Handling Contradicting Requirements Using Desirability Functions

Zsolt Réthy, Zoltán Koczor, József Erdélyi

University of West Hungary, Ady Endre u. 5, 9400 Sopron, E-mail: rethyzs@wla.hu

Budapest Tech, Doberdó út 6, 1034 Budapest

Abstract: Desirability functions serve to turn multiple response optimization problems into single response problems. A new set of these functions will be introduced within this paper that are based on an analogous probabilistic model for predicting the optimal starting time of a process to meet a given time schedule. We use loss-function-like functions and the probability of not meeting a given target to find the optimal set of input parameters where statistical properties are as we expect.

1 Introduction

Optimizing parameters is a critical issue during the development of any product. For this reason several methodologies have been worked out, most of them multi-criteria decision making techniques. A special set of functions called desirability functions [1][2][3] has been developed for doing this as easily as turning the multiple response problem to a single response problem. We aim to give an alternative from a different point of view, namely for cases where we are able to define loss functions for not meeting the specification limits, and we have information about the distribution of each response variable.

2 Calculating Risk for a Production Time Schedule with Given Loss Functions for Failing to Meet the Specification Limits

Risk can be defined for every activity, as can the processes and decisions we are responsible for. Generally speaking, risk is a function of the probability of an event and the causes of the event: $R(A) = P(A)L(A)$ (1)

where R is the calculated risk, P is the probability of the event A, and L is the calculated total loss [4].

If we define a certain loss or a loss function for exceeding the specification limit for a response variable, and calculate the distribution of the parameter, then we have the probability of meeting the specification limits. So the loss function multiplied by the given probability is the risk function for the parameter.

For the production time schedule we can define two mutually contradicting economic requirements:

- the economic consequences of the delay when we cannot make the time schedule and
- the costs of the early stockpiling and of the additional resources needed for the former.

These requirements can be represented as loss functions.

We can assume that these losses can be calculated simply. The total time, and so the deadline for the production, can be calculated i.e. from process and logistical times. However, there is a certain variation around this time point that we assume to be a normal distribution.

The risk function can be calculated as follows:

$$L(t^*) = L_1(t)\left(1-\Phi_{t,s}(T_{kr})\right) + L_2(t)\int_{-\infty}^{T_{kr}}\varphi_{t,s}(t)dt = L_1(t)\left(1-\Phi_{t,s}(T_{kr})\right) + L_2(t)\Phi_{t,s}(T_{kr})dt \tag{2}$$

where

- $\Phi_{t,s}$ is the distribution of the total time with t^* mean and s standard deviation $\left(t \in N(\mu,\sigma)\right)$
- $\varphi_{t,s}$ is the density of the total time with t^* mean and s standard deviation
- Tkr – the nominal total production time (upper specification limit)
- L1(t) – the cost of exceeding the specification limit
- L2(t) – the cost of premature production

We use the fact that the probability of being below Tkr is

$$P(t < T_{kr}) = \Phi_{t,s}(T_{kr}) \tag{3}$$

Assuming the same process flow for different start times, the t^* mean can be changed by changing the start time. The risk function can be represented graphically as Figure 1 shows. The optimal start time will be the one where risk is minimal, i.e. minimizing the risk function.

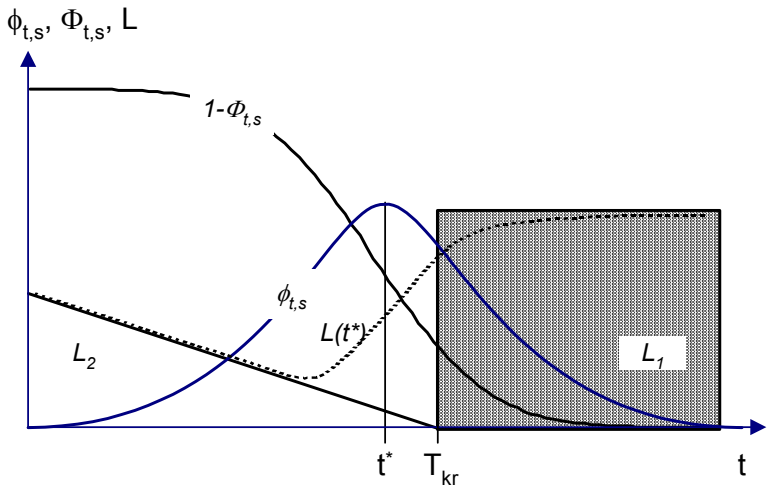


Figure 1

Loss functions and the density function showing the uncertainty of the deadline

3 Defining Desirability Functions Based on Loss Functions

Consider that we have a process with a number of inputs and a number of response variables. The distribution of each response variable is known, and some of them have lower and/or upper specification limits, while others have only a target value defined.

Similarly to the model shown in Figure 1, we can create functions that represent the loss when moving away from the target and the probability of not meeting the target. The latter has the advantage of taking not only the mean but also the standard deviation into account. A couple of papers already handle this in different ways e.g. [5] defines desirability functions based on the mean and deviation to bind the desirability levels to Six Sigma quality levels.

We would like to handle the newly created functions as desirability functions, with the assumption that the response variables are independent and are distributed normally.

First we have to collect information about the process

to have an estimate for the function between the input parameters, denoted x, and the responses – denoted $\hat{y} = f(\mathbf{x})$ [6];

- to have an estimate for the standard deviation of the responses.
- Further to this, we have to
- define the target value (T) for each response variable;
- define a helper desirability function (δ-function) for each response variable;
- and define a d-function for each response variable based on the δ-function and the probability of not meeting the target;
- both the δ-functions and the modified d-functions should fulfil some requirements such as the image set should be $[0,\ 1]$.

Similarly to the definition of risk above, we define the desirability as the product of the desirability helper function (based on loss functions but in the opposite manner) and a probability value.

This means that similarly to the previous scheduling model, the loss function has its minimum in the target. However, that means in our interpretation, that the helper desirability function has its maximum at the target.

We define a simple linear helper function (see Figure 2) for each side of the target (this is similar to the desirability function of [2]):

$$\delta_{il}(\mathbf{x}) = \begin{cases} 1 + k_{il}(\hat{y}(\mathbf{x}) - T), & T - \frac{1}{k} \le y \le T \\ 0, & otherwise \end{cases} \tag{4}$$

for the left side and

$$\delta_{ir}(\mathbf{x}) = \begin{cases} 1 - k_{ir}(\hat{y}(\mathbf{x}) - T), & T \le y \le T + \frac{1}{k_{ir}} \\ 0, & otherwise \end{cases} \tag{5}$$

for the right side, where k_{il} and k_{ir} are constants. For these constants, we can take into account some multiple of the standard deviation.

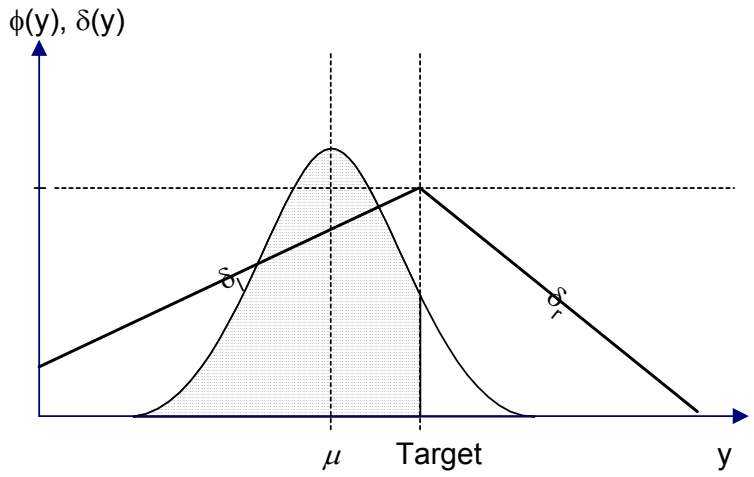


Figure 2

Linear helper functions and the density function of a response variable

Alternatively, we can use helper functions based on Taguchi's quality loss function.

While the traditional quality philosophy binds the loss to the point where a quality characteristic exceeds a specification limit, [7] defines a quadratic function for the loss so that it is only zero if we meet the target value exactly. The quality loss function looks as follows:

$$L(x) = k(x - T)^2 \tag{6}$$

So the helper functions based on this would be (see Figure 3):

$$\delta_{Ti}(\mathbf{x}) = 1 - k_i\left(\hat{y}(\mathbf{x}) - T\right)^2; \tag{7}$$

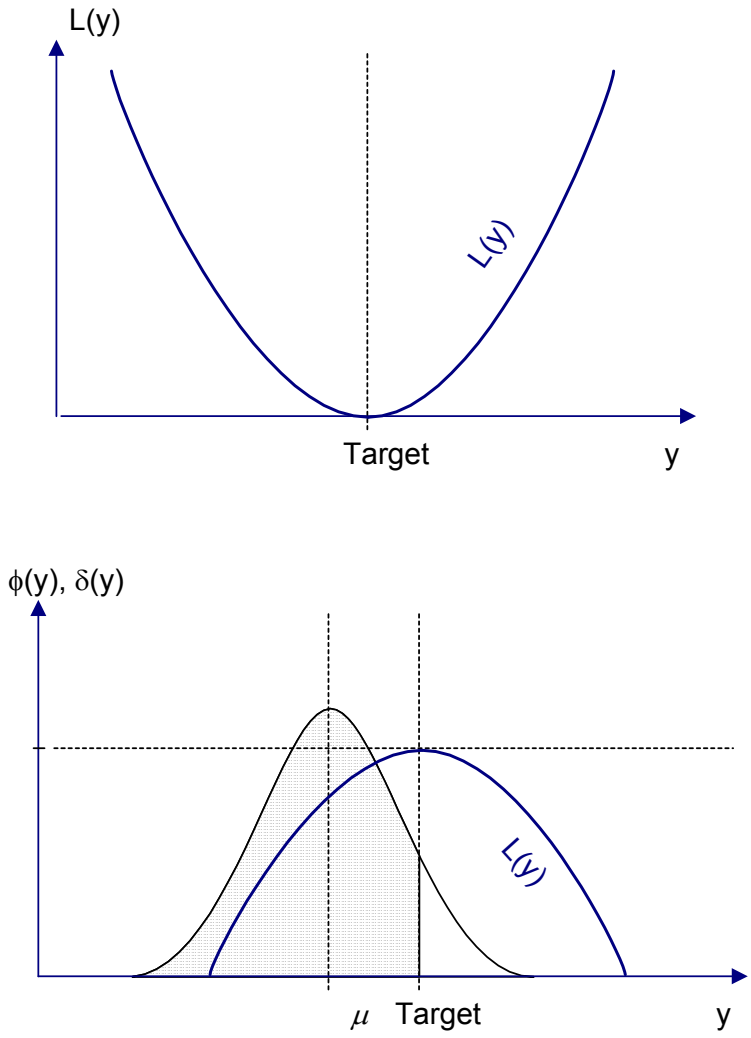


Figure 3:

The Taguchi quality loss function (above) and the helper function created for a response variable (below)

The desirability function will be the sum of each side's desirability:

$$d_i(\mathbf{x}) = \delta_{il}\Phi_{\mu,\sigma}(T) + \delta_{ir}\left(1 - \Phi_{\mu,\sigma}(T)\right) \tag{8}$$

where μ is the mean and σ is the standard deviation. In order not to lose generality, the original desirability functions introduced in [2] can be used for the responses where it's more appropriate e.g. for responses with lower/upper specification limits.

Calculation of the composite desirability

In [6] a calculation with individual weights is proposed for each desirability function rather than using *y*-*d* value pairs for weighting. This latter is more difficult to use and interpret.

$$D_{Derr} = \sqrt[S]{\prod_{i=1}^{q} d_i^{w_i}}, \tag{9}$$

where

$$S = \sum_{i=1}^{q} w_i \tag{10}$$

Based on this calculation method, we also use individual weights and the composite desirability should be maximized.

$$D_{\delta} = \sqrt[S]{\prod_{i=1}^{q} d_i^{w_i}}, \tag{11}$$

Once we have the maximized value, the optimal input parameters can be calculated from it.

Conclusions

Given the generality of the problem a widely usable model can be built that gives the correct answers. As part of our research we created an application that includes expert system features to get an optimal solution. The properties of this application are:

- A standard interface for data input.
- A data exclusion mechanism that "forgets" given extreme values and archives older data - enabling to describe process trends.
- Stores older data sets in the database.
- Practice-oriented outputs.

The usability of the application was tested and proven on a real-life example in practice.

References

[1] Harrington Jr., E.C.(1965): "The desirability function". Industrial Quality Control 21 (10) p. 494-498.

[2] Derringer, G. and Suich, R. (1980): "Simultaneous Optimization of Several Response Variables". Journal of Quality Technology Vol. 12. No. 4. 1980, p. 214-219.

[3] del Castillo, E. ; Montgomery, D.C. and McCarville, D.R. (1996): „Modified desirability functions for multiple response optimization". Journal of Quality Technology Vol. 28(3) p. 337-344.

[4] Koczor, Z. ; Marschall, M; Némethné Erdődi, K. and Réthy, Zs. (1996): "Quality assurance methods to determine the optimum of risk based on the product's characteristics and the information on manufacturing". Anyagvizsgálók Lapja 1996/4. p. 123-126.

[5] Ribardo, C. L. (2000): „Desirability functions for comparing arc welding parameter optimization methods and for addressing process variability under Six sigma assumptions". Dissertation, The Ohio State University, Columbus, OH.

[6] Derringer, G. C. (1994): "A balancing act: optimizing a product's properties". Quality Progress, June 1994, p. 51-58.

[7] Taguchi, G. (1986): „Introduction to quality engineering". Asian Productivity Organization, Tokyo.

Chalcogenide Materials for Solar Energy Conversion

Péter Turmezei

Institute of Microelectronics and Technology, Budapest Tech
Tavaszmező u. 17, H-1084 Budapest, Hungary

Abstract: The problem of electrical energy storage can possibly be solved with the help of electrochemical solar cells, which are suitable to generate either electrical energy or hydrogen gas under special conditions. The greatest problem of the electrochemical solar cell technology is to find novel materials which have appropriate properties for electrochemical energy conversion. In this work Cd_4GeSe_6, a novel material for electrochemical solar cells, will be presented.

1 Introduction

Solar cell technology is a very developed area of microelectronics; however, there are still some problems under research. One of the greatest problems in solar cell applications is the storage of electrical energy. This problem can possibly be solved with the help of electrochemical solar cells, which are suitable to generate either electrical energy or hydrogen under special conditions [1]. The technology of electrochemical solar cells has some technical and scientific problems. One such problem is photocorrosion, which occurs at the electrolyte–semiconductor interface. Photocorrosion damages the semiconductor electrode during the operation of the solar cell. This problem can be solved by avoiding the photocorrosion effect with the help of dye sensitized nanocrystalline TiO_2 material. The band diagram of the dyesensitized photoelectrochemical solar cell is shown in Fig. 1. A possible direction of this research is the search for novel materials with appropriate properties for electrochemical applications. One of the important groups of such semiconductor compounds is the chalcogenides such as Cd_4GeSe_6.

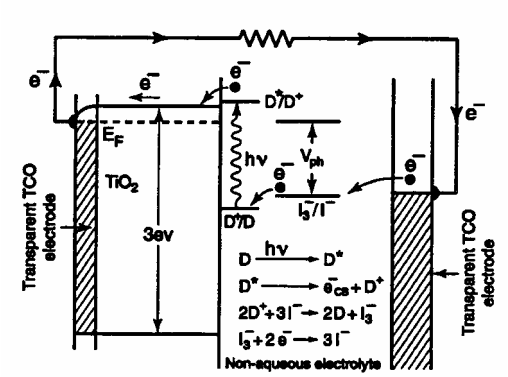


Figure. 1. The band diagram of the dyesensitized photoelectrochemical solar cell

In this work Cd_4GeSe_6, a novel material for electrochemical solar cells, will be presented. The properties of this material will be investigated, which has been scarcely done before, and that is why these properties are not known in detail. Cd_4GeSe_6 belongs to the agryrodite family, of which lattice parameters were determined [2]. The band gap and type of band transition was determined by absorption and the I-V characteristics was determined by photoelectrochemical method [3]. Furthermore it was found that this material shows very good resistivity against photocorrosion [4]. The knowledge of the electrical parameters of the Cd_4GeSe_6–electrolyte junction is very important for solar cell applications. It was also determined in this work. The properties of the Cd_4GeSe_6 crystal–electrolyte junction are investigated with impedance analysis. The evaluation of the measured data was carried out with the help of a computer program developed by us in Pascal language. We used an equivalent circuit with physical meanings, this circuit was appropriate for the calculations [5].

