MODELLING FOR CONTROL: UNDERSTANDING ROLE AND FUNCTION OF REGULATORY NETWORKS IN MICROORGANISMS

Krist V. Gernaey[*,], Morten Lind[***] and Sten B. Jørgensen[**]**

[*] *Industrial Electrical Engineering and Automation, Lund University, Box 118, SE-221 00 Lund, Sweden*
[**] *CAPEC, Department of Chemical Engineering, Technical University of Denmark, DK-2800 Lyngby, Denmark. E-mail: sbj@kt.dtu.dk*
[***] *Ørsted•DTU, Automation, Technical University of Denmark, Building 326, DK-2800 Lyngby, Denmark*

Abstract: Microorganisms maintain a constant monitoring of extra-cellular physicochemical conditions in order to respond and modify their gene expression patterns accordingly. Modelling plays an essential role in developing an understanding of the regulatory networks in microorganisms. Means-end analysis, also called functional modelling, is proposed as a methodology to model regulatory networks. A modelling example for the *E. coli lac* operon illustrates that the methodology allows breaking down complex regulatory networks into elementary building blocks. The proposed modelling formalism enables a straightforward communication between systems engineers and biologists on the mechanisms underlying complex regulatory functions. Copyright© 2005 IFAC.

Keywords: Control, functional modelling, means-end analysis, systems biology, systems engineering

1. INTRODUCTION

Microbial function is carefully controlled through a network of proteins and other signalling molecules, which enables microorganisms to react on changes in their environment. Microorganisms maintain a constant monitoring of extra-cellular physicochemical conditions in order to respond and modify their gene expression patterns accordingly. Such adaptation of the cell to changes in the environment is crucial for the survival of the cell, since it allows economical use of cellular resources (Lengeler *et al.*, 1999), as a result of regulating the expression of all genes to produce the optimal amount of gene product at any given point in time. Thus microorganisms constitute examples of entire autonomous chemical plants, which are able to produce and reproduce despite shortage of raw materials and energy supplies. The similarities between microorganisms and chemical plants almost naturally lead to an interest of systems engineering in understanding biological function.

The industrial interest in the understanding of biological function is a consequence of the tremendous and steadily growing list of products resulting from biotransformation processes (see e.g. Cheetham, 2004). Improved understanding of the regulatory mechanisms responsible for expression of the gene encoding a product of interest might lead to higher production rates (more product can be produced within an existing industrial facility), increased production yields (raw materials can be utilised more efficiently), and shorter time to market. Thus, for an industrial biotransformation process, the results of improved understanding of biological function are directly related to increased profit.

Understanding the regulatory networks in microorganisms, and especially understanding how to couple the microbial regulatory functions and the higher level process and production control functions is a prerequisite for process engineering. The purpose of this paper is a discussion of basic modelling problems that arise when attempting to describe regulatory networks and their function in microorganisms, as well as to propose a method to model the regulatory networks. First, fundamental modelling problems are highlighted. Means-end analysis and functional modelling are subsequently introduced as suitable methods to represent the complex interactions in regulatory networks. Their use is illustrated using the *Escherichia coli* lactose utilisation (*lac*) operon as an example. The paper ends with a discussion of the results and conclusions.

2. FUNDAMENTAL MODELLING PROBLEMS

2.1. Levels of abstraction

Models of the biological system play a central role in both reverse and forward engineering. However, a model of the biological system represents different types of knowledge and assumptions about the system depending on the problem to be solved. A general problem in the modelling of dynamic systems is to determine a proper level of abstraction. Most natural and artificial systems can be modelled on a variety of levels but the choice of level is of particular importance for biological systems due to their extreme complexity. Unfortunately, levels can be defined relative to several dimensions in the modelling problem. For example, we can describe the spatial structure (the anatomy) on many part-whole levels, and we can also describe the behaviour (dynamics) at several part-whole levels of temporal resolution.

