the system itself. The advantage of simulation is that the experiment can be performed relatively inexpensively and in the case of biotechnological simulations, without potentially harmful repercussions.

5.2 Dynamic integrated simulation

One of the most involved simulation procedures, namely dynamic integrated simulation, is a powerful means of determining the complex performance and response of a system [3]. The purpose of this study is to firstly characterise the model presented in Section 7.2 in order to simulate blood glucose response. The model can then be employed to predict beforehand what the influence of other disturbances such as medicine, exercise, etc. will be on blood sugar levels.

The physical system can be modelled in a virtual software environment. However, the software system is still an abstract representation of reality and the degree of fidelity still requires somewhat of a trade-off. An abstract model cannot hope to reflect realistically all the details of the behaviour of the actual system. Often many of the details are simply not known.

The more detailed the simulation, the more accurate and detailed the required input data has to be. Often simulations also generate masses of detailed output data. It is therefore usually not cost-effective (and sometimes counter productive) to model too much detail. It is certainly not feasible to try to include all the dysfunctional models of behaviour of a system in a simulation. The creation of software objects for the model components therefore requires careful consideration of the type of questions that the simulation program is supposed to answer [4].

Careful consideration went into the design of the simulation model aimed at predicting blood glucose response. Because the entire human energy system is so complex (see Section 6.2), many of the processes were simplified. These include processes like cellular uptake of energy, endocrine control, etc.

Mathematically, component models are derived as a combination of fundamental principles and empirical correlation coefficients. In the case of the glycaemic energy subsystem simulation model, presented in Chapter 7, the simulation procedure and component models make use of explicit equations in order to ensure solvability.

Because the component models are designed generically to a certain extent, the simulation procedure or simulation "engine" is able to simulate any control strategy or even a defect such as diabetes. Typical pre-constructed control units (for example counter regulation hormone models) are readily available as part of the overall simulation process. These can be implemented at any time to investigate different control effects.

5.3 Model design

The simulation engine can be divided into two distinct parts, namely,

- the component object model and;
- the system solution method.

The component object model contains the basis upon which all the system components in the simulation model are based. These models consist mainly of fundamental and first principal equations such as those derived in earlier chapters of this study. These are used for calculating the dynamic response based on physical reaction and empirical correlation coefficients for the required component. The detailed derivation of these components will be discussed later in Section 7.3.

The simulation process contained within the system solution method on the other hand consists of solving the "energy flow" or "energy balance" of the component models. To arrive at an acceptable calculation, both numerical and explicit solving techniques are used at every time interval. However, for the simplified model no solving of differential equations is necessary, only numerical methods and algorithms are used.

The following steps are repeated at each time interval:

- At the beginning of each time interval the energy flow of the system is solved to compute the correct amount of energy that flows to and from each component.
- After the energy flow simulation has been completed each component model is explicitly solved for a set number of iterations within the time interval or until explicit values are reached. This allows each component within the simulation to reach steady state.
- The above two steps continue to repeat until the required number of iterations are reached or until the end of the required time constraint is attained.

To account for accurate steady state modelling the control models are furthermore processed at the beginning of each time interval. All the controller elements are solved according to their predefined control strategies and their required control inputs. Stoecker discusses various other methods for simulating dynamic control that fall outside the scope of this study [5].

In summary, the proposed simulation engine can be classified as a discrete, deterministic and dynamic integrated system simulation model.

5.4 Solving the simulation model

5.4.1 Energy flow between model components

As described above, the first step at any give time interval is to solve the flow of energy to and from the model components. To set up a given configuration of the system, a schematic layout is used. The layout describes the system components, the connections between the components (such as energy flow pathways) as well as the controller connections. The controller connections do not represent actual flow, but rather represent signals to and from controller components. These layouts are discussed in detail in Chapter 7.

Usually engineering simulation models for dynamic fluid flow applications take into account three distinct laws of nature. These are the well-known laws of conservation of mass, momentum and energy [6]. All three operate on the same principle, namely that the resultant flow at any given point in the model has to be zero over any given control volume.