2 The Chalcogenide Material

A possible direction of semiconductor research is the search for novel materials with appropriate properties for different applications. One of the important groups of such semiconductor compounds is the group of chalcogenides, a well-known example of binary compounds. They are good photoconductors and have high absorption coefficient. Material properties can be improved and modified by forming ternary, quaternary etc. compounds of the above. Ternary chalcogenid materials, such as Cd_4GeSe_6, were synthesized in which new covalent chemical bonds appeared. Due to these covalent bonds these materials show higher resistance against corrosion. This novel property in itself makes novel applications, such as photoelectrochemical energy conversion electrode, possible.

In this work the properties of Cd_4GeSe_6 are investigated which are until now scarcely studied and therefore not known in details. The existing data in the literature differ over a wide range even for fundamental material parameters such as lattice parameters, band gap or type of band transition. This material belongs to the agrirodite family, which belongs to the monoclinic crystal class. The structure of chemically analogous compounds was investigated earlier [6]. The optical parameters of Cd_4GeSe_6 were scarcely studied possibly because of the difficulty of making larger pieces of single crystal.

The synthesis of Cd_4GeSe_6 crystal can be carried out from chalcogenide and dichalcogenide sources. The crystallizing period is several weeks long. The Cd_4GeSe_6 is a stable crystal and keeps its stability even at high temperatures under normal atmospheric conditions.

3 Photoelectrochemical Investigation

The band gap was determined by absorption [3] and photoelectrochemical [7] methods and was found to be 1.7 and 1.75 eV respectively. A further reference [8] gives 1.5 eV band gap and indirect band transition. Ref. [9] gives 1.9 eV for band gap from photoluminescence measurement at 10 K. These strongly different photoelectrochemical and photoluminescence results are refined in Ref. [3].

The impedance measurements were performed in an electrochemical cell under potentiostatic control. The electrolytes were 0.05 M H_2SO_4 and solution. The impedance analysis was carried out with the perturbation of some mV. The modeling of an electrolyte– semiconductor junction is a difficult problem because the values of the circuit elements exhibited frequency dependence. In this work we determined the proper values of equivalent circuit components with their physical meaning for the transfer function of the junction. The parameters of the equivalent circuit are very important to know for device applications. A simple equivalent circuit with physical meaning was appropriate for the calculation [5]. It contains three parallel branches, one branch is a resistance R_1, the second branch is a swinging circuit (R_2C_2) and the third branch is a capacitor (C_3).

The evaluation of the measured data was carried out with the help of a computer program developed by us in Turbo Pascal language. The transfer function of the equivalent circuit has three solutions (one zero and two poles). In the first step these three roots were fitted in the same time with the help of the least square method. The minimum of the error is determined with the help of gradient method, until the error becomes less than 1 Hz. The value of the constant in the transfer function was determined from the amplitude diagram with similar method [10]. The R_1 resistance represents the charge transfer that is the electrochemical reaction at the interface. The value ranges from 6.2 to 7.1 $k\Omega/mm^2$. R_2 and C_2 represent the surface levels and deep centers where the values range between 0.6 and 1.6 $k\Omega$

/mm^2 and between 5.3 and 7.6 nF/mm^2, respectively. The element C_3 means the space charge capacitance and its value is between 0.62 and 0.78 nF/mm^2 without bias voltage. The space charge capacitance shows a little larger value in KOH then in H_2SO_4 solution. Measured and fitted amplitude and phase diagrams of the junction are shown in Fig. 2.

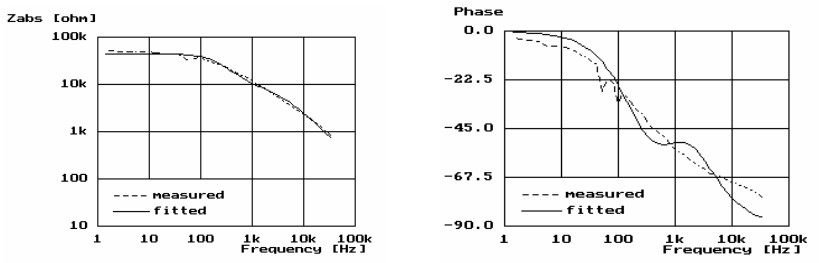


Figure 2. The amplitude and phase diagrams of the Cd_4GeSe_6 and 0.05 M H_2SO_4.

The junction of electrolyte–Cd_4GeSe_6 crystal was investigated with impedance analysis. We set-up an equivalent circuit of this junction. The electrical parameters of the junction were determined which are very important to know for device applications. The space charge capacitance was found to be about 0.9 nF/mm^2. The charge transfer resistance was about 5 kΩ/mm^2. The values of the elements of the RC circuit which represent the surface levels are 1.7 kΩ/mm^2 and 1.5 nF/mm^2, respectively. This model describes the electrical behavior of the junction in the whole investigated frequency range. Furthermore the surface morphology of Cd_4GeSe_6 crystal was investigated after electrochemical treatment [4].

References

[1] Á. Nemcsics: Solar Cells and their Developing Perspective; Academic Publisher, Budapest 2001

[2] K.-F. Hesse, M. Czank and Á. Nemcsics; Z. Kristallogr. 216**,** (2000) 14

[3] I. Kovách, Á. Nemcsics and Z. Lábadi; Inorganic Materials, 39 (2003) 108

[4] Á. Nemcsics; to be published

[5] Á. Nemcsics; phys. stat. sol. (a), 173, (1999) 405

[6] P. Quenez and O. Gorchov; J. Cryst. Growth 26 (1974) 55

[7] W. F. Kuhs, R. Nitsche, K. Scheunemann; Mater. Res. Bull. 14 (1979) 241

[8] B. von Krebs, J. Mandt; Z. Anorg. Allg. Chem. 388 (1972) 193

[9] P. Quenez, A. Maurer; J. Phys. 36 (1975) 83

[10] P. Turmezei, Á. Nemcsics; phys. Stat. Sol. (c) 0 (2003) 967

Behavior and Design Intent Based Product Modeling

László Horváth, Imre J. Rudas

John von Neumann Faculty of Informatics, Budapest Tech
Népszínház u. 8, H-1081 Budapest, Hungary
horvath.laszlo@nik.bmf.hu
rudas@bmf.hu

Abstract: A knowledge based modeling of mechanical products is presented for industrial CAD/CAM systems. An active model is proposed that comprise knowledge from modeling procedures, generic part models and engineers. Present day models of mechanical systems do not contain data about the background of human decisions. This situation motivated the authors at their investigations on exchange design intent information between engineers. Their concept was extending of product models to be capable of description of design intent information. Several human-computer and human-human communication issues were considered. The complex communication problem has been divided into four sub-problems, namely communication of human intent source with the computer system, representation of human intent, exchange of intent data between modeling procedures and communication of the represented intent with humans. Paper discusses the scenario of intelligent modeling based engineering. Then key concepts for the application of computational intelligence in computer model based engineering systems are detailed including knowledge driven models as well as areas of their application. Next, behavior based models with intelligent content involving specifications and knowledge for the design processes are emphasized and an active part modeling is proposed and possibilities for its application are outlined. Finally, design intent supported intelligent modeling is discussed.

1 Introduction

Model based development of products uses advanced process-centric digital product and manufacturing process definition techniques. Shapes, parts, assembly relationships, joints, tolerances, finite element related parameters, cutting tool paths for computer control of machining and product structures are main objects to be described in feature and associativity driven product models. Products are positioned for e-business among others by the help of using model-based approach. Scope of engineering modeling is being extended to the entire life cycle of products. Comprehensive application of virtual technology is the essence of the

concept of digital enterprise where all engineering activities and equipment controls are done within a highly integrated modeling system.

Key participants of engineering processes are still skilled engineers who are sitting at their computers with advanced user communication features. Engineers expect more and more computer assistance at their decisions in changed industrial environments where quick and efficient engineering decisions are needed to survive competition in the market. However, modeling methods that really enhance human decision assistance capability of engineering modeling procedures during interactive part modeling sessions are not available. Powerful decision assistance use analyses that need behavior based modeling of engineering objects in various circumstances. In other words, analysis of object behavior is done in virtual. The authors investigate possibilities and means of new enhancements in knowledge based active modeling procedures.

Advancements in part related engineering modeling is motivated both by expectations for high performance parts with well-engineered shape. Performance of modeled objects can be assessed by using of various implementations of finite element analysis. Advanced shapes are best produced by advanced surface model driven and computer controlled manufacturing of parts or tools for making parts [6]. Stand alone part model objects are integrated into product models by using of structure descriptions and associativity definitions. Results of the part design are protected against modification by the definition of shape, dimensions and associativities as constraints.

Despite these fantastic advancements some important aspects of modeling could not follow this evolution. One of them is application of active models. The above outlined scenario is a good starting point for development of virtual environment based engineering modeling where advanced knowledge technology is integrated with advanced product modeling technology. The purpose of this paper is to give an outline of an advanced concept and a modeling method for intelligent model features in engineering models. This modeling is suitable for implementation in industrial CAD/CAM systems. Preliminaries of the reported research are developments in methodology to integrate manufacturing process modeling with form feature based part modeling by using of relationship and constraint definitions [1].

Design intent covers results of a complex human thinking process. Modeling tools in present day CAD/CAM systems are not suitable for description of this process. Advanced, customer oriented product design requires frequent modification of earlier decisions. Moreover, modification of result of a decision often causes a need to modification of results of other decisions. This is impossible in the lack of proper background information for the original decision. Consequently, modifications of original decisions require contribution of the original decision-maker or other authorized engineer. Because live communication between engineers typically is not available, the only effective way is to describe their

intent in the product model to be modified. Inevitably, computer modeling should be extended to this area. The authors propose a computer method to assist the modification of decisions by modeling of design intent.

Paper discusses some issues of intelligent modeling based engineering. As an introduction, it outlines the scenario of the related activities and systems. Then key concepts for the application of computational intelligence in computer model based engineering systems are detailed including knowledge driven models as well as areas of their application. Next, behavior based models with intelligent content involving specifications and knowledge for the design processes are emphasized, an active part modeling is proposed, and possibilities for its application are outlined. Finally, design intent supported intelligent modeling is discussed.

2 Intelligent Content in Virtual Environments

An important assumption in advanced models of mechanical systems is that the model, in which new or modified features are defined, is created earlier using other modeling system and by other engineer. At the same time multiple designer operation mode in concurrent group work of engineers is assumed where several engineers handle a model. Both modeling procedures and designers utilize model-related knowledge in order to achieve effective product modeling process. Role of knowledge communication associated with model data communication is assistance of multiple designer and multiple modeling system related problem solving.

Knowledge content of model entities can be utilized in the design process at definition of product objects and simulations. An initial concept to integrate knowledge in modeling was to include knowledge to modeling procedures of CAD/CAM systems. This approach supports representation of generally applicable and domain related knowledge. However, most of the knowledge is company, product, even human related one and it changes from company to company and product to product. The only effective way is integration this knowledge in models of abstract or instance objects in the product model. This approach is anticipated to be one of the most researched and developed areas in model based engineering design during the next few years by the authors.

Significant part of knowledge necessary at later processing of a model is modeled object, model and modeling system specific (Fig. 1). This knowledge is not available at the application of the model so that it is to be exchanged with models between modeling systems. Built-in knowledge then helps engineers at development, modification and application of the model. It also prevents model quality from deterioration at its later application and modification. Knowledge

content and other advanced features resulted modeling where models can answer most of important questions about the modeled real world object before, during and after their manufacturing. Recently, advanced modeling systems that include models of this kind are called as virtual environments**.**

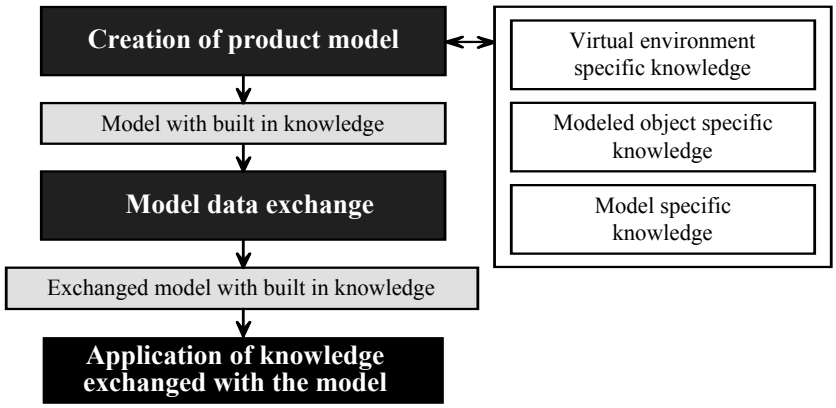


Figure 1
Knowledge in engineering modeling

Model with the capability of reaction using behavior related knowledge acts an intelligent design of the modeled object that can communicate built in knowledge with modeling procedure or human to save earlier decisions and human intent while new decisions and intents are captured in the model. Knowledge content of model is developed with its development. Behavior based models with intelligent content involve specifications and knowledge for the design processes (Fig. 2). Specifications are results of design with appropriate explanations. Model of design intent is considered to be described as specification in [2]. Knowledge normally is related to given specification but it is also can be independent of any specification. Definition of specifications and knowledge needs authorization according to role of engineers in the product development team and stage of the design process.

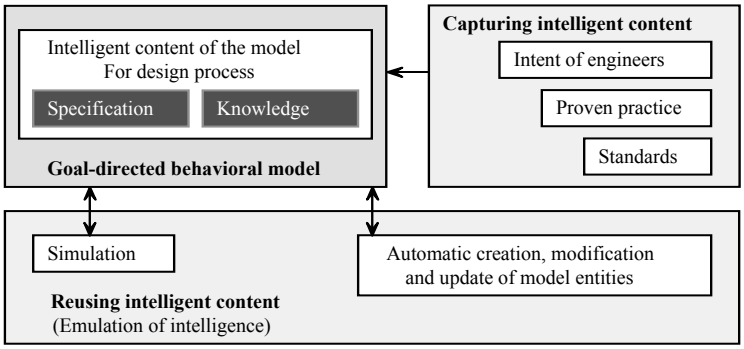


Figure 2
Knowledge assisted modeling

Forthcoming development and application of the model utilize its intelligent content at automatic creation, modification and update of model entities. Representation of knowledge should be as simple as can be so that it is easy to define by engineers in their every day practice. Most appropriate forms of knowledge are formulas, rules and checks because these are natural in engineering design. Compliance of the model with proven practices and standards can be ensured. Behavior based modeling offer conversion of implicit engineering practice into explicit knowledge.

Creating a new model is enhanced by application of models of abstract objects. An abstract object carries characteristics of a set of similar objects. At creating of a new modeled object instance from the model of an abstract object, actual characteristics of the instance object are set. This process can be automated by including knowledge in model objects. Model of an abstract object may involve domain, company and designer related knowledge. Model of an instance object generally contains domain, company, product and designer related knowledge. Information on origin and validation of the utilized knowledge should be included otherwise responsibility for the product can not be evaluated.

An advanced form of model objects in virtual environments is autonomous intelligent agent that is created by human or computer procedures (Fig. 3). Intelligent agents are autonomously working procedures in the software system with goal-directed behavior interacting with given environments [5]. Human control is realized directly by interaction with model creating procedures or indirectly by instruction or knowledge placed in agents. Behavior of agents is modeled in a multi-agent system. This modeling environment constitutes a reactive system. Reactive behavior of agents is controlled by appropriate creation of the model. Series of circumstances are identified and responded. Models are utilized in automatic or human controlled interactive, real-time simulations. Simulations are applied for analysis of critical situations and events during manufacturing and application of the modeled product. Simulation is a key technique for virtual prototyping where advanced modeling is applied to move physical prototyping activities into virtual environments. Real time assessment and analysis are assisted by appropriate intelligent procedures.