Another way to define levels in biological systems is to consider their functional organisation. The idea here is to describe the biological system as a goal directed system and to decompose the system into subsystems so that each subsystem serves the needs or provides the means for its superordinate system. The analysis that brings about this type of system information is usually called means-end analysis or functional modelling, and has been developed within cognitive science and artificial intelligence research. The use of means-end analysis to define levels of abstraction is a powerful approach to handle the modelling of complex dynamic systems (Lind, 1994). It is of particular importance for modelling systems with embedded control systems, such as biological systems. Control systems play a direct role in the constitution of functional levels (Lind, 2004) and their function can therefore not be described properly without means-end concepts. Note that when using concepts of means-end analysis we must distinguish carefully between the concepts of behaviour and function. The two notions are often confused so that function is more or less synonymous with behaviour. We stress the teleological meaning of function so that it represents the role the system has in the fulfilment of a purpose or goal. Behaviour refers to what happens when a system reacts to an intervention or a disturbance. Descriptions of behaviour have accordingly no connotations to purposes or goals and are therefore distinct from functional descriptions.

2.2. The interpretation problem

The interpretation problem originates in the multifunctional nature of microbial systems. Where a subsystem in most engineering systems only serves one or a few functions, it may serve many functions that may be interdependent in microbial systems. A function is not an inherent property of the subsystem but is defined relative to other subsystems and the purposes of the system of which they are the parts. A protein may thus serve at least three different functions. It can serve as a substance (material or product, e.g. in protein degradation reactions) in a metabolic process, it can serve as an enzyme promoting another reaction and it can act on the DNA for promoting or blocking the expression of genetic information (transcription factor). The complexity of microbial systems originates in this unique ability of proteins to enter into a multitude of functional relations.

The identification of functions requires knowledge of how a subsystem contributes to the whole. This knowledge about the functional organisation of the system is a prerequisite for the formulation of a set of ODEs describing the system, because it determines the level of abstraction adopted and the system features to be included in the equations (Lind, 2004). A clear distinction must be made between organisational (functional) complexity and behavioural complexity (Doyle, 2004): Behavioural complexity can be expressed by ODEs, but we need other concepts to model the organisational complexity. The purpose of a model of the organisational complexity is to define in formalised language the functional relations between subsystems and the biological system as a whole. Such a model comprises an abstract qualitative representation which can be used to communicate the understanding obtained of the biological system. Often informal sketches or graphics are used to communicate functional knowledge. However, more formal concepts are required in order to ensure clear semantics and consistency of the models. A formalised model of the functional organisation is therefore a complement to and not merely a mediocre or less accurate version of an ODE model.

3. MEANS-END ANALYSIS

3.1. Elementary actions

The possibility of defining a set of elementary action types has been addressed by Von Wright (1963), and has been explored further for application in means-end analysis of complex dynamic systems by Lind (1994, 2004). The elementary action types are actually derived from a set of corresponding elementary change types. The idea is that an action results in a change of the state. Conceptually, the change caused by the action would not appear if the action was not done. The definition of an action contains therefore a reference to a hypothetical situation which is not realised because the action was done. Now, by defining a change as a transition between two states, we can define four so-called elementary changes (Table 1). Each change is defined by both a linguistic description and a logic formula, which is composed of a proposition p representing the world state, a temporal operator T (Then) and one of the four change verbs "happens", "remains", "disappears" and "remains absent". In this way the formula ~pTp (~p Then p) is a logic representation of the change described by "p happens".

Table 1. The elementary action types of Von Wright (1963); p denotes a state, T denotes 'then', and I denotes 'instead'

Types of elementary change		Types of elementary action	
Description	Formula	Description	Formula
p happens	~pTp	produce p	~pTpI~p
p remains	pTp	maintain p	pTpI~p
p disappears	pT~p	destroy p	pT~pIp
p remains absent	~pT~p	suppress p	~pT~pIp

We shall not go into details about the logic definitions here. However, it is notable that the list of elementary changes is a logically complete list, so that all changes in the world can be defined provided we define the state in question by a proposition p (actually we will also need to combine elementary changes). Each elementary change has a corresponding elementary action type as indicated in Table 1. The action formula contains the temporal operator T (Then) and an additional operator I (Instead) used to indicate the hypothetical state. The logical formula ~pTpI~p represents in his way the action "produce p". If the action was not done the state of the world would be ~p instead of p. The list of elementary actions can actually be expanded with four additional action types not shown in the table. These actions would correspond to actions where the agent refrains from intervening with the world. The total number of elementary actions is accordingly eight. The four (eight) elementary action types define a generic set of actions which have the great advantage of being defined on a logic basis. This means that the completeness of the action types is ensured. The elementary action types (Table 1) form therefore a very attractive basis for the definition of concepts for modelling system functions. Note that the action types are generic because they are defined without specifying the proposition p. The action types can therefore be specialised to specific problem domains by proper specification of p.