In other words, if for example the flow of energy is considered at an organ, the amount of energy that flows into the organ has to be equal to the amount of energy that flows out of the organ unless the organ stores some of the energy or if some of previously stored energy is released or if some of the energy is used (burnt). Equation (5.1) describes this principle for energy flow ($\dot{E}$). The equation implies that energy cannot be created nor be destroyed.

$$\dot{E}_{in} = \dot{E}_{out} + (\dot{E}_{stored} - \dot{E}_{released}) + \dot{E}_{used} \tag{5.1}$$

How the storage, release and utilisation of the energy occur is dependent on the component model within which it is calculated. The flow between the components is only described by $\dot{E}_{in}$ and $\dot{E}_{out}$. However, for the conservation laws mentioned above to hold true over any control volume, both $\dot{E}_{in}$ and $\dot{E}_{out}$ have to be connected to other organs or at least to external flow components (such as the digestive component).

Equation (5.1) considers energy flow. Both mass flow and momentum conservation laws can also be solved in a similar fashion. However, in this study the simplification of only the human energy system was the key issue, so only energy flow was considered. The dependency on the other two conservation laws was therefore ignored.

This approach enables a very simple solving technique for the energy flow network. It only consists of a set of linear equations that need to be solved. The straightforward nature of the human energy system model allows the set of linear equations to consist of fewer unknown variables than equations. Solving with a simple substitution method therefore almost always produces a solution.

For this purpose a Gaussian elimination algorithm with backward substitution is used [7]. The algorithm aims to eliminate all the variables in the linear system and thereby isolate one. Then, using backward substitution, the other equations can be solved in sequence by again isolating one unknown variable at a time.

5.4.2 Energy flow to and from the components

After the flow of energy between the model components have been solved, the component models themselves need to be solved numerically and in the correct order. For this purpose the same schematic layout mentioned in Section 5.4.1 is used.

The layout establishes the configuration but the solution order however still needs to be determined. Tarjan devised a recursive algorithm to transform any complex schematic layout into a spanning tree [8]. This algorithm is adapted to determine the order in which the components that constitute the human energy system need to be solved.

Shortly, the algorithm, also used by Van Heerden, is as follows [9]:

1. Store all the system components in a "component list". At first ignore all the controller components.
2. All the components in the simulation model for which the properties are already known are stored in a separate list, namely the "initial list".
3. The next step is to apply the Tarjan Depth-First-Search algorithm with each component in the initial list to determine starting points [8]. These components are subsequently deleted from the component list and placed in a stack associated with each "tree".

4. At this stage all the components that remain in the component list will be part of closed loops. Choose any component in the list and reapply the Tarjan Depth-First-Search algorithm [8]. This last step has to be repeated until the component list is empty. The first component in any of the "trees" will be regarded as the unknown component for which an "unknown list" can be compiled.

After this algorithm is applied, each tree consists of the components in any given flow network. The trees can be traversed and the network can consequently be solved. However, the interactions between the various trees have not been taken into account. These interactions will occur at components that form part of two or more trees.

To solve this cross referencing problem a technique presented by Van Heerden is used [9]. This algorithm entails the following steps:

1. Solve all the flow pathways to compile a "known list" from the initial list.

2. Then traverse a tree, and insert the components that are found in a "visited list". All the components in the visited list are then deleted from the tree.

3. If all incoming pathways are not in the known list then the next tree is selected and the above steps are repeated until all the trees are empty.

Eventually all the components will be visited and inserted into the visited list. The components can then be solved individually in the order in which they appear in the visited list. As variables from the first solved components become available, they can be used in the next components, and so on.

5.5 Computer implementation

In the late 1980's the object-oriented method for software construction has become the method of choice for design and implementation of large complicated computer programs [10]. In contrast to the old algorithmic method, which viewed a computer program as a complicated data processing tool, the basic idea behind the object-oriented philosophy is that the software must emulate the real world. In object-oriented programming the distinction between data and algorithm is much more diffuse [11].