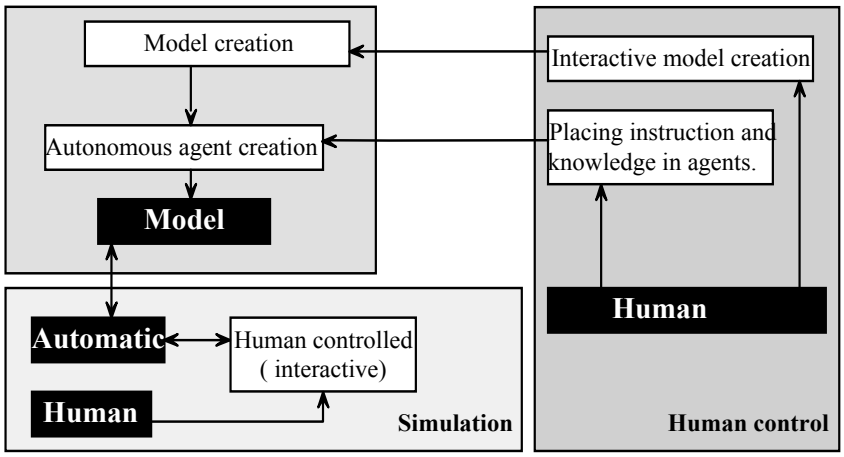


Figure 3
Model objects as agents

Goal-directed behavioral representation in agent-based modeling of engineering objects offers advanced simulation by emulation of intelligence. This is allowed by knowledge-driven modeling that captures and reuses intelligent content (Fig. 2). At the same time intuition, creativity and innovation of humans are also utilized in the course of their direct application, offered by interactivity, or by enriching the knowledge of agents. Enhanced competition-orientation of design engineering urges and stimulates application of behavioral techniques.

A virtual environment is developed specially for a problem area. Situations based on series of circumstances are analyzed at application of virtual environment for problem solving purpose. Virtual environment is used to determine influence of prevailing circumstances on some parameters in the model. Circumstances are created by humans or generated automatically in the virtual environment. In engineering practice there are two typical simulations. Best appropriate variants are selected or consequences of a decision are revealed for given set of circumstances. The first simulation allows revealing all parameters that influence a selected parameter. The second simulation allows determining impacts. It is best applied to design modifications. Finally, variants can be adapted or combined by engineers and the new variant can be analyzed in the virtual environment.

Advanced shape centered engineering design uses form feature driven shape-modeling [8]. Form features are elementary build blocks for shape models and act as modifiers for a previous shape. A sequence of shape modifications leads to the final shape of the part. Other non-geometric part and part manufacturing information including suitable and available manufacturing resources is mapped to form features. Typical shape representation of form features is unified topology-geometry. The authors have proposed application of marked Petri net with some extensions as representation of part manufacturing process model

features [4]. In the Petri net model a transition represents a setup or an operation process object. Manufacturing process model features are in associative relation with form features defined on the part to be manufactured.

Purpose of the reported research is getting more information about nature and characteristics of feature based product models then development of a unique active modeling approach and method. The research involves analysis of structure and behavior some typical features represented in models of mechanical systems.

3 Methodology for Modeling in Virtual Systems

Recent practice in modeling by CAD/CAM systems is definition of form features then attaching attributes and geometric model representations to them. In the author's approach the feature approach is extended and generic or instance part related knowledge is included in the model on a way that allows for its active application. Some previous background research activities by the author for modeling of part manufacturing processes, human-computer procedures and design intent are utilized. The method is appropriate for modeling on the basis of application oriented reference models.

One of the most effective methods for integration product model related partial models is definition of associativities between model entities [3] (Fig. 4.). Creation and modification of model entities rely on definition then maintaining of associativities. Associativities to be maintained are defined as constraints. Maintaining associativities at modification of models means propagation of the effect of changes in models. Propagation of any change of model at any stage of modeling makes whole product design consistent with intents, goals and decisions. Knowledge is often related to associativities so that it is beneficial to include it in associativity definitions. Modeling procedures generate associativity alternatives appropriate for the actual situation and offer them for humans in the course of interactive definition of models. This feature of modeling systems prevents erroneous associativity definitions by humans. Perhaps one of the best examples for application of associativities is assembly modeling in mechanical systems. As a typical method for automation of this activity, computer procedures propose the most appropriate constraints for part placement while human drags the part into position by a pointing device. Mechanical constraints are created then used to adjust part position and establish contacts automatically.

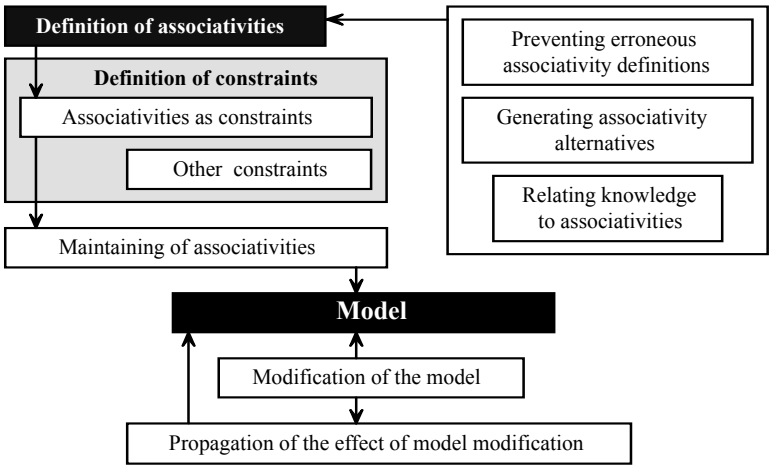


Figure 4
Associativities in knowledge based models

Creating and global modification of a multi-surface shape complex as a single surface while preserving design characteristics demands shape definition related knowledge both in modeling procedures and models (Fig. 5.). Taking styling, mechanical design and manufacturing knowledge and specification into consideration often results conflict to be resolved by the designer who is responsible for the related decisions. Knowledge acts according to the purpose and specification of modification. A typical purpose can be fitting a surface complex in a given solid model environment [7]. The related knowledge is represented in the form of rules, checks, control curves, etc. Input parameters as guiding surfaces, other outside world entities and digitized physical geometry are used by knowledge assisted surface modification procedures.

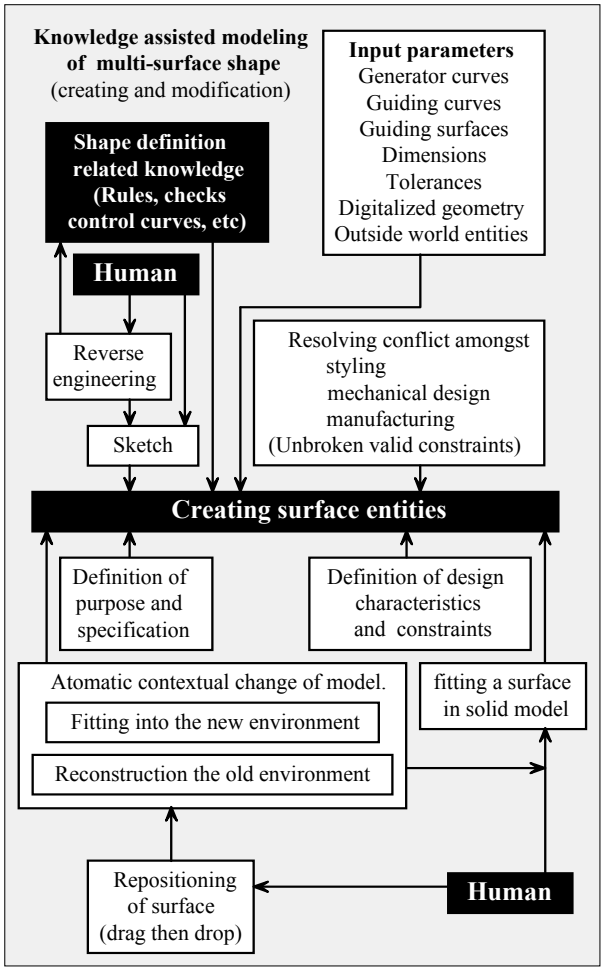


Figure 5
Knowledge based handling of complex surface features

Previously established design constraints valid for the surface to be modified such as point, tangent and curvature continuity must be unbroken during modifications. Existing topology and topology related knowledge is applied to propagate effects of surface modifications. Other important area for application of intelligent computing is recognition of sketched shape and creating surface model for it. Other surface related intelligent modeling can be utilized by reverse engineering to transform shape related knowledge from the physical world to virtual.

Repositioning by dragging then dropping of a form feature by human interaction of human during development of a part model is followed by its mathematically correct automatic fitting into the new environment and reconstruction the old

environment without any other human interaction. This feature of intelligent modeling is called as automatic contextual change of model. It is enabled by behavior based, reactive geometric model.

Other important area of knowledge based modeling is nonlinear mathematical optimization problem of mechanical parts by using of numerical algorithms. Mathematical programming optimizes design for design goal while satisfies specific design limits. Design limits, such as material strength or allowable displacements are functional requirements of the design process. Design goal represents the optimization intent such as cost, volume, time, mass, stress and displacement. Sensitivity analysis provides information the degree to which a change in each design parameter influences the structural performance. At adaptive analysis a converged analysis solution is achieved automatically.

4 Part Model with Active Characteristics

The authors proposed an active part model that is able to inform designers about consequences of creating a new, a modified or a new instance model entity. It comprises knowledge from three sources, namely modeling procedure, generic part model and designer. Because feature based part model is supposed to be applied, representation of feature related knowledge was analyzed. Modeling of a part is considered as a single process from conceptualization to manufacturing even to product life end procedure, according to the scope of product modeling. In the feature approach extended by active knowledge, comprehensive groups of features as volume adding and subtracting form features, form conditioning features, finite element features, load features, machining features, measurement features, associativity features, rule features and check features serve full feature orientation throughout the part related engineering process. Design alternatives, offered by humans or part modeling procedures, can be recorded in the part model together with the related knowledge.

New features can be launched for the modeling system both by human and remote created model. There is an actual set of known features in the modeling system at each moment (Fig. 6.). Most of the features are generic ones and part models include their instances. Others are defined only for the model under development and can or must not be applied in other models. A generic model can be applied generally or only by given individuals. Privacy policy is an important aspect at implementation of this approach. As an example, some features can be applied only within given projects.

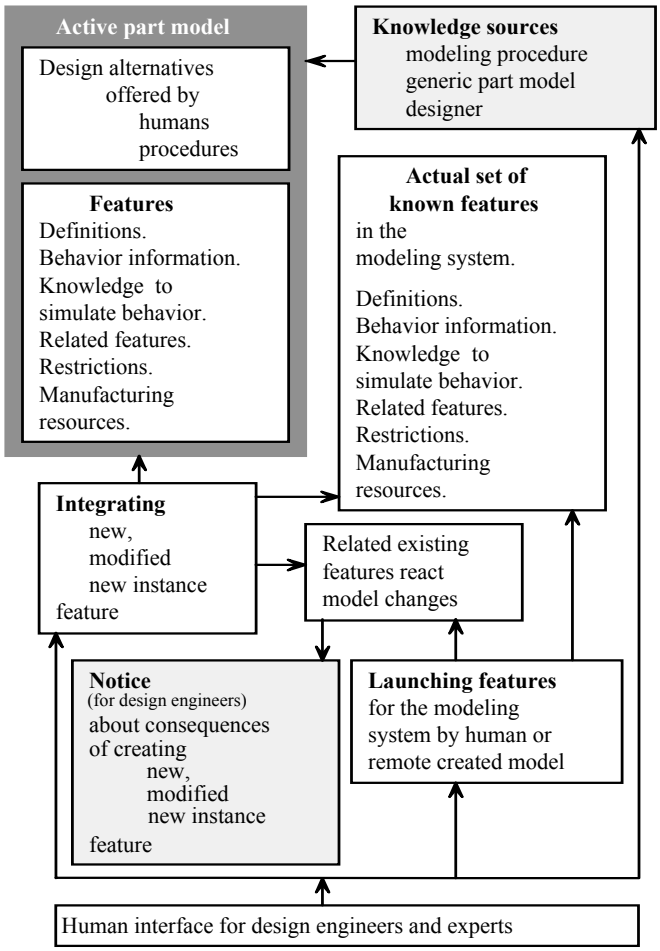


Figure 6
Active model

Features are in possession of information and knowledge necessary to simulate behavior of the modeled objects. At including a new or modified feature, some environment related information as previously defined related features, restriction definitions for prospective features and production resources are defined in order to integrate the new feature in the existing part model. Receiving this information, the related existing features react to the feature related model development activity. Consequently, features are aimed to create information about the effect of model changes and to communicate this with the related features. The above outlined approach also offers a real solution for reconstruction of exchanged models in remote receiving CAD/CAM systems. Definitions and behavior information for features also can be placed by engineers and experts in their hosts

then they can be accessed through Internet. Also, advice taking can be made available by remotely residence engineers in this way.

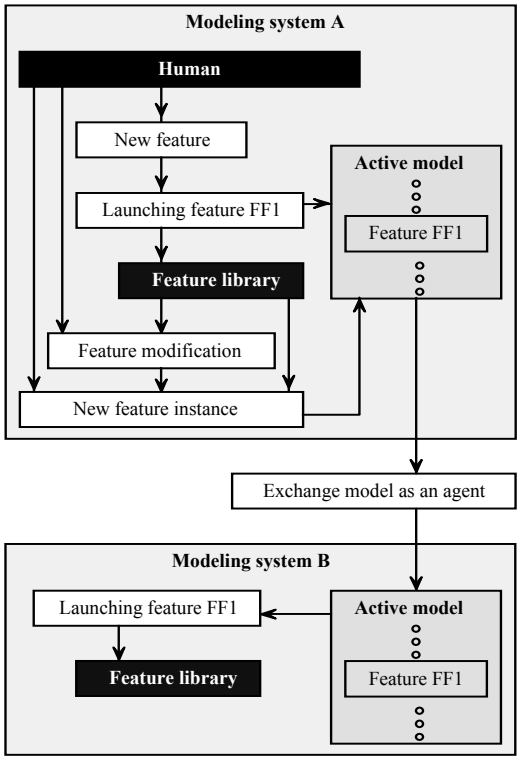


Figure 7
Feature in active model

Feature definition in case of agent based active model is outlined in Fig. 7. Feature definitions are stored in feature library in the modeling system *A*. During creation of the active model human defines a new feature *FF1*. Other possibilities are definition of modifications or instances of new features Besides feature instance specifications, knowledge to launch the feature *FF1* by the active model in the modeling system *B* is also included in the model.

5 Design Intent Supported Intelligent Modeling

Procedures for human-computer interaction (HCI) are adapted and enhanced for effective communication based design intent modeling. Modeling of design intent is in close connection with knowledge acquisition, data access and authority control, functions for human interactions, behavior of humans, human-human

communications and human errors (Fig. 8). The authors restricted their research to development of design intent handling methodology and processes including effects of human related issues on modeling of human intent.

Design intent often appears in some form of knowledge, but its source is not a single knowledge base. The related knowledge is domain but also product, designer and customer related. Consequently, it can not be involved in the modeling system in one of the conventional ways. Members of a group of engineers who are working on a product design can describe or point to different knowledge sources. Sometimes a knowledge is of personal nature in other cases access to a knowledge source is allowed for several engineers in a work group.

Quality of a decision depends on the performance of the human decision-maker. This performance can be increased effectively by the using of computer based decision assistance. At the beginning of her or his career an engineer is not well trained. Description of intents of skilled engineers can support beginners at their decision making. For that reason computer can learn decisions of skilled and experienced engineers in order to support decisions of less skilled engineers. Threshold knowledge can be defined and described and then used for the purpose of excluding untrained or careless engineers from decision making and avoiding fatal errors. This approach is called as mutual adaptive human-computer interfacing [9].

Design intent often is based on observations. Design or planning tasks are solved during cognitive processes where results of decisions are found out by designers. In this case it is very difficult to describe the human thinking process. Sometimes variants for types, parameter value ranges and discrete parameter values can be defined in intent descriptions. The authors considered modeling intent for model variants among the single decided variant. This makes it possible to change the decision to a more appropriate one during the application of the original product model. Understanding a given situation can be enhanced by appropriate intent descriptions from other engineers. This is the case of human-human communication using intent description.