Another remarkable aspect of the action types is that they have a direct correspondence with the types of control functions used in control engineering (Table 2). The completeness of the action types implies accordingly that any control function can be described by proper combinations of these four functions! Note that the descriptions of the controls do not represent the implementation of the controls. The descriptions only define the control purpose.

Table 2. The elementary action types

Elementary action	Control action
Produce	Steer
Maintain	Regulate
Destroy	Trip
Suppress	Interlock

3.2. A language for modelling functions of microbial systems

The elementary action types can be used as a systematic basis for derivation of modelling concepts for a particular problem domain. As an example Multilevel Flow Modelling (MFM) (Lind, 1994) can be mentioned. A basic set of modelling concepts adapted to the domain of microbial systems is proposed in Fig. 1. Each of the actions shown can be defined formally as specific interpretations of the elementary action types or as compositions of two elementary actions. We will not go into all details here. Instead, we prefer to demonstrate with an example (see 4) how the modelling concepts can be used to represent the functional organisation of microbial systems.

The MFM modelling language (Fig. 1) comprises three types of concepts. It contains a set of concepts for representing action (functions), concepts to represent goal states and a set of concepts for representing means-end relations between actions, sets of actions and goals. It should be stressed that MFM represents the actions or transformations done to material, energy or information flows (fluxes) in a complex system. However, it does not represent the flows or fluxes themselves. This may seem disturbing, but the abstractions provided by MFM describe how the systems of transformation of the various substances (energy, material or information) are organised into means-end networks. The levels of abstraction can therefore not be defined without implicitly thinking in concepts of flows or fluxes. It should be noted that MFM also includes concepts to model part-whole relations, as well as concepts to model relations between functions and physical structures, but these relations are not used here (see Lind (1999) for more detail on these relations). A deeper understanding of most of the concepts presented in Fig. 1 can be obtained by a study of the application example presented in 4.

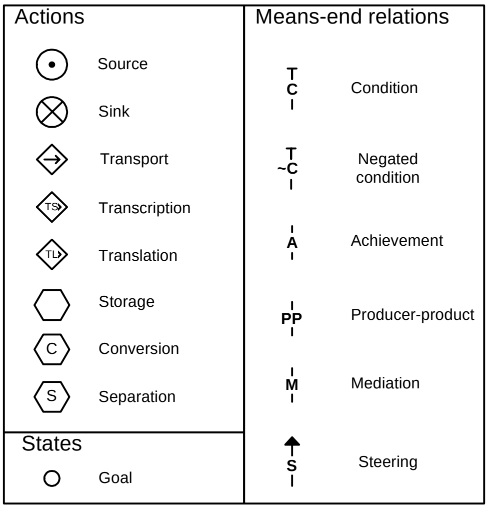

Figure 1. MFM modelling concepts adapted to the microbial domain

4. MEANS-END ANALYSIS APPLIED TO THE *LAC* OPERON

4.1. Induction of the lac operon

The mechanism that plays a role in transcription of the *lac* operon in the absence of extra-cellular glucose is induction (Fig. 2 and 3). The *lac* operon consists of 3 structural genes (Fig. 2), containing the genetic code for enzymes responsible for the uptake and the conversion of the substrate lactose into its building blocks glucose and galactose. In a simplified representation (Fig. 2) the structural genes are preceded by one operator and one promoter. In the absence of extra-cellular glucose and lactose, the *lac* operon is repressed. The repression of the *lac* operon originates from the presence of a 4th gene, containing the genetic code for a repressor protein. This *lac* repressor gene, or regulatory gene, has its own promoter (P_i in Fig. 2) allowing RNA polymerase to bind to P_i and to transcribe the *lac* repressor gene. The ribosomes translate the *lac* repressor mRNA, to form the *lac* repressor protein. In the absence of lactose, the *lac* operon is repressed, meaning that the *lac* repressor protein is bound to the operator region of the *lac* operon, preventing the RNA polymerase to bind to the promoter of the structural genes, and thus repressing the transcription of the structural genes (Fig. 2).