The data and the methods are regarded as equally important, both forming an integral part of the software object. With this philosophy a physical object is described by a set of attributes and actions. MacRandall and French discussed the benefits of the object-oriented method from a building energy simulation point of view [12],[13].

The close correspondence between software representation and physical objects is ideal for simulation purposes. With this method it has become feasible to create a simulation model which is easy to use, easy to configure and easy to extend [4].

All the system components in the simulation model are modelled using objects. The internal states and processes that occur inside the components are all coded as methods within the classes and hence the objects. This approach structures the program in an understandable and adjustable manner that can be visualised as will be shown in Chapter 7.

This programming approach allows for far better control over the elements of large computer applications, such as the simulation model and solver. A base class is constructed that defines a generic component, like for example a human organ that has in and out flow connections. Then a more detailed class is defined for every type of organ containing storage and utilisation methods. Then an even more detailed child organ is derived to describe the finer details of every specific organ. If changes have to be made to all the organ flow connections only the base class needs to be edited. This allowed much faster development of the simulation component models.

Object-oriented programming therefore allows a very modular approach to representing reality in a virtual manner [11]. New control strategy classes or component models can easily be coupled with existing classes without interfering with the methods within those components. These generic techniques are implemented in the simulation engine for simulation of the entire human energy system.

5.6 Conclusion

Very basic simulation techniques were therefore used for the construction of the human energy system simulation model. Better techniques and more efficient algorithms have undoubtedly been developed and would probably make the solving procedures even more streamlined. However, the procedure presented here produced adequate results and the possible improvement due to better techniques did not justify the extra research required for their implementation.

In the following chapters the techniques mentioned here will be employed to construct the models based on the detailed design presented in Chapter 7.

5.7 References

[1] LeBrun, L., Simulation of HVAC systems, *Renewable Energy* **Volume 5, Part 5, August**, 1151-1158 (1994).

[2] Cellier, F. E., Continuous System Modelling, *Springer-Verlag*, (1991).

[3] Arndt, D. C., Integrated dynamic simulation of large thermal systems, *Thesis presented in partial fulfilment of the requirements for the degree Philosophiae Doctor in the Faculty of Engineering, University of Pretoria*, **November** (2000).

[4] Lombard, C., Two-Port Simulation of HVAC Systems, an Object Oriented Approach, *Thesis presented in partial fulfilment of the requirements for the degree Philosophiae Doctor in the Faculty of Engineering, University of Pretoria*, **January** (1996).

[5] Stoecker, W. F., Design of thermal systems, *Mc-Graw-Hill Book Company, Incorporated, New York*, **3rd Edition** (1989).

[6] Rousseau P. G., Mathews E. H. & Malan A. G., Verification and application of a new component-based simulation technique for HVAC systems, *R&D Journal*, **Volume 12, Number 1**, 1-12 (1996).

[7] Burden, R. L. & Faires, J. D., Numerical Analysis, *PWS-KENT Publishing Company, 20 Park Plaza, Boston, MA 02116-4324*, **5th Edition** (1993).

[8] Tarjan, R., Depth-first search and linear graph algorithms, *SIAM Journal of Computing*, **Volume 1, Issue 2**, 146-160 (1972).

[9] Van Heerden, E., Integrated simulation of building thermal performance, HVAC system and control, *Thesis presented in partial fulfilment of the requirements for the degree Philosophiae Doctor in the Faculty of Engineering, University of Pretoria*, **June** (1997).

[10] Bell, D., Morrey, I. & Pugh, J., Software engineering – A programming approach, *Prentice-Hall International*, (1987).

[11] Osier, D., Grobman, S. & Batson, S., Teach yourself Delphi™ 3 in 14 days, SAMS Publishing, 201 West 103rd Street, Indianapolis, Indiana 46290 (1997).

[12] MacRandall, D., Some trends in computing: The implications of simulation, *Energy and Buildings*, **Volume 10**, 249-258 (1988).

[13] French, J. C., Object-oriented programming of HVAC control devices, *ASHRAE Journal*, **December**, 33-41 (1991).