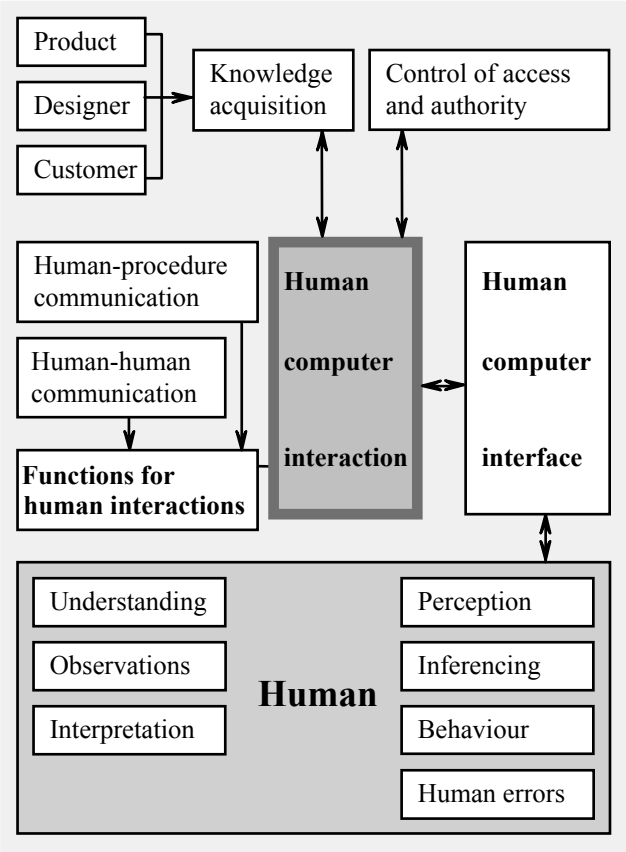


Figure 8
Human related issues

Intent description sometimes is a history description of the background of a decision by using of a chain of intent entities. Fig. 9 shows a typical chain of design intent entities and several referred entities from the outside world. Intent is mapped to a result of a decision in the product model as an entity, a parameter or a relationship of parameters. List of referred entities acts in the intent description as information source needed at processing of the intent description. Referred entities can be accessed using links included in the intent description. It should be emphasized that Fig. 4 shows only one of the possible styles of history. A history can be a simple identification of the decision-maker such as the management of a company or an authority without any explanation. History is considered as a chain of explanations for stages of a decision. Generic product models can involve generic intent models.

In the case of our example (Fig. 9) firstly a goal is defined for the decision then the related taxonomy is revealed. This is followed by a consideration of the

applied procedure as a thinking process of the engineer. In other words she or he did not apply any idea from handbooks, etc. Next the applied method is selected taking into consideration of the choice that is offered by the selected procedure. Alternative procedures and methods can be involved or referred if necessary. The procedure needs input data that were defined using production rules, functions and experimental results. The origin of the experimental results is an important element of the intent description.

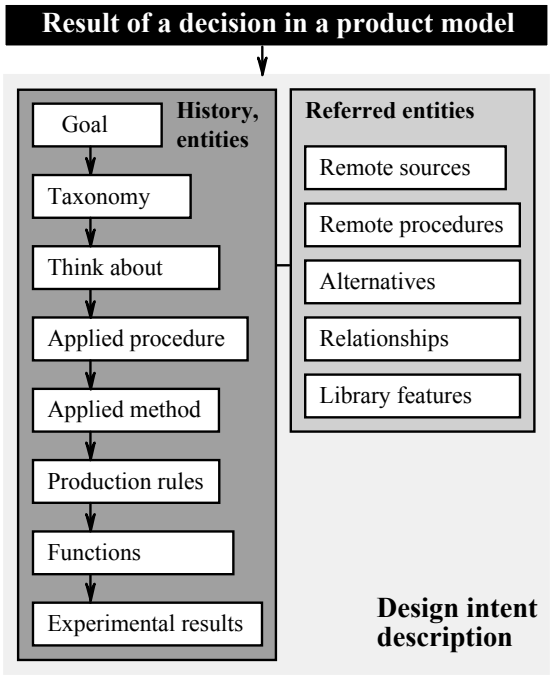


Figure 9
Structure of a design intent

Several typical considerations at modeling of design intent in the practice of modeling of mechanical parts are illustrated and discussed below by an example. Complex surfaces are often created using complex curves that govern their shape. Fig. 10 shows a section of a complex surface that consists of six component surfaces (*A*-*F*). Structural requirements were considered then surface types and dimensions were decided. Relationships of dimensions were defined. Then values of some dimensions were calculated. Fixed values of dimensions were defined as constraints. Other dimensions are allowed to be modified within well-defined ranges. These ranges were not described in the part model. Instead, one of the possible values was defined as a constraint. To assist later modification of this decision and to prevent later changes of the dimensions to move into illegal range, the allowed ranges of these dimensions were recorded as design intent entities. Component surfaces *A* and *F* should be flat and component surface *E* should be

cylindrical. All component surfaces are described using rational B-spline functions that offer free modification of their shape. Consequently, type of *A*, *E* and *F* surfaces should be constrained to avoid modification of their shapes as free form surfaces. There is a relationship between dimensions *A4* and *R1*. Elements of the intent description for decisions at the above discussed example are:

- *Decision on relationship* A_4-R_1. This decision is based on a method by an expert. Several alternative solutions were defined. These are involved in the intent model.
- *Shape of component surfaces B and D.* Modification of these free form surfaces is allowed. Although dimensions $A_{1\ min}$ and *A6* limit the modification of these surface components.
- *Shape of flat component surface C.* This surface can be modified as a free form surface if mating surfaces of other parts are modified accordingly. Cost consequences can be estimated by a procedure that is related to the intent description and can be accessed through Internet.
- *Fillet surfaces* connecting component surfaces can be modified with keeping the continuity between the connected surfaces and them.

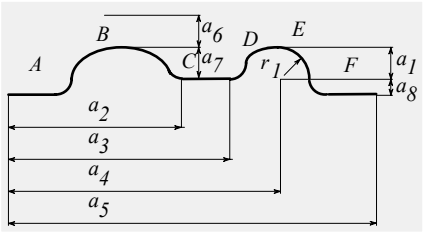


Figure 10
Design intent at the creation of a section for creating a complex surface

Is an intent count for much? Similar questions affect application of intent information. The answer is often subjective. Consequently, decisions can be assisted by the description of origin, limits, strength and consequence of omitting of intent. On the other side, sometimes engineer is not able to produce a good explain for a decision. This is because intuition and anticipation are still important factors at decisions of skilled and experienced engineers. Finally, an intent description acts on behalf of the engineer who created it.

Conclusions

The reported research is aimed to develop an enhanced, knowledge-based version of feature, associativity and constraint driven modeling of parts. Paper gives an outline of the proposed process-centric model based development of products using digital definition techniques in the form of active model. The proposed model includes model representations that describe behavior of modeled objects in different circumstances. Some possibilities and means of new enhancements in

active modeling procedures that work on the basis of knowledge based methods are investigated. Automatic, reactive feature based propagation of any change of model at any stage of the modeling process makes design consistent with intents, goals and decisions of the related engineers. Feature models involve specification and knowledge representations necessary to simulate behavior of the modeled objects. Active models act as agents after exchange them with other modeling systems at applications of models.

Active models are proposed to describe human originated knowledge on the basis of design intent modeling. Advanced product modeling handles model entities for description of results of human decisions, but not for description of background of those decisions. Frequent modification of models is done by engineers other than the original decision-makers. Intent of the original designer is required to do this without any loss of quality of the model. The authors proposed a method for modeling the background of decisions as design intents. They proposed a method for mapping design intent model descriptions to product model entities. Intent modeling describes human thinking process and considerations behind human decisions. Intent description can be used by both model application processes and humans who handle them. Implementation of the method assumes object oriented open surface modeling systems. Referred intent entities from the outside world are applied to link intent information to the intent model that are not economical or not allowed to involve in the intent model. The proposed method can be considered as an extension of collaborative engineering. Outstanding importance of its application is where the original decision-maker is not accessible and the only assistance at modification or evaluation of an earlier decision is application of design intent description. Intent modeling is not modeling of what should be done by an engineer but what did an engineer and what are consequences of future work of other engineers. It can be considered as a well-prepared substitution of the engineer by computer procedures.

Acknowledgments

The authors gratefully acknowledge the grant provided by the OTKA Fund for Research of the Hungarian Government. Project number is T 037304.

References

[1] L. Horváth and I. J. Rudas: "Virtual technology based associative integration of modeling of mechanical parts", *Journal of Advanced Computational Intelligence*, vol. 5, No. 5, pp. 269-278, 2001.

[2] L. Horváth, I. J. Rudas and C. Couto, "Integrationration of Human Intent Model Descriptions in Product Models", in book *Digital Enterprise - New Challenges Life-Cycle Approach in Management and Production*,: Kluwer Academic Publishers, 2001, pp: 1-12.

[3] L. Horvath, I. J. Rudas: "Integrated Associative Modeling of Parts and their Machining Process by Features*", in Proceedings of the ICMT 2001 conference*, Singapore, pp. 316-321.

[4] Laszlo Horvath, Imre J. Rudas, "Evaluation of Petri Net Process Model Representation as a Tool of Virtual Manufacturing", *in Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics, Information, Intelligence and Systems*, IEEE, San Diego,California, USA, 1998, Volume 1, pp. 178-183.

[5] M. Tambe, W. L. Johnson, R. Jones, F. Koss, J. Laird, P. Rosenbloom, and K. Schwamb, Intelligent agents for interactive simulation environments, AI Magazine, v. 16, n. 1, 1995.

[6] Les Piegl - Wayne Tiller: The Nurbs Book, Springer-Verlag, 1997

[7] Freitag, S., Opitz, K.,: Enhancements in blending algorithms, Hewlett-Packard Journal, October 1995, pp. 24-34

[8] Mannistö, T., Peltonen, H., Martio, A. Sulonen, R.: Modeling generic product structures in STEP, Computer-Aided Design, Vol. 30, No. 14, 1998, pp. 1111-1118

[9] Yoshikawa, H. - Takahashi, M.: Conceptual Design of Mutual Adaptive Interface. Preprint of Integrated Systems Engineering Conference, Baden-Baden, 1994, pp. 221-226.

Left-continuous t-norms in Fuzzy Logic: an Overview*

János Fodor

Dept. of Biomathematics and Informatics, Faculty of Veterinary Sci.
Szent István University, István u. 2, H-1078 Budapest, Hungary
E-mail: Fodor.Janos@aotk.szie.hu

Abstract: In this paper we summarize some fundamental results on left-continuous t-norms. First we study the nilpotent minimum and related operations in considerable details. This is the very first example of a left-continuous but not continuous t-norm in the literature. Then we recall some recent extensions and construction methods.

Keywords: Associative operations, Triangular norm, Residual implication, Left-continuous t-norm, Nilpotent minimum.

1 Introduction

The concept of Fuzzy Logic (FL) was invented by Lotfi Zadeh [20] and presented as a way of processing data by allowing partial set membership rather than only full or non-membership. This approach to set theory was not applied to engineering problems until the 70's due to insufficient small-computer capability prior to that time.

In the context of control problems (the most successful application area of FL), fuzzy logic is a problem-solving methodology that provides a simple way to arrive at a definite conclusion based upon vague, ambiguous, imprecise, noisy, or missing input information. FL incorporates a simple, rule-based "IF X AND Y THEN Z" approach to solving a control problem rather than attempting to model a system mathematically.

When one considers fuzzy subsets of a universe, in order to generalize the Boolean set-theoretical operations like intersection, union and complement, it is quite natural to use *interpretations* of logic connectives $\wedge$, $\vee$ and $\neg$, respectively [12]. It is assumed that the conjunction $\wedge$ is interpreted by a *triangular norm* (*t-norm* for short), the disjunction $\vee$ is interpreted by a *triangular conorm* (shortly: *t-conorm*), and the negation $\neg$ by a *strong negation*.

Although engineers have learned the basics of theoretical aspects of fuzzy sets and logic, from time to time it is necessary to summarize recent developments even in such a fundamental subject. This is the main aim of the present paper.

* This research has been supported in part by OTKA T046762.

Therefore, we focus on recent advances on an important and rather complex subclass of t-norms: on *left-continuous t-norms*. The standard example of a left-continuous t-norm is the *nilpotent minimum* [4,15]. Starting from our more than ten years old algebraic ideas, their elegant geometric interpretations make it possible to understand more on *left-continuous t-norms with strong induced negations*, and construct a wide family of them. Studies on properties of fuzzy logics based on left-continuous t-norms, and especially on the nilpotent minimum (NM) have started only recently; see [1,14,13,18,19] along this line.

2 Preliminaries

In this section we briefly recall some definitions and results will be used later. For more details see [5,12].

A bijection φ of the unit interval onto itself preserving natural ordering is called an *automorphism* of the unit interval. It is a continuous strictly increasing function satisfying boundary conditions $\varphi(0) = 0, \varphi(1) = 1$.

A *strong negation* N is defined as a strictly decreasing, continuous function $N\colon [0,1] \to [0,1]$ with boundary conditions $N(0) = 1,\ N(1) = 0$ such that N is involutive (i.e., $N(N(x)) = x$ holds for any $x \in [0,1]$). A standard example of a strong negation is given by $N_{\mathrm{st}}(x) = 1 - x$. Any strong negation N can be represented as a φ *-transform* of the standard negation (see [17])

$$N(x) = \varphi^{-1}(1 - \varphi(x))$$

for some automorphism φ of the unit interval. In this case the strong negation is denoted by N_φ.

A *t-norm* T is defined as a symmetric, associative and nondecreasing function $T\colon [0,1]^2 \to [0,1]$ satisfying boundary condition $T(1,x) = x$ for all $x \in [0,1]$.

A *t-conorm* S is defined as a symmetric, associative and nondecreasing function $S\colon [0,1]^2 \to [0,1]$ satisfying boundary condition $S(0,x) = x$ for all $x \in [0,1]$.

For any given t-norm T and strong negation N a function S defined by $S(x,y) = N(T(N(x), N(y)))$ is a t-conorm, called the *N-dual t-conorm of T*. In this case the triplet (T,S,N) is called a *De Morgan triplet*.

Well-accepted models for conjunction (AND), disjunction (OR), negation (NOT) are given by t-norms, t-conorms, strong negations, respectively. In this paper we will focus mainly on t-norms.

The definition of t-norms does not imply any kind of continuity. Nevertheless, such a property is desirable from theoretical as well as practical points of view.

A t-norm T is *continuous* if for all convergent sequences $\{x_n\}_{n\in\mathbb{N}}$, $\{y_n\}_{n\in\mathbb{N}}$ we have

$$T\left(\lim_{n\to\infty} x_n, \lim_{n\to\infty} y_n\right) = \lim_{n\to\infty} T(x_n, y_n).$$

The structure of continuous t-norms is well known, see [12] for more details, especially Section 3.3 on *ordinal sums*.

3 Left-continuous t-norms

In many cases, weaker forms of continuity are sufficient to consider. For t-norms, this property is *lower semicontinuity* [12, Section 1.3]. Since a t-norm T is non-decreasing and commutative, it is lower semicontinuous if and only if it is *left-continuous* in its first component. That is, if and only if for each $y \in [0,1]$ and for all non-decreasing sequences $\{x_n\}_{n\in\mathbb{N}}$ we have

$$\lim_{n\to\infty} T(x_n, y) = T\left(\lim_{n\to\infty} x_n, y\right).$$

If T is a left-continuous t-norm, the operation $I_T : [0,1]^2 \to [0,1]$ defined by

$$I_T(x,y) = \sup\{t \in [0,1] \mid T(x,t) \le y\} \tag{1}$$

is called the *residual implication* (shortly: R-implication) generated by T. An equivalent formulation of left-continuity of T is given by the following property ($x, y, z \in [0,1]$):

(R) $\quad T(x,y) \le z \quad$ if and only if $\quad I_T(x,z) \ge y.$

We emphasize that the formula (1) can be computed for any t-norm T; however, the resulting operation I_T satisfies condition (R) if and only if the t-norm T is left-continuous. An interesting underlying algebraic structure of left-continuous t-norms is a commutative, residuated integral l-monoid, see [6] for more details.

4 Nilpotent Minimum and Maximum

The first known example of a left-continuous but non-continuous t-norm is the so-called *nilpotent minimum* [4] denoted as $T^{\mathbf{nM}}$ and defined by

$$T^{\mathbf{nM}}(x,y) = \begin{cases} 0 & \text{if } x+y \le 1, \\ \min(x,y) & \text{otherwise.} \end{cases} \tag{2}$$

It can be understood as follows. We start from a t-norm (the minimum), and re-define its value below and along the diagonal $\{(x,y) \in [0,1] \mid x+y=1\}$. So, the question is natural: if we consider any t-norm T and "annihilate" its original values below and along the mentioned diagonal, is the new operation always a t-norm? The general answer is "no" (although the contrary was "proved" in [15] where the same operation also appeared).

The definition (2) can be extended as follows. Suppose that φ is an automorphism of the unit interval. Define a binary operation on $[0,1]$ by

$$T_\varphi^{\mathbf{nM}}(x,y) = \begin{cases} 0 & \text{if } \varphi(x)+\varphi(y) \le 1 \\ \min(x,y) & \text{if } \varphi(x)+\varphi(y) > 1 \end{cases}. \tag{3}$$

Thus defined $T_\varphi^{\mathbf{nM}}$ is a t-norm and is called the *φ-nilpotent minimum*.