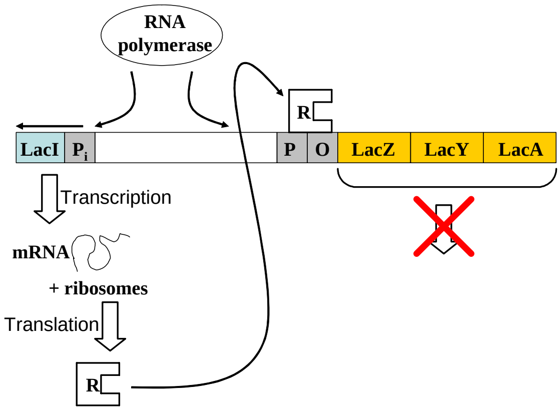

Figure 2. Repressed *lac* operon (Yildirim and Mackey, 2003). LacI = repressor protein gene; LacZ = ß-galactosidase gene; LacY = ß-galactoside permease gene; LacA = ß-galactoside transacetylase gene; O = operator; P = *lac* operon promoter; P_i = promoter repressor protein; R = repressor protein

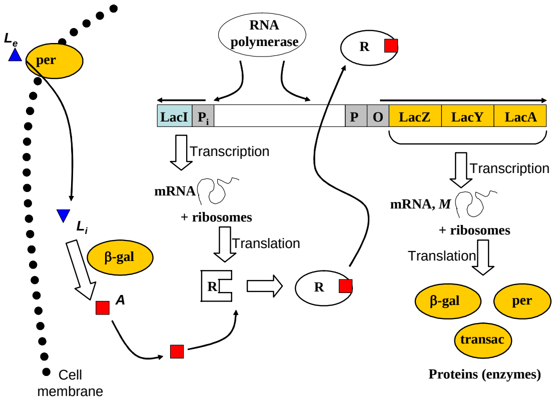

Figure 3. Induced *lac* operon (Yildirim and Mackey, 2003). L_e = extra-cellular lactose, L_i = intra-cellular lactose, A = allolactose; ß-gal = ß-galactosidase; per = ß-galactoside permease; transac = ß-galactoside transacetylase

Allolactose is the inducer of the *lac* operon, and results from the intracellular conversion of lactose following uptake trough the cell membrane (Lengeler *et al.*, 1999; Yildirim and Mackey, 2003). Indeed, in the absence of extra-cellular glucose and when lactose is present in the growth medium, lactose is transported into the cell by the ß-galactoside permease (Fig. 3). Intra-cellular lactose is subsequently converted into glucose, galactose and allolactose. The *lac* repressor protein undergoes a conformational change after binding the inducer allolactose, and is then no longer capable of binding to the operator region of the structural genes (Fig. 3).

RNA polymerase can now bind to the promoter of the structural genes and produce mRNA, which is subsequently converted into proteins by the ribosomes. This induction mechanism of the *lac* operon is a positive feedback loop: Increasing intra-cellular lactose concentrations will lead to an increase of the expression of the *lac* operon, and thus result in an increased production of for example permease enzyme molecules which will again lead to increased intra-cellular lactose concentrations, until the maximum protein production rate is reached. Depletion of extra-cellular lactose will result in repression of the *lac* operon.