Clearly, the following equivalent form of $T_\varphi^{\mathbf{nM}}$ can be obtained by using the strong negation N_φ generated by φ:

$$T_\varphi^{\mathbf{nM}}(x,y) = \begin{cases} \min(x,y) & \text{if } y > N_\varphi(x) \\ 0 & \text{otherwise} \end{cases}.$$

Extension of $T_\varphi^{\mathbf{nM}}$ for more than two arguments is easily obtained and is given by $T_\varphi^{\mathbf{nM}}(x_1,\ldots,x_n) = \min_{i=1,n}\{x_i\}$ if $\min_{i\neq j}\{\varphi(x_i)+\varphi(x_j)\} > 1$, and $T_\varphi^{\mathbf{nM}}(x_1,\ldots,x_n) = 0$ otherwise.

The N_φ-dual t-conorm of $T_\varphi^{\mathbf{nM}}$, called the *φ-nilpotent maximum*, is defined by

$$S_\varphi^{\mathbf{nM}}(x,y) = \begin{cases} \max(x,y) & \text{if } \varphi(x)+\varphi(y) < 1 \\ 1 & \text{otherwise} \end{cases}.$$

Clearly, $(T_\varphi^{\mathbf{nM}}, S_\varphi^{\mathbf{nM}}, N_\varphi)$ yields a De Morgan triple.

In the next theorem we list the most important properties of $T_\varphi^{\mathbf{nM}}$ and $S_\varphi^{\mathbf{nM}}$. These are easy to prove.

Theorem 1. *Suppose that φ is an automorphism of the unit interval. The t-norm $T_\varphi^{\mathbf{nM}}$ and the t-conorm $S_\varphi^{\mathbf{nM}}$ have the following properties:*
(a) The law of contradiction holds for $T_\varphi^{\mathbf{nM}}$ as follows:

$$T_\varphi^{\mathbf{nM}}(x, N_\varphi(x)) = 0 \quad \forall x \in [0,1].$$

(b) The law of excluded middle holds for $S_\varphi^{\mathbf{nM}}$:

$$S_\varphi^{\mathbf{nM}}(x, N_\varphi(x)) = 1 \quad \forall x \in [0,1].$$

(c) There exists a number α_0 depending on φ such that $0 < \alpha_0 < 1$ and $T_\varphi^{\mathbf{nM}}$ is idempotent on the interval $]\alpha_0, 1]$:

$$T_\varphi^{\mathbf{nM}}(x,x) = x \quad \forall x \in]\alpha_0, 1].$$

(d) With the previously obtained α_0, $S_\varphi^{\mathbf{nM}}$ is idempotent on the interval $[0, \alpha_0[$:

$$S_\varphi^{\mathbf{nM}}(x,x) = x \quad \forall x \in [0, \alpha_0[.$$

(e) There exists a subset X_φ of the unit square such that $(x,y) \in X_\varphi$ if and only if $(y,x) \in X_\varphi$ and the law of absorption holds on X_φ as follows:

$$S_\varphi^{\mathbf{nM}}(x, T_\varphi^{\mathbf{nM}}(x,y)) = x \quad \forall (x,y) \in X_\varphi.$$

(f) There exists a subset Y_φ of the unit square such that $(x,y) \in Y_\varphi$ if and only if $(y,x) \in Y_\varphi$ and the law of absorption holds on Y_φ as follows:

$$T_\varphi^{\mathbf{nM}}(x, S_\varphi^{\mathbf{nM}}(x,y)) = x \quad \forall (x,y) \in Y_\varphi.$$

(g) If A, B are fuzzy subsets of the universe of discourse U and the α-cuts are denoted by A_α, B_α, respectively ($\alpha \in [0,1]$), then we have

$$A_\alpha \cap B_\alpha = [T_\varphi^{\mathbf{nM}}(A,B)]_\alpha \quad \forall \alpha \in]\alpha_0, 1]$$

and

$$A_\alpha \cup B_\alpha = [S_\varphi^{\mathbf{nM}}(A,B)]_\alpha \quad \forall \alpha \in [0, \alpha_0[,$$

where α_0 is given in (c).
(h) $T_\varphi^{\mathbf{nM}}$ is a left-continuous t-norm and $S_\varphi^{\mathbf{nM}}$ is a right-continuous t-conorm. □

4.1 Where does Nilpotent Minimum Come from?

Nilpotent minimum has been discovered not by chance. There is a study on contrapositive symmetry of fuzzy implications [4]. A particular case of those investigations yielded nilpotent minimum. Some of the related results will be cited later in the present paper.

Let T be a left-continuous t-norm and N a strong negation. Consider the residual implication I_T generated by T, defined in (1).

Contrapositive symmetry of I_T with respect to N (CPS(N) for short) is a property that can be expressed by the following equality:

$$I_T(x,y) = I_T(N(y), N(x)) \quad \forall x, y \in [0,1]. \tag{4}$$

Unfortunately, (4) is generally not satisfied for I_T generated by a left-continuous (even continuous) t-norm T. In [4] we proved the following result.

Theorem 2 ([4])**.** *Suppose that T is a t-norm such that condition **(R)** is satisfied, N is a strong negation. Then the following conditions are equivalent ($x, y, z \in [0,1]$).*

(a) I_T has contrapositive symmetry with respect to N;
(b) $I_T(x,y) = N(T(x, N(y)))$;
(c) $T(x,y) \le z$ if and only if $T(x, N(z)) \le N(y)$.

In any of these cases we have

(d) $N(x) = I_T(x,0)$,
(e) $T(x,y) = 0$ if and only if $x \le N(y)$. □

In the case of continuous t-norms we have the following unicity result (see also [11]).

Theorem 3. *Suppose that T is a continuous t-norm. Then I_T has contrapositive symmetry with respect to a strong negation N if and only if there exists an automorphism φ of the unit interval such that*

$$T(x,y) = \varphi^{-1}(\max\{\varphi(x) + \varphi(y) - 1, 0\}), \tag{5}$$
$$N(x) = \varphi^{-1}(1 - \varphi(x)). \tag{6}$$

In this case I_T is given by

$$I_T(x,y) = \varphi^{-1}(\min\{1 - \varphi(x) + \varphi(y), 1\}). \quad \square \tag{7}$$

When I_T is any R-implication and I_T does not have contrapositive symmetry then we can associate another implication with I_T. Suppose that T is a t-norm which satisfies condition **(R)**. Define a new implication associated with I_T as follows:

$$x \to_T y = \max\{I_T(x,y), I_T(N(y), N(x))\}. \tag{8}$$

If I_T has contrapositive symmetry then $x \to_T y = I_T(x,y) = I_T(N(y), N(x))$.

Define also a binary operation $*_T$ by

$$x *_T y = \min\{T(x,y), N[I_T(y, N(x))]\}. \tag{9}$$

Obviously, $*_T = T$ if (4) is satisfied by $I = I_T$. Even in the opposite case, this operation $*_T$ is a fuzzy conjunction in a broad sense and has several nice properties as we state in the next theorem.

Theorem 4. *Suppose that T is a t-norm such that **(R)** is true, N is a strong negation such that $N(x) \geq I_T(x,0)$ for all $x \in [0,1]$ and operations $\to_T$ and $*_T$ are defined by (8) and (9), respectively. Then the following conditions are satisfied:*

(a) $1 *_T y = y$;
(b) $x *_T 1 = x$;
(c) $*_T$ *is nondecreasing in both arguments;*
(d) $x \to_T y \geq z$ *if and only if* $x *_T z \leq y$. $\square$

In Table 1 we list most common t-norms and corresponding operations I_T, $*_T$, $\to_T$, with $N(x) = 1 - x$.

Therefore, nilpotent minimum can be obtained as the conjunction $*_{\min}$. In general, $*_T$ is not a t-norm, not even commutative. Sufficient condition to assure that $*_T$ is a t-norm is given in the next theorem.

Theorem 5. *For a t-norm T and a strong negation N, if $y > N(x)$ implies $T(x,y) \leq N(I_T(y, N(x)))$ then $*_T$ is also a t-norm.* $\square$

T	$\min(x,y)$	$\max(x+y-1,0)$	xy
I_T	$1,\ x \le y$ y otherwise	$\min(1-x+y,1)$	$\min\ \ 1, \frac{y}{x}$
$*_T$	$\min(x,y),\ x+y>1$ $0,\qquad x+y\le 1$	$\max(x+y-1,0)$	$\min\ \ xy, \frac{x+y-1}{y}$
$\to_T$	$1,\qquad x \le y$ $\max(1-x,y),\ x>y$	$\min(1-x+y,1)$	$\max\ \ \frac{y}{x}, \frac{1-x}{1-y}$

Table 1. Some t-norms and associated connectives

4.2 Implications Defined by Nilpotent Minimum and Maximum

Consider the De Morgan triple $(T_\varphi^{\mathbf{nM}}, S_\varphi^{\mathbf{nM}}, N_\varphi)$ with an automorphism φ of the unit interval and define the corresponding S-implication:

$$I(x,y) = S_\varphi^{\mathbf{nM}}(N_\varphi(x), y) \tag{10}$$

$$= \begin{cases} 1, & x \le y \\ \max(N_\varphi(x), y), & x > y \end{cases}. \tag{11}$$

One can easily prove that the R-implication defined by $T_\varphi^{\mathbf{nM}}$ coincides with the S-implication in (11).

Proposition 1. *Let φ be any automorphism of the unit interval. Then we have for all $x, y \in [0,1]$ that*

$$I_{T_\varphi^{\mathbf{nM}}}(x,y) = S_\varphi^{\mathbf{nM}}(N_\varphi(x), y). \quad \square$$

As a trivial consequence, $I_{T_\varphi^{\mathbf{nM}}}$ always has contrapositive symmetry with respect to N_φ.

Now we list the most important and attractive properties of $I_{T_\varphi^{\mathbf{nM}}}$. Their richness is due to the fact that R- and S-implications coincide and thus advantageous features of both classes are combined.

1. $I_{T_\varphi^{\mathbf{nM}}}(x,.)$ is non-decreasing
2. $I_{T_\varphi^{\mathbf{nM}}}(.,y)$ is non-increasing
3. $I_{T_\varphi^{\mathbf{nM}}}(1,y) = y$

4. $I_{T_\varphi^{\mathbf{nM}}}(0, y) = 1$

5. $I_{T_\varphi^{\mathbf{nM}}}(x, 1) = 1$

6. $I_{T_\varphi^{\mathbf{nM}}}(x, y) = 1$ if and only if $x \leq y$

7. $I_{T_\varphi^{\mathbf{nM}}}(x, y) = I_{T_\varphi^{\mathbf{nM}}}(N_\varphi(y), N_\varphi(x))$

8. $I_{T_\varphi^{\mathbf{nM}}}(x, 0) = N_\varphi(x)$

9. $I_{T_\varphi^{\mathbf{nM}}}(x, I_{T_\varphi^{\mathbf{nM}}}(y, x)) = 1$

10. $I_{T_\varphi^{\mathbf{nM}}}(x, .)$ is right-continuous

11. $I_{T_\varphi^{\mathbf{nM}}}(x, x) = 1$

12. $I_{T_\varphi^{\mathbf{nM}}}(x, I_{T_\varphi^{\mathbf{nM}}}(y, z)) = I_{T_\varphi^{\mathbf{nM}}}(y, I_{T_\varphi^{\mathbf{nM}}}(x, z)) = I_{T_\varphi^{\mathbf{nM}}}(T_\varphi^{\mathbf{nM}}(x, y), z)$

13. $T_\varphi^{\mathbf{nM}}(x, I_{T_\varphi^{\mathbf{nM}}}(x, y)) \leq \min(x, y)$

14. $I_{T_\varphi^{\mathbf{nM}}}(x, y) \geq \min(x, y)$

Notice that $I_{T_\varphi^{\mathbf{nM}}}$ can also be viewed as a QL-implication defined by

$$\begin{aligned} S(x, y) &= S_\varphi^{\mathbf{nM}}(x, y), \\ N(x) &= N_\varphi(x) \\ T(x, y) &= \min(x, y) \end{aligned}$$

in (4), as one can check easily by simple calculus.

Therefore, this QL-implication (which is, in fact, an S-implication and an R-implication at the same time) also has contrapositive symmetry with respect to N_φ. Concerning this case, the following unicity result was proved in [4].

Theorem 6 ([4]). *Consider a QL-implication defined by* $\max_\varphi(N_\varphi(x), T(x, y))$, *where T is a t-norm. This implication has contrapositive symmetry with respect to N_φ if and only if* $T = \min$. □

5 Extensions and Constructions

In this section we summarize some important results on left-continuous t-norms obtained by Jenei and other researchers.

5.1 Left-continuous t-norms with Strong Induced Negations

The notions and some of the results in the above Theorem 2 were formulated in a slightly more general framework in [7]. We restrict ourselves to the case of left-continuous t-norms with strong induced negations; i.e., T is a left-continuous t-norm and the function $N_T(x) = I_T(x, 0)$ (the negation induced by T) is a strong negation.

Moreover, in a sense, a converse statement of Theorem 2 was also established in [7]: If T is a left-continuous t-norm such that $N_T(x) = I_T(x, 0)$ is a strong negation, then (a), (b) and (c) necessarily hold with $N = N_T$.

Already in [3], we studied the above algebraic property (c). Geometric interpretations of properties (b) and (c) were given in [7] under the names of *rotation invariance* and *self-quasi inverse property*, respectively. More exactly, we have the following definition.

Definition 1. Let $T : [0, 1]^2 \to [0, 1]$ be a symmetric and non-decreasing function, and let N be a strong negation. We say that T admits the *rotation invariance* property with respect to N if for all $x, y, z \in [0, 1]$ we have

$$T(x, y) \leq z \quad \text{if and only if} \quad T(y, N(z)) \leq N(x).$$

In addition, suppose T is left-continuous. We say that T admits the *self quasi-inverse* property w.r.t. N if for all $x, y, z \in [0, 1]$ we have

$$I_T(x, y) = z \quad \text{if and only if} \quad T(x, N(y)) = N(z). \quad \square$$

For left-continuous t-norms, rotation invariance is exactly property (c) in Theorem 2, while self quasi-inverse property is just a slightly reformulated version of (b) there. Nevertheless, the following geometric interpretation was given in [7]. If N is a the standard negation and we consider the transformation $\sigma : [0, 1]^3 \to [0, 1]^3$ defined by $\sigma(x, y, z) = (y, N(z), N(x))$, then it can be understood as a rotation of the unit cube with angle of $2\pi/3$ around the line connecting the points $(0, 0, 1)$ and $(1, 1, 0)$. Thus, the formula $T(x, y) \leq z \iff T(y, N(z)) \leq N(x)$ expresses that the part of the unit cube above the graph of T remains invariant under σ. This is illustrated in the first part of Figure 1.

The second part of Figure 1 is about the self quasi-inverse property which can be described as follows (for quasi-inverses of decreasing functions see [16]). For a left-continuous t-norm T, we define a function $f_x : [0, 1] \to [0, 1]$ as follows: $f_x(y) = N_T(T(x, y))$. It was proved in [7] that f_x is its own quasi-inverse if and only if T admits the self quasi-inverse property. Assume that N is the standard negation. Then the geometric interpretation of the negation is the reflection of the graph with respect to the line $y = 1/2$. Then, if it is applied to the partial mapping $T(x, \cdot)$, extend discontinuities of $T(x, \cdot)$ with vertical line segments. Then the obtained graph is invariant under the reflection with respect to the diagonal $\{(x, y) \in [0, 1] \mid x + y = 1\}$ of the unit square.

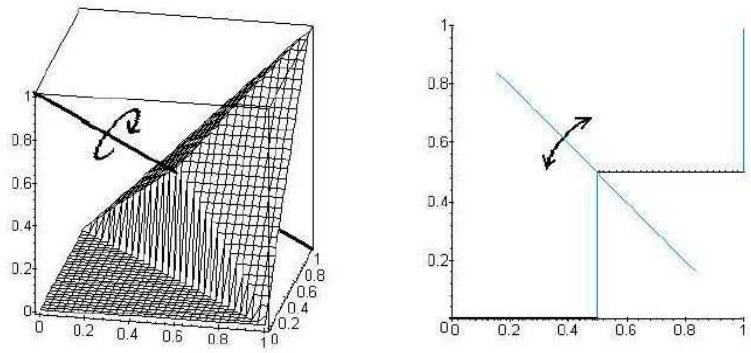


Fig. 1. Rotation invariance property (left). Self quasi-inverse property (right).

5.2 Rotation Construction

Theorem 7 ([9]). *Let N be a strong negation, t its unique fixed point and T be a left-continuous t-norm without zero divisors. Let T_1 be the linear transformation of T into $[t,1]^2$. Let $I^+ =]t,1]$, $I^- = [0,t]$, and define a function $T_{\mathbf{rot}} : [0,1]^2 \to [0,1]$ by*

$$T_{\mathbf{rot}}(x,y) = \begin{cases} T(x,y) & \text{if } x,y \in I^+, \\ N(I_{T_1}(x,N(y))) & \text{if } x \in I^+ \text{ and } y \in I^-, \\ N(I_{T_1}(y,N(x))) & \text{if } x \in I^- \text{ and } y \in I^+, \\ 0 & \text{if } x,y \in I^-. \end{cases}$$

Then $T_{\mathbf{rot}}$ is a left-continuous t-norm, and its induced negation is N.

When we start from the standard negation, the construction works as follows: take any left-continuous t-norm without zero divisors, scale it down to the square $[1/2,1]^2$, and finally rotate it with angle of $2\pi/3$ in both directions around the line connecting the points $(0,0,1)$ and $(1,1,0)$. This is illustrated in Fig. 2.

Remark that there is another recent construction method of left-continuous t-norms (called rotation-annihilation) developed in [10].

5.3 Annihilation

Let N be a strong negation (i.e., an involutive order reversing bijection of the closed unit interval). Let T be a t-norm. Define a binary operation $T_{(N)} : [0,1]^2 \to [0,1]$ as follows:

$$T_{(N)}(x,y) = \begin{cases} T(x,y) & \text{if } x > N(y) \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

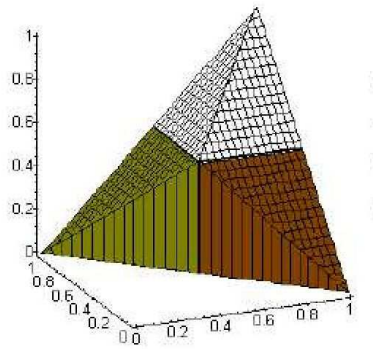


Fig. 2. $T^{\mathbf{nM}}$ as the rotation of the min, with the standard negation

We say that T *can be N-annihilated* when $T_{(N)}$ is also a t-norm. So, the question is: which t-norms can be N-annihilated? The above results show that $T = \min$ is a positive example.

A t-norm T is said to be a *trivial annihilation* (with respect to the strong negation N) if $N(x) = I_T(x, 0)$ holds for all $x \in [0, 1]$. It is easily seen that if a continuous t-norm T is a trivial annihilation then $T_{(N)} = T$.

Two t-norms T, T' are called *N-similar* if $T_{(N)} = T'_{(N)}$. Let T be a continuous non-Archimedean t-norm, and $\langle [a, b]; T_1 \rangle$ be a summand of T. We say that this summand is *in the center* (w.r.t. the strong negation N) if $a = N(b)$.

Theorem 8 ([8])**.** *(a) Let T be a continuous Archimedean t-norm. Then $T_{(N)}$ is a t-norm if and only if $T(x, N(x)) = 0$ holds for all $x \in [0, 1]$.*

(b) Let T be a continuous non-Archimedean t-norm. Then $T_{(N)}$ is a t-norm if and only if

- *either T is N-similar to the minimum,*
- *or T is N-similar to a continuous t-norm which is defined by one trivial annihilation summand in the center.* □

Interestingly enough, the nilpotent minimum can be obtained as the limit of trivially annihilated continuous Archimedean t-norms, as the following result states.

Theorem 9 ([8])**.** *There exists a sequence of continuous Archimedean t-norms T_k $(k = 1, 2, \ldots)$ such that*

$$\lim_{k \to \infty} T_k(x, y) = T^{\mathbf{nM}}(x, y) \qquad (x, y \in [0, 1]).$$

Moreover, for all k, T_k is a trivial annihilation with respect to the standard negation.

The nilpotent minimum was slightly extended in [2] by allowing a weak negation instead of a strong one in the construction. Based on this extension, monoidal t-norm based logics (MTL) were studied also in [2], together with the involutive case (IMTL). Ordinal fuzzy logic, closely related to $T^{\mathbf{nM}}$, and its application to preference modelling was considered in [1]. Properties and applications of the $T^{\mathbf{nM}}$-based implication (called R_0 implication there) were published in [14]. Linked to [2], the equivalence of IMTL logic and NM logic (i.e., nilpotent minimum based logic) was established in [13].

6 Conclusion

In this paper we have presented an overview of some fundamental results on left-continuous t-norms. The origin and basic properties of the very first left-continuous (and not continuous) t-norm called *nilpotent minimum* was recalled in some details. Extensions and general construction methods for left-continuous t-norms were also reviewed from the literature.

Acknowledgement

The author is grateful to S. Jenei for placing Figures 1–2 at his disposal.

References

1. B. De Baets, F. Esteva, J. Fodor and Ll. Godo, Systems ofordinal fuzzy logic with application to preference modelling, *Fuzzy Sets and Systems* **124** (2001) 353-359.
2. F. Esteva and L. Godo, Monoidal t-norm based logic: towards a logic for left-continuous t-norms, *Fuzzy Sets and Systems* **124** (2001) 271–288.
3. J.C. Fodor, A new look at fuzzy connectives, *Fuzzy Sets and Systems* **57** (1993) 141–148.
4. J.C. Fodor, Contrapositive symmetry of fuzzy implications, *Fuzzy Sets and Systems* **69** (1995) 141–156.
5. J.C. Fodor and M. Roubens, *Fuzzy Preference Modelling and Multicriteria Decision Support*, (Kluwer Academic Publishers, Dordrecht, 1994).
6. U. Höhle, Commutative, residuated l-monoids, in: U. Höhle and E.P. Klement, Eds., *Non-Classical Logics and their Applications to Fuzzy Subsets. A Handbook of the Mathematical Foundations of Fuzzy Set Theory.* (Kluwer Academic Publishers, Boston, 1995).
7. S. Jenei, Geometry of left-continuous t-norms with strong induced negations, *Belg. J. Oper. Res. Statist. Comput. Sci.* **38** (1998) 5–16.
8. S. Jenei, New family of triangular norms via contrapositive symmetrization of residuated implications, *Fuzzy Sets and Systems* **110** (2000) 157–174.
9. S. Jenei, Structure of left-continuous t-norms with strong induced negations. (I) Rotation construction, *J. Appl. Non-Classical Logics* **10** (2000) 83–92.

10. S. Jenei, Structure of left-continuous triangular norms with strong induced negations. (II) Rotation-annihilation construction, *J. Appl. Non-Classical Logics*, **11** (2001) 351–366.
11. P. Smets and P. Magrez, Implications in fuzzy logic, *Int. J. Approximate Reasoning* **1** (1987) 327–347.
12. E.P. Klement, R. Mesiar, and E. Pap, *Triangular Norms*, (Kluwer Academic Publishers, Dordrecht, 2000).
13. D. Pei, On equivalent forms of fuzzy logic systems NM and IMTL, *Fuzzy Sets and Systems* **138** (2003) 187-195.
14. D. Pei. R_0 implication: characteristics and applications, *Fuzzy Sets and Systems* **131** (2002) 297 - 302.
15. P. Perny, *Modélisation, agrégation et exploitation des préférences floues dans une problématique de rangement*, (PhD thesis, Université Paris-Dauphine, Paris, 1992).
16. B. Schweizer and A. Sklar, *Probabilistic Metric Spaces*, (North-Holland, New York, 1983).
17. E. Trillas, Sobre funciones de negación en la teoría de conjuntos difusos, *Stochastica* **III** (1979) 47–60.
18. S.-M. Wang, B.-S. Wang and F. Ren, NMŁ, a schematic extension of F.Esteva and L.Godos logic MTL, *Fuzzy Sets and Systems* (to appear)
19. S.-M. Wang, B.-S. Wang and X.-Y. Wang, A characterization of truth-functions in the nilpotent minimum logic, *Fuzzy Sets and Systems* **145** (2004) 253-266.
20. L.A. Zadeh, Fuzzy sets, *Information and Control* **8** (1965) 338–353.

Approximate Reasoning in Fuzzy Systems Based on Pseudo-analysis and Uninorm Residuum

Márta Takács

Department of Information Technology, Budapest Tech
Népszínház utca 8, H-1081 Budapest, Hungary
takacs.marta@nik.bmf.hu

Abstract: The paper introduces novel residuum-based reasoning systems in a pseudo—analysis based uninorm environment. Based on the definitions and theorems for lattice ordered monoids and left continuous uninorms and t-norms, certain distance-based operators are focused on, with the help of which the uninorm-residuum based approximate reasoning system becomes possible in Fuzzy Logic Control (FLC) systems, but as it will be shown, this type of the reasoning partially satisfies the conditions for approximate reasoning and inference mechanism for FLC systems.

Keywords:FLC, approximate reasoning, uninorms

1 Introduction

The modelling of the uncertain processes in our modern society is a very complex problem. Since the systems are multi-criterial and multipart, decision processes become increasingly vague and hard to analyse. The human brain possesses some special characteristics that enable it to learn and reason in a vague and fuzzy environment. Naturally, there have been models that have investigated the behaviour of complex systems before the introduction of the fuzzy systems, but fuzzy systems have proven to be to a greater or lesser degree more useful than the classical models.

The real time functioning of the dynamic engineering system is a necessary condition. The earlier, differential-equation based models, with the expansion of the complexity of the real systems are growing out of all proportion, and it is very hard to construct a real time decision making model under those circumstances. Since the fuzzy logic control systems are based on linguistic variables [14], and fuzzy approximate reasoning, and therefore they are successful and effective [15].

Generally, the fundamental of the decision making in fuzzy based real systems is the approximate reasoning, which is a rule-based system. Knowledge representation in a rule-based system is done by means of IF…THEN rules. Furthermore, approximate reasoning systems allow fuzzy inputs, fuzzy antecedents, fuzzy consequents. "Informally, by approximate or, equivalently, fuzzy reasoning, we mean the process or processes by which a possibly imprecise conclusion is deduced from a collection of imprecise premises. Such reasoning is, for the most part, qualitative rather than quantitative in nature and almost all of it falls outside of the domain of applicability of classical logic", [13]. This fuzzy representation allows a closer match with many of the important concepts of practical affairs, which lack the sharp boundaries assumed by classical logic.

The fundamental point arising from the classical two-valued logic is that it imposes a dichotomy on any mathematical model. However, in many cases of daily life, a borderline between the two possibilities is not evident. There are, of course the generalizations of two-valued logic, the multi-valued logics. In this model there may be a finite or infinite number of truth values, that is, an infinite number of degrees to which a property may be present. But in contrast with multi-valued logics, "fuzzy logic differs from conventional logical systems in that it aims at providing a model for approximate rather than precise reasoning", [1].

Fuzzy logic also provides a system that is sufficiently flexible and expressive to serve as a natural framework for the semantics of natural languages. Also in fuzzy logic truth itself is allowed to be such as "quite true", "more or less true",[2]. Although application-oriented fuzzy systems seek to be simple and comprehensible, it is obvious that they are heavily related to the fields of classical multi-valued logic, operation research and functional analysis. There are numerous models that are yet to gain exact mathematical description, but have already proven their applicability in practice.

This latter question seems to be answered. Experts are confident in using special types of operations in fuzzy systems, such as t-norms, t-conorms, uninorms, and more generally, aggregation operators, and researchers are more and more meticulous in providing exact mathematical definitions for those. The focus is on certain properties of operators (continuity and representability, for instance), and those classes of operators are highlighted that correspond to the applications. This paper is a step towards the investigation of this problem by reviewing a specific case, where the investigated structure is a real semi-ring with pseudo-operations [16].

The question raised previously was whether there are general operation groups which satisfy the residuum-based approximate reasoning, but at the same time are easily comprehensible and acceptable to application-oriented experts. How come the residuum-based approximate reasoning is not as wide-spread in mathematical logic as the Mamdani-type? Why is the application of operations in fuzzy systems limited to that of the minimum and the product? In [11] only a partial answer to

this question is laid out. The axiom system presented in [11] greatly contributes to this. Situated between the theses, these axioms declare the expectations of the approximate reasoning systems, and it becomes clear to what degree they satisfy or violate this system.

The theoretical basis used in [11] provided by [7], which is currently the leading work in the world concerning fuzzy operators. It represents the monography describing t-norms, t-conorms and related operators for fuzzy sets and numbers. The other basic background of this research are the distance-based operators introduced by Rudas in his work [8],[9]. The characteristics of those operators were further investigated by Rudas and the author of this paper in various joint projects [10]. The research heavily relies on approximate reasoning and fuzzy logic theory (focusing on implications). The following works were mainly used in this investigation: [2], [4].

Concerning the structure of the work is the next: the first section an overview of pseudo-analysis based uninorm operators is given, with its role in residuum-based approximate reasoning. It is emphasized which operators from distance based operator group are the ones corresponding to the uninorm-based residuum known so far. The next section also introduces novel reasoning systems in a uninorm environment. Based on the theorems shown in the first section, certain distance-based operators are focused on, with the help of which the residuum-based approximate reasoning system becomes possible.

2 Mathematical Background of the Novel Approximate Reasoning Method

2.1 Real Semiring and the Pseudo-operators

The base for the pseudo-analysis is a real semiring, defined in the following way:

Let $[a,b]$ be a closed subinterval of $[-\infty,+\infty]$ (in some cases semi-closed subintervals will be considered) and let $\preceq$ be a total order on $[a,b]$. A *semiring* is the structure $(\preceq,\oplus,\otimes)$ if the following hold:

- $\oplus$ is pseudo-addition, i.e., a function $\oplus:[a,b]\times[a,b]\to[a,b]$ which is commutative, non-decreasing (with respect to $\preceq$), associative and with a zero element denoted by $\mathbf{0}$;
- $\otimes$ is pseudo-multiplication, i.e., a function $\otimes:[a,b]\times[a,b]\to[a,b]$ which is commutative, positively non-decreasing ($x\preceq y$ implies

$x \otimes z \preceq x \otimes y$ where $z \in [a,b]_+ = \{z | z \in [a,b], \mathbf{0} \preceq z\}$ associative and for which there exists a unit element denoted by **1**.

- $\mathbf{0} \otimes z = \mathbf{0}$
- $x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$

Three basic classes of semirings with continuous (up to some points) pseudo-operations are:

(i) The pseudo-addition is an idempotent operation and the pseudo-multiplication is not.

(ii) Semi-rings with strict pseudo-operations defined by a monotone and continuous generator function $g : [a,b] \to [0,+\infty]$, i.e., *g*-semirings $x \oplus y = g^{-1}(g(x) + g(y))$ and $x \otimes y = g^{-1}(g(x)g(y))$.

(iii) Both operations, $\oplus$ and $\otimes$, are idempotent.

More on this structure can be found in [16], [17].

In this paper we will consider the interval $[a,b]$ to be the unit interval $[0,1]$, in which case pseudo-operations will be t-conorm, t-norm and uninorm.

2.2 Uninorms

Both the neutral element 1 of a t-norm and the neutral element 0 of a t-conorm are boundary points of the unit interval. However, there are many important operations whose neutral element is an interior point of the underlying set. The fact that the first three axioms (commutativity, associativity, monotonicity) coincide for t-norms and for t-conorms, i.e., the only axiomatic difference lies in the location of the neutral element, has led to the introduction of a new class of binary operations closely related to t-norms and t-conorms.

A *uninorm* is a binary operation U on the unit interval, i.e., a function $U : [0,1]^2 \to [0,1]$ which satisfies the following properties for all $x, y, z \in [0,1]$

$U(x,y) = U(y,x)$, i.e. the uninorm is commutative,

$U(U(x,y),z) = U(x,U(y,z))$, i.e. the uninorm is associative,

$x \le y \Rightarrow U(x,z) \le U(y,z)$, i.e. the uninorm monotone,

$U(e,x) = x$, i.e., a neutral element exists, which is $e \in [0,1]$.

2.2.1 Lattice Ordered Monoids and Left Continuous Uninorms and T-norms

Let L be a non-empty set. Lattice is a partially (totally) ordered set which for any two elements $x, y \in L$ also contains their *join* $x \vee y$ (i.e., the least upper bound of the set $\{x, y\}$), and their *meet* $x \wedge y$ (i.e., the greatest lower bound of the set $\{x, y\}$), denoted by $(L, \preceq)$. Secondly, $(L, *)$ is a semi-group with the neutral element. Following [13], [14] let the following be introduced:

Definition 2.1.

Let $(L, \preceq)$ be a lattice and $(L, *)$ a semi-group with the neutral element.

(i) The triple $(L, *, \preceq)$ is called a *lattice-ordered monoid* (or an *l-monoid*) if for all $x,y,z \in L$ we have

(*LM1*) $x * (y \vee z) = (x * y) \vee (x * z)$ and

(*LM2*) $(x \vee y) * z = (x * z) \vee (y * z)$.