4.2. Functional modelling of the lac operon induction

Fig. 4 represents the induction mechanism of the *lac* operon using the symbols introduced in Fig. 1. The logic of the model can be explained by starting with the bottom part of Fig. 4, which represents the uptake of lactose and the conversion of lactose to allolactose, the inducer of the *lac* operon. The model shows in box I that the transport of lactose over the cell wall is carried out ('mediated') by the ß-galactoside permease, which is produced as the result of the translation of the mRNA (top part of the model). The subsequent conversion of lactose to allolactose is catalysed ('enabled') by the β-galactosidase. The allolactose is afterwards assumed to be distributed in the cellular cytoplasm by diffusion (modelled as a 'transport function'). The functions described comprise the means for achieving (A in Fig. 4) the conditions for allolactose to be present in the cell.

By changing perspective, and thus moving upwards in the model, the set of functions in boxes II and III in Fig. 4 are now considered, describing how the state of the repressor (R) is influenced by the various functions of the microbial system. The principles to describe the production of repressor protein in detail (box II) are similar as for the flow model in box IV, representing the transcription and translation of the structural genes of the *lac* operon: The lacI gene is transcribed, and the resulting mRNA is translated into the repressor protein. The function in box II provides the presence of R; in other words it provides the means for producing the repressor protein R.

The repressor protein can follow 3 possible paths (box III): 1) It can follow the bottom path where it binds with the operator of the *lac* operon, and will thus block transcription; 2) It can follow the middle path, where it binds with the inducer allolactose, and undergoes a conformational change; 3) It can follow the upper path, where the repressor protein is degraded (modelled as a 'Sink'). The presence of allolactose conditions two of the transports in box III. When allolactose is present the repressor protein binds to the allolactose. In the absence of allolactose the repressor protein binds to the *lac* operon operator. In the first case, the functions described in box III comprise the means for achieving a de-repressed *lac* operon, and thus transcription of the *lac* operon structural genes occurs. Again, we can now move upwards in the model: The transcription and translation of the *lac* operon is represented in box IV. The *lac* operon structural genes are transcribed into a polycistronic mRNA, and during the subsequent translation process the mRNA results in the different proteins (ß-galactoside permease and β-galactosidase). Conversion of the polycistronic mRNA to several proteins is modelled by combining the translation process with a subsequent separation function in Fig. 4. Note that production of the third protein encoded in the *lac* operon, ß-galactoside transacetylase, is not shown in Fig. 4, since that enzyme is assumed not to play any significant role in the induction mechanism. The diffusion of both proteins into the cytoplasm is finally illustrated in box V and VI. Note that ß-galactoside permease is a mediator (transport enzyme), whereas the role of β-galactosidase is to be an enabler (a catalyst). Now the loops to box I are closed.

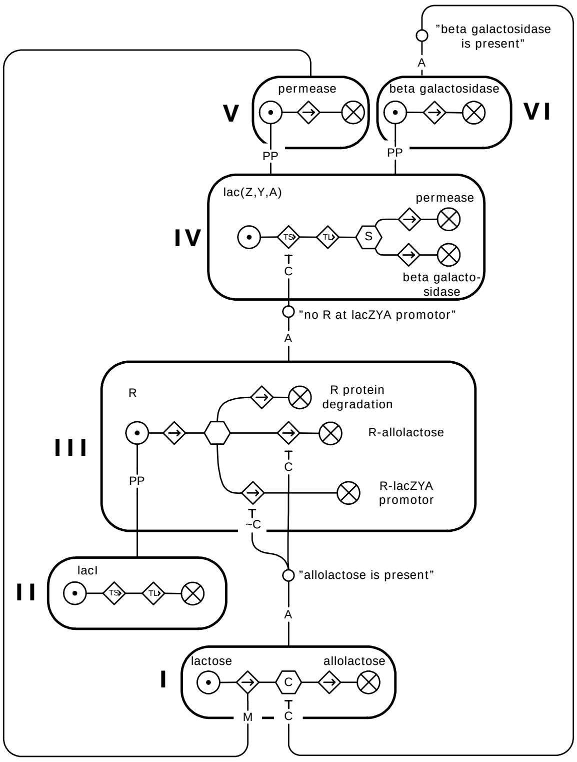

Figure 4. MFM representation of the *lac* operon induction mechanism (see also Fig. 2 and 3)

5. DISCUSSION

To reveal and understand the regulatory network mechanisms constitutes one of the most significant scientific challenges in the post-genomic era. From a production process perspective, it is an interesting question how systems engineers should couple the detailed description and understanding of the functioning of microorganisms (the micro scale) to the higher process level descriptions (the macro scale). The proposed MFM modelling based methodology is especially suitable to support this coupling of the microbial regulatory functions and the higher level process and production control functions, since the same set of symbols might be used to represent the flows at the process as well as at the detailed (micro) level. This ability distinguishes the proposed methodology from existing methods to represent regulatory network mechanisms: the actions and means-end relation symbols (see Fig. 1) provide a high degree of transparency on the way system states interact with each other.