(ii) An $(L, *, \preceq)$ *l*- monoid is said to be *commutative* if the semi-group $(L, *)$ is commutative.

(iii) A commutative $(L, *, \preceq)$ *l*- monoid is said to be *commutative, residuated l-monoid* if there exists a further binary operation $\rightarrow_*$ on L, i.e., a function $\rightarrow_*: L^2 \rightarrow L$ (*the* $*$ *residuum*), such that for all $x,y,z \in L$ we have

(*Res*) $x * y \preceq z$ if and only if $x \preceq (y \rightarrow_* z)$.

(iv) An *l*-monoid $(L, *, \preceq)$ is called an *integral* if there is a greatest element in the lattice $(L, \preceq)$ (often called the universal upper bound) which coincides with the neutral element of the semi-group $(L, *)$.

Obviously, each *l*-monoid $(L, *, \preceq)$ is a partially ordered semi-group, and in the case of commutativity the axioms (*LM1*) and (*LM2*) are equivalent.

In the following investigations the focus will be on the lattice $([0,1], \leq)$, we will usually work with a complete lattice, i.e., for each subset A of L its join $\bigvee A$ and its $\bigwedge A$ exist and are contained in L. In this case, L always has a greatest element, also called the *universal upper bound.*

Example 2.1. If we define $* : [0,1]^2 \rightarrow [0,1]$ by

$$x * y = \begin{cases} min(x,y) & if \quad x + y \le 1 \\ max(x,y) & otherwise \end{cases}$$

then $([0,1],*,\le)$ is a commutative, residuated *l*-monoid, and the *-residuum is given by

$$x \to_* y = \begin{cases} max(1-x,y) & if \quad x \le y \\ min(1-x,y) & otherwise \end{cases}.$$

It is not an integral, since the neutral element is 0.5.

The operation * results in an *uninorm*, and special types of distance based operators (see [9], [2] and section 4. prom the paper).

The following result is on important characterization of left-continuous uninorms.

Theorem 2.1.

For each function $U : [0,1]^2 \to [0,1]$ the following are equivalent:

(*i*) $([0,1],U,\le)$ is a commutative, residuated *l*-monoid, with a neutral element

(*ii*) U is a left continuous uninorm.

In this case the U-residuum $\to_U$ is given by

(*ResU*) $$x \to_U y = sup\{z \in [0,1] | U(x,z) \le y\}.$$

Proof. (In the [9])

The work of De Baets, B. and Fodor, J. [2] presents general theoretical results related to residual implicators of uninorms, based on residual implicators of t-norms and t-conorms.

Residual operator R_U, considering the uninorm U , can be represented in the following form:

$$R_U(x,y) = \sup\{z | z \in [0,1] \wedge U(x,z) \le y\}.$$

Uninorms with the neutral elements $e = 0$ and $e = 1$ are t-norms and t-conorms, respectively, and related residual operators are widely discussed, we also find suitable definitions for uninorms with neutral elements $e \in]0,1[$.

If we consider a uninorm U with the neutral element $e \in]0,1[$, then the binary operator R_U is an implicator if and only if $(\forall z \in]e,1[)(U(0,z) = 0)$. Furthermore R_U is an implicator if U is a disjunctive right-continuous idempotent uninorm with unary operator g satisfying $(\forall z \in [0,1])(g(z) = 0 \Leftrightarrow z = 1)$.

The residual implicator R_U of uninorm U can be denoted by Imp_U.

Consider a uninorm U, then R_U is an implicator in the following cases:

(i) U is a conjunctive uninorm,

(ii) U is a disjunctive representable uninorm,

(iii) U is a disjunctive right-continuous idempotent uninorm with unary operator g satisfying

$(\forall z \in [0,1])(g(z) = 0 \Leftrightarrow z = 1)$.

3 Approximate Reasoning and the Fuzzy Logic Control

In control theory and also in theory of the approximate reasoning introduced by Zadeh in 1979, [11] much of the knowledge of system behaviour and system control can be stated in the form of if-then rules. The Fuzzy Logic Control, FLC has been carried out searching for different mathematical models in order to supply these rules.

In most sources it was suggested to represent an

if x is A then x is B

rule in the form of fuzzy implication (shortly $Imp(A,B)$, relation (shortly $R(A,B)$), or simply as a connection (for example as a t-norm, $T(A,B)$) between the so called rule premise: x is A and rule consequence: y is B. Let x be from universe X, y from universe Y, and let x and y be linguistic variables. Fuzzy set A in X is characterised by its membership function μ_A: $x \rightarrow [0,1]$. The most significant differences between the models of FLC-s lie in the definition of this connection, relation or implication.

The other important part of the FLC is the inference mechanism. One of the widely used methods is the Generalised Modus Ponens (GMP), in which the main point is, that the inference y is B' is obtained when the propositions are:

- the i^{th} rule from the rule system of n rules: if x is A_i then y is B_i
- and the system input x is A'.

GMP sees the real influences of the implication or connection choice on the inference mechanisms in fuzzy systems ([3],[10]). Usually the general rule consequence for i^{th} rule from a rule system is obtained by

$$B_i'(y) = \sup_{x \in X} T(A'(x), Imp(A_i(x), B_i(y))).$$

3.1 The Axioms of Inference Mechanism

Let *RB* a fuzzy rule system of *n* rules, with rule premises *x* is A_i and rule consequences: *y* is B_i ($i = 1,2,...n$). Let *x* be from universe *X*, *y* from universe *Y*, and let *x* and *y* be linguistic variables. Fuzzy set A_i on $X \subset \Re$ finite universe is characterized by its membership function μ_{Ai}: $x \to [0,1]$, and fuzzy set *B* on $Y \subset \Re$ universe is characterized by its membership function μ_{Bi}: $y \to [0,1]$. Let *x* is *A'* be the system input, where *A'* is characterized by its membership function $\mu_{A'}$: $x \to [0,1]$.

Applying the generalized compositional rule of inference to given components, the *i-th* rule output, with respect to the given *RB* and given system input *A',* is *y* is B_i' given by the expression

$$B_i'(y) = \sup_{x \in X} T(A'(x), \mathrm{Imp}(A_i(x), B_i(y))),$$

where, on a general level, Imp is the relationship between rule base premise and rule base consequence, satisfying the following conditions:

(out1) If the input coincides with one of the premises, then the resulting output coincides with the corresponding consequence, i.e.,

$$(\exists i \in \{1,2,...n\})(A' = A_i) \text{ then } B_i' = B_i .$$

(out2) For each normal input *A'* the output is not contained in all consequences, i.e.,

$$(\exists i \in \{1,2,...n\})(B_i' < B_i) .$$

(out3) The rule output belongs to the convex hull of B_i, ($i \in I$), where

$$I = \{i \mid 1 \le i \le n, Supp(A') \cap Supp(A_i) \ne 0\}.$$

In [9] we can find an axiom system on the same principle.

4 Residuum-based Approximate Reasoning with Distance-based Uninorms

In fact the uninorms offer new possibilities in fuzzy approximate reasoning, because the low level of covering over of rule premise and rule input has measurable influence on rule output as well. In some applications the meaning of that novel t-norms, has practical importance. The modified Mamdani's approach, with similarity measures between rule premises and rule input, does not rely on

the compositional rule inference any more, but still satisfies the basic conditions supposed for the approximate reasoning for a fuzzy rule base system [15].

Having results from [2], we can introduce residuum-based inference mechanism using distance-based uninorms.

4.1 Modified Distance-based Operators

The distance-based operators can be expressed by means of the min and max operators as follows (the only modification on distance based operators described in [9] is the boundary condition for neutral element *e*):

the *maximum distance minimum operator with respect* to $e \in [0,1]$ is defined as

$$max_e^{min} = \begin{cases} max(x,y), & \text{if } y > 2e - x \\ min(x,y), & \text{if } y < 2e - x \\ min(x,y), & \text{if } y = 2e\text{-}x \end{cases},$$

the *minimum distance minimum operator* with respect to $e \in [0,1]$ is defined as

$$min_e^{min} = \begin{cases} min(x,y), & \text{if } y > 2e - x \\ max(x,y), & \text{if } y < 2e - x \\ min(x,y), & \text{if } y = 2e\text{-}x \end{cases}.$$

The distance-based operators have the following properties

max_e^{min} *and* max_e^{max} are uninorms,

the dual operator of the uninorm max_e^{min} *is* max_{1-e}^{max} *, and*

the dual operator of the uninorm max_e^{max} *is* max_{1-e}^{min}.

Based on results from [2] and [3] we conclude:

Operator $max_{0.5}^{min}$ is a conjunctive left-continuous idempotent uninorm with neutral element $e \in [0,1]$ with the super-involutive decreasing unary operator $g(x) = 2e - x = 2 \cdot 0.5 - x \Rightarrow g(x) = 1 - x$.

Operator $min_{0.5}^{max}$ is a disjunctive right-continuous idempotent uninorm with neutral element $e \in]0,1]$ with the sub-involutive decreasing unary operator $g(x) = 2e - x = 2 \cdot 0.5 - x \Rightarrow g(x) = 1 - x$.

4.2 Idempotent Uninorms and the Residual Implicators of Uninorms

A binary operator V is called idempotent, if $V(x,x) = x, (\forall x \in X)$. It is well known, that the only idempotent t-norm is *min*, and the only t-conorm is *max*.

In [2], by De Baets, B. and Fodor, J., has studied two important classes of uninorms: the class of left-continuous and the class of right-continuous ones.

If we suppose a unary operator *g* on set [0,1], then *g* is called super-involutive if $g(g(x)) \geq x$ for $(\forall x \in [0,1])$.

A binary operator U is a conjunctive left-continuous idempotent uninorm with neutral element $e \in]0,1]$ if and only if there exist a super-involutive decreasing unary operator *g* with fixpoint e and $g(0) = 1$ such that U for any $\forall (x,y) \in [0,1]^2$ is given by

$$U(x,y) = \begin{cases} min(x,y) & if \quad y \leq g(x) \\ max(x,y) & elsewhere \end{cases}.$$

Residual operator R_U, considering the uninorm U , can be represented in the following form:

$$R_U(x,y) = sup\{z | z \in [0,1] \wedge U(x,z) \leq y\}.$$

Uninorms with neutral elements $e = 0$ and $e = 1$ are t-norms and t-conorms, respectively, and related residual operators are widely discussed.

If we consider a uninorm U with neutral element $e \in]0,1[$, then the binary operator R_U is an implicator if and only if $(\forall z \in]e,1[)(U(0,z) = 0)$. Furthermore R_U is an implicator if U is a disjunctive right-continuous idempotent uninorm with unary operator *g* satisfying $(\forall z \in [0,1])(g(z) = 0 \Leftrightarrow z = 1)$.

The residual implicator R_U of uninorm U can be denoted by Imp_U.

4.3 Residual Implicators of Distance Based Operators

According to Theorem 8. in [2] we introduce implicator of distance based operator $max_{0.5}^{min}$.

Consider the conjunctive left-continuous idempotent uninorm $max_{0.5}^{min}$ with the unary operator $g(x) = 1 - x$, then its residual implicator $Imp_{max_{0.5}^{min}}$ is given by

$$Imp_{max_{0.5}^{min}} = \begin{cases} max(1-x, y) & if \quad x \le y \\ min(1-x, y) & elsewhere \end{cases} \qquad (1)$$

4.4 Residuum-based Approximate Reasoning with Distance Based Operator

Although the minimum plays an exceptional role in fuzzy control theory, there are situations requiring new models. In system control one would intuitively expect to make the powerful coincidence between fuzzy sets stronger, and the weak coincidence even weaker. The distance-based operators group satisfy these properties. Let we consider a mathematical approach: residuum-based approximate reasoning and inference mechanism. Hence, and because of the results from sections of this paper we can n consider the general rule consequence for *i*-th rule from a rule system as

$$B_i'(y) = \sup_{x \in X} \left(max_{0.5}^{min} \left(A'(x), Imp_{max_{0.5}^{min}} (A_i(x), B_i(y)) \right) \right)$$

or, using formula (*3.1.*)

$$B_i'(y) = \sup_{x \in X} \begin{cases} \max_{0.5}^{\min} (A'(x), \max(1 - A_i(x), B_i(y))) & if \ A_i(x) \le B_i(y) \\ \max_{0.5}^{\min} (A'(x), \min(1 - A_i(x), B_i(y))) & elsewhere \end{cases} \qquad (2)$$

The rule base output is constructed as a crisp value calculated with a defuzzification model, from rule base output. Rule base output is an aggregation of all rule consequences $B_i'(y)$, from the rule base. As aggregation operator, in this case, dual operator $max_{0.5}^{max}$ of $max_{0.5}^{min}$ can be used.

$$B'_{out}(y) = \max_{0.5}^{\max} (B_n'(y), \max_{0.5}^{\max} (B_{n-1}'(y), \max_{0.5}^{\max} (...., \max_{0.5}^{\max} (B_2'(y), B_1'(y))))).$$

4.5 Uninorm-residuum Based Approximate Reasoning and the Axioms of Inference Mechanism

Axiom (out1): Taken into account Proposition 13. from [2], it can be conclude, that conjunctive left-continuous idempotent uninorm $max_{0.5}^{min}$ and its implicator $Imp_{max_{0.5}^{min}}$ satisfy the inequality

$$B'_i(y) = max_{0.5}^{min}\left(A'(x), Im\, p_{max_{0.5}^{min}}(A_i(x), B_i(y))\right) \le B_i(y)$$

for *i*-th rule in rule base system, if $A'(x) = A_i(x)$ for all $x \in X$. It means, that this

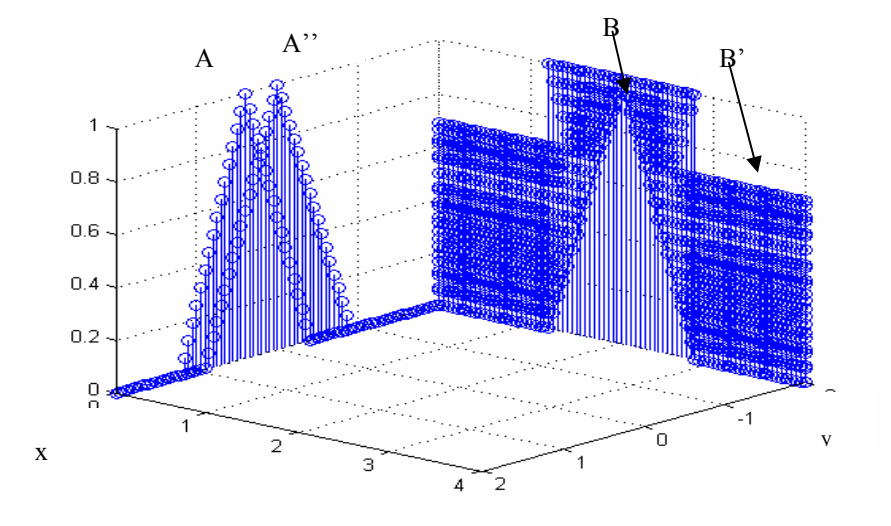


Figure 1

Output *B'* calculating with uninorm-residuum based approximate reasoning

type of reasoning partially satisfies the conditions for approximate reasoning, hence $B'_i(y) = B_i(y)$ or $B'_i(y) < B_i(y)$ if $A'(x) = A_i(x)$ for all $x \in X$.

Axiom (out2)). In most of cases uninorm-residuum based approximate reasoning violates this axioms of inference mechanism, because for normal input *A'* the output is contained in all consequences, if we have not "faired" rule.

Axiom (out3): If $A \neq A'$, the rule output belongs not to the convex hull of B_i, $(i=1,n)$.

Figure 1. shows the situation, where $A \neq A'$, calculating rule output B' with the expression (2).

In [8] it was proved in simulations, that in this case if all the rules, where the rule doesn't has real influence on output, have been real time eliminated, the results are acceptable.

Conclusions

Based on the definitions and theorems for lattice ordered monoids and left continuous uninorms and t-norms, certain distance-based operators are focused on, with the help of which the uninorm-residuum based approximate reasoning system becomes possible in Fuzzy Logic Control (FLC) systems. This type of reasoning partially satisfies the conditions for approximate reasoning and inference mechanism for FLC systems.

References

[1] Bellmann, R.E., Zadeh., L.A., (1977), *Local and fuzzy logic,* Modern Uses of Multiple-Valued Logic, edited by Dunn, J.M., Epstein, G., Reidel Publ., Dordrecht, The Netherlands, pp. 103-165.

[2] De Baets, B., Fodor, J., (1999), *Residual operators of uninorms*,, Soft Computing 3., (1999), pp. 89-100.

[3] Fodor, J., Rubens, M., (1994), *Fuzzy Preference Modeling and Multi-criteria Decision Support*. Kluwer Academic Pub.,1994.

[4] G. Hancke, Á. Szeghegyi: „*Nonlinear Control Via TP Model Transformation: The TORA System Example*", In the Proc.Of the Symposium on Applied Machine Intelligence (SAMI 2004), Herlany, Slovakia, Januar 16-17 2004, pp.333 - 340.

[5] Klement, E.P., Mesiar, R, Pap, E., '*Triangular Norms*', Kluwer Academic Publishers, 2000, ISBN 0-7923-6416-3

[6] Rudas, I., *Absorbing-Norms,* in Proceedings of the 3th International Symposium of Hungarian Researchers Computational Intelligence, Buadpest, Nov. 2002., pp. 25-43.

[7] Rudas, Imre; Kaynak, O., New Types of Generalized Operations, Computational Intelligence, Soft Computing and Fuzzy-Neuro Integration with Applications, Springer NATO ASI Series. Series F, Computer and Systems Sciences, Vol. 192. 1998. (O. Kaynak, L. A. Zadeh, B. Turksen,I. .J. Rudas editors), pp. 128-156.

[8] M. Takacs: Investigation on a Special Group of Fuzzy Implication Operators and Fuzzy Inference Mechanism Using a Simplified Rule Base

System, Proceedings of the 1997 International Conference on Neural Information Processing and Intelligent Information Systems, New Zealand, Dunedin, Univ. of Otago, 23-28 November 1997, Vol. 2, pp. 793-796.

[9] Takacs, M., Approximate Reasoning in Fuzzy Systems Based On Pseudo-Analysis, Phd Thesis, Univ. of Novi Sad, 2004.

[10] Zadeh, L. A., A Theory of approximate reasoning, In Hayes, J., and editors, Mashine Intelligence, Vol. 9, Halstead Press, New York, 1979., pp. 149-194.

[11] Zadeh, L. A., (1999), From Computing with Numbers to Computing with Words – From Manipulation of Measurements to manipulation of Perceptions, In Proc. Of EUROFUSE –SIC Conf. 1999., Budapest, pp. 1-3.

[12] Zimmermann, H.J., (1991), Fuzzy Sets, Decision Making and Expert Systems. Kluwer, Boston, 1991.

[13] Pap, E., Triangular norms in modelling uncertainly, non-linearity and decision, in Proceedings of the 2th International Symposium of Hungarian researchers Computational Intelligence, ISBN 963 7154 06 X, pp. 7-18.

[14] Birkhoff, G., (1973), Lattice theory. American Mathematical Siciety, Providence

[15] Takacs, M., (2003c), Approximate reasoning with Distance-based Operators and degrees of coincidence, in Principles of Fuzzy Preference Modelling and Decision Making, edited by Bernard de Baets, János Fodor, Academia Press Gent, 2003., ISBN 90-382-0567-8

[16] Pap, E., Ralević, N., (1999), Pseudo-operations on finite intervals, Novi Sad J. Math. Vol. 29 (1) (1999), pp. 9-18.

[17] Pap, E., Štajner, I., (1999), Generalized pseudo-convolution in the theory of probabilistic metric spaces, information, fuzzy numbers, optimization, system theory, Fuzzy Sets and Systems 102., pp. 393-415.

TP model transformation based observer design to 2-D Aeroelastic System

Péter Baranyi — Yeung Yam

baranyi@alpha.ttt.bme.hu
yyam@acae.cuhk.edu.hk

Abstract: This paper presents a case study how to apply the recently proposed TP model transformation technique, that has been introduced for nonlinear state-feedback control design, to nonlinear observer design. The study is conducted through an example. This example treats the question of observer design to the prototypical aeroelastic wing section with structural nonlinearity. This type of model has been traditionally used for the theoretical as well as experimental analysis of two- dimensional aeroelastic behavior. The model investigated in the paper describes the nonlinear plunge and pitch motion of a wing, and exhibits complex nonlinear behavior. In preliminary works this prototypical aeroelastic wing section was stabilized by a state-feedback controller designed via TP model transformation and linear matrix inequalities. Numerical simulations are used to provide empirical validation of the resulting observer.

1 Introduction

The main goal of the paper is to study how to apply the TP (Tensor Product) model transformation to observer design. The motivation of this goal is that the TP model transformation was proposed under the Parallel Distributed Compensation (PDC) design framework [1] for nonlinear state feedback controller design [2, 3]. The TP model transformation is capable of transforming a given time varying (parameter dependent, where the parameters may include state variables) linear state-space model into time varying convex combination of finite number of linear time invariant models. Whether the given model is analytical model or just an outcome of black box identification (e.g. neural net or fuzzy approximation with Takagi-Sugeno, Mamdani or Rudas [4, 5] type inference operator) is irrelevant. The resulting linear time invariant models can then be readily substituted into Linear Matrix Inequalities (LMI), available under the PDC design framework, to determine a time varying (parameter dependent, where the parameters may include state variables) nonlinear controller according to given control specifications. The whole above design can be executed numerically by computers and hence the controller can be determined without analytical derivations in acceptable time. In most cases not all of the state variables are available, but only some of them. This paper studies how to apply the result of the TP model transformation to observer design under the PDC design framework similarly to the controller design. The resulting observer can then be applied to estimate the unavailable state variables.

The example of this paper is about the observer design to the prototypical aeroelastic wing section. A few papers were printed in last years dealing with the state-feedback control design of the prototypical aeroelastic wing section via TP model transformation, for instance see [6, 7, 8]. This paper focuses attention on the observer design to the prototypical aeroelastic wing section since not all of the state variables of the prototypical aeroelastic wing section are available in reality.

2 Nomenclature

This section is devoted to introduce the notations being used in this paper: $\{a,b,\ldots\}$: scalar values, $\{\mathbf{a},\mathbf{b},\ldots\}$: vectors, $\{\mathbf{A},\mathbf{B},\ldots\}$: matrices, $\{\mathcal{A},\mathcal{B},\ldots\}$: tensors.

$\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$: vector space of real valued $(I_1\times I_2\times\cdots\times I_N)$-tensors. Subscript defines lower order: for example, an element of matrix $\mathbf{A}$ at row-column number i,j is symbolized as $(\mathbf{A})_{i,j}=a_{i,j}$. Systematically, the i-th column vector of $\mathbf{A}$ is denoted as $\mathbf{a}_i$, i.e. $\mathbf{A}=\begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 & \cdots\end{bmatrix}$. $\diamond_{i,j,n},\ldots$: are indices. $\diamond_{I,J,N},\ldots$: index upper bound: for example: $i=1..I$, $j=1..J$, $n=1..N$ or $i_n=1..I_n$. $\mathbf{A}_{(n)}$: n-mode matrix of tensor $\mathcal{A}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$. $\mathcal{A}\times_n\mathbf{U}$: n-mode matrix-tensor product. $\mathcal{A}\otimes_n\mathbf{U}_n$: multiple product as $\mathcal{A}\times_1\mathbf{U}_1\times_2\mathbf{U}_2\times_3..\times_N\mathbf{U}_N$. Detailed discussion of tensor notations and operations is given in [9].

3 Basic concepts

The detailed description of the TP model transformation and PDC design framework is beyond the scope of this paper and can be found in [1, 2, 3, 6]. In the followings a few concepts are presented being used in this paper, for more details see [1, 2, 3, 6].

3.1 Parameter-varying state-space model

Consider parameter-varying state-space model:

$$\dot{\mathbf{x}}(t)=\mathbf{A}(\mathbf{p}(t))\mathbf{x}(t)+\mathbf{B}(\mathbf{p}(t))\mathbf{u}(t) \tag{1}$$

$$\mathbf{y}(t)=\mathbf{C}(\mathbf{p}(t))\mathbf{x}(t)+\mathbf{D}(\mathbf{p}(t))\mathbf{u}(t),$$

with input $\mathbf{u}(t)$, output $\mathbf{y}(t)$ and state vector $\mathbf{x}(t)$. The system matrix

$$\mathbf{S}(\mathbf{p}(t))=\begin{pmatrix}\mathbf{A}(\mathbf{p}(t)) & \mathbf{B}(\mathbf{p}(t))\\ \mathbf{C}(\mathbf{p}(t)) & \mathbf{D}(\mathbf{p}(t))\end{pmatrix}\in\mathbb{R}^{O\times I} \tag{2}$$

is a parameter-varying object, where $\mathbf{p}(t)\in\Omega$ is time varying $N-$dimensional parameter vector, where $\Omega=[a_1,b_1]\times[a_2,b_2]\times..\times[a_N,b_N]\subset\mathbb{R}^N$ is a closed hypercube. $\mathbf{p}(t)$ can also include some (or all) elements of $\mathbf{x}(t)$.

3.2 Convex state-space TP model

Equ. (2) can be approximated for any parameter $\mathbf{p}(t)$ as a convex combination of the R number of LTI system matrices $\mathbf{S}_r$, $r = 1..R$. Matrices $\mathbf{S}_r$ are also termed as vertex system matrices. Therefore, one can define weighting functions $w_r(\mathbf{p}(t)) \in [0,1] \subset \mathbb{R}$ such that matrix $\mathbf{S}(\mathbf{p}(t))$ belongs to the convex hull of $\mathbf{S}_r$ as $\mathbf{S}(\mathbf{p}(t)) = co\{\mathbf{S}_1, \mathbf{S}_2, .., \mathbf{S}_R\}_{\mathbf{w}(\mathbf{p}(t))}$, where vector $\mathbf{w}(\mathbf{p}(t))$ contains the weighting functions $w_r(\mathbf{p}(t))$ of the convex combination. The control design methodology, to be applied in this paper, uses univariate weighting functions. Thus, the explicit form of the convex combination in terms of tensor product becomes:

$$\begin{pmatrix} \dot{\mathbf{x}}(t) \\ \mathbf{y}(t) \end{pmatrix} \approx \tag{3}$$

$$\left(\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} .. \sum_{i_N=1}^{I_N} \prod_{n=1}^{N} w_{n,i_n}(p_n(t)) \mathbf{S}_{i_1,i_2,..,i_N} \right) \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix}.$$

(3) is termed as TP model in this paper. Function $w_{n,j}(p_n(t)) \in [0,1]$ is the j-th univariate weighting function defined on the n-th dimension of Ω, and $p_n(t)$ is the n-th element of vector $\mathbf{p}(t)$. I_n (n=1,...,N) is the number of univariate weighting functions used in the n-th dimension of the parameter vector $\mathbf{p}(t)$. The multiple index $(i_1, i_2, ..., i_N)$ refers to the LTI system corresponding to the i_n-th weighting function in the n-th dimension. Hence, the number of LTI vertex systems $\mathbf{S}_{i_1,i_2,..,i_N}$ is obviously $R = \prod_n I_n$. One can rewrite (3) in the concise TP form as:

$$\begin{pmatrix} s\mathbf{x}(t) \\ \mathbf{y}(t) \end{pmatrix} \approx \mathcal{S} \overset{N}{\underset{n=1}{\otimes}} \mathbf{w}_n(p_n(t)) \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix}, \tag{4}$$

that is

$$\mathbf{S}(\mathbf{p}(t)) \underset{\varepsilon}{\approx} \mathcal{S} \overset{N}{\underset{n=1}{\otimes}} \mathbf{w}_n(p_n(t)).$$

Here, ε represents the approximation error, and row vector $\mathbf{w}_n(p_n) \in \mathbb{R}^{I_n}$ contains the weighting functions $w_{n,i_n}(p_n)$, the $N+2$ -dimensional coefficient tensor $S \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N \times O \times I}$ is constructed from the LTI vertex system matrices $\mathbf{S}_{i_1,i_2,...,i_N} \in \mathbb{R}^{O \times I}$. The first N dimensions of S are assigned to the dimensions of Ω. The convex combination of the LTI vertex systems is ensured by the conditions:

Definition 1 *The TP model (4) is convex if:*

$$\forall n, i, p_n(t) : w_{n,i}(p_n(t)) \in [0,1]; \tag{5}$$

$$\forall n, p_n(t) : \sum_{i=1}^{I_n} w_{n,i}(p_n(t)) = 1. \tag{6}$$

This simply means that $\mathbf{S}(\mathbf{p}(t))$ is within the convex hull of LTI vertex systems $\mathbf{S}_{i_1,i_2,..,i_N}$ for any $\mathbf{p}(t) \in \Omega$.

Remark 1 $\mathbf{S}(\mathbf{p}(t))$ *has finite TP model representation in many cases* ($\varepsilon = 0$ *in (4)). However, one should face that exact finite element TP model representation does not exist in general* ($\varepsilon > 0$ *in (4)), see [10, 11]. In this case* $\varepsilon \mapsto 0$*, when the number of LTI systems involved in the TP model goes to* ∞*. In the present observer design, the state-space dynamic model of the prototypical aeroelastic wing section can be exactly represented by a finite convex TP model.*

4 Model of the prototypical aeroelastic wing section

In the past few years various studies of aeroelastic systems have emerged. [12] presents a detailed background and refers to a number of papers dealing with the modelling and control of aeroelastic systems. The following provides a brief summary of this background. [13] and [14] proposed non-linear feedback control methodologies for a class of non-linear structural effects of the wing section [15]. Papers [13, 16, 12] develop a controller, capable of ensuring local asymptotic stability, via partial feedback linearization. It has been shown that by applying two control surfaces global stabilization can be achieved. For instance, global feedback linearization technique were introduced for two control actuators in the work of [12]. TP model transformation based control design was introduced in [6, 7, 8]. This control design ensures asymptotic stability with one control surface and is capable of involving various control specification beyond stability.

4.1 Equations of Motion

In this paper, we consider the problem of flutter suppression for the prototypical aeroelastic wing section as shown in Figure 1. The aerofoil is constrained to have two degrees of freedom, the plunge h and pitch α. The equations of motion of the system have been derived in many references (for example, see [17], and [18]), and can be written as

$$\begin{pmatrix} m & mx_\alpha b \\ mx_\alpha b & I_alpha \end{pmatrix} \begin{pmatrix} \ddot{h} \\ \ddot{\alpha} \end{pmatrix} + \begin{pmatrix} c_h & 0 \\ 0 & c_\alpha \end{pmatrix} \begin{pmatrix} \dot{h} \\ \dot{\alpha} \end{pmatrix} + \tag{7}$$
$$+ \begin{pmatrix} k_h & 0 \\ 0 & k_\alpha(\alpha) \end{pmatrix} \begin{pmatrix} h \\ \alpha \end{pmatrix} = \begin{pmatrix} -L \\ M \end{pmatrix},$$

where

$$L = \rho U^2 b c_{l_\alpha} \left(\alpha + \frac{\dot{h}}{U} + \left(\frac{1}{2} - a \right) b \frac{\dot{\alpha}}{U} \right) + \rho U^2 b c_{l_\beta} \beta \tag{8}$$
$$M = \rho U^2 b^2 c_{m_\alpha} \left(\alpha + \frac{\dot{h}}{U} + \left(\frac{1}{2} - a \right) b \frac{\dot{\alpha}}{U} \right) + \rho U^2 b c_{m_\beta} \beta,$$

and where x_α is the non-dimensional distance between elastic axis and the centre of mass; m is the mass of the wing; I_α is the mass moment of inertia; b is semi-chord of the wing, and c_α and c_h respectively are the pitch and plunge structural

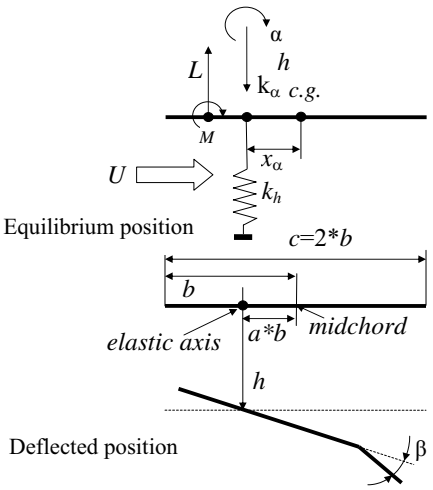


Figure 1: Aeroelastic model

damping coefficients, and k_h is the plunge structural spring constant. Traditionally, there have been many ways to represent the aerodynamic force L and moment M, including steady, quasi-steady, unsteady and non-linear aerodynamic models. In this paper we assume the quasi-steady aerodynamic force and moment, see work [17]. It is assumed that L and M are accurate for the class of low velocities concerned. Wind tunnel experiments are