The applied modelling concept has been demonstrated to enable modelling the changes in qualitative behaviour of microorganisms, and is as such able to summarise available process knowledge. If quantitative dynamic models were desired, then these could be developed within each region of qualitative behaviour, using the logic in the MFM model as a support in the generation of detailed mathematical descriptions.

The final result of applying process engineering might be improved considerably when process-relevant parts of the intracellular regulatory networks are understood better, and the methodology proposed in this paper can significantly contribute to representing and subsequently expanding that understanding. Maybe most importantly, applying the MFM modelling methodology to regulatory networks in microorganisms almost naturally leads to modularising the network into elemental building blocks that are understandable for systems engineers as well as biologists. Thus, the proposed modelling method could contribute substantially to the systems biology field by providing a formalism that allows biologists and systems engineers to communicate efficiently about regulatory network functions.

The proposed description, which is based on MFM (Lind, 1994), might eventually lead to combining basic understanding of microbial behaviour with the semiotics of control. This combination leads to simple schematics for describing fundamental roles of molecules in cells, and their reactions for control and coordination of microbial behaviour. In this respect, the flexibility of the MFM modelling formalism is especially noteworthy: In fact, in the *lac* operon example the lacI gene is expressed constitutively. This means that transcription and translation of the gene to the resulting repressor protein does not necessarily have to be modelled in detail in box II in Fig. 4. Indeed, since we assume that no regulatory mechanisms are involved in this process, the presence of the repressor protein could have been modelled by only including a ‘source’ for the repressor protein in box III, thereby omitting box II from the model. Thus, MFM models are flexible and can be extended easily. This can for example also be illustrated by the straightforward extension of the *lac* operon induction mechanism (Fig. 4) to also include the glucose effects (results not shown).

The proposed modelling methodology is not only useful in reverse engineering, where it could be applied to represent hypotheses on the operation of complex regulatory network systems. To our opinion MFM models could also be used in forward engineering, to design regulatory network building blocks before developing a detailed mathematical description.

6. CONCLUSIONS

Means-end analysis allows breaking down complex regulatory networks into elementary building blocks. The resulting models can be extended easily. The formalism enables a straightforward communication between systems engineers and biologists on the mechanisms underlying regulatory functions.

REFERENCES

Cheetham P.S.J. (2004) Bioprocesses for the manufacture of ingredients for foods and cosmetics. Adv. Biochem. Engin./Biotechnol., **86**, 83 – 158.

Doyle F.J. III (2004) A systems approach to modeling and analyzing biological regulation. Proceedings of the International Symposium on Advanced Control of Chemical Processes (ADCHEM2003), 11 - 14 Jan. 2004, Hong Kong.

Lengeler J.W., Drews G. and Schlegel H.G. (1999) Biology of the prokaryotes. Thieme Verlag, Stuttgart (Germany)/Blackwell Science, Oxford (UK).

Lind M. (1994) Modeling goals and functions of complex industrial plant. Applied Artificial Intelligence, **8**, 259 – 283.

Lind M. (1999) Plant modeling for human supervisory control. Trans. Institute of Measurement and Control, **21**, 171 – 180.

Lind M. (2004) Means and ends of control. Proc. IEEE Conf. Systems Man and Cybernetics, October 10-13, The Hague, Holland.

Von Wright G.H. (1963) Norm and action – A logical enquiry. Routledge and Kegan Paul, London.

Yildirim N. and Mackey M.C. (2003) Feedback regulation in the lactose operon: A mathematical modeling study and comparison with experimental data. Biophys. J., **84**, 2841 – 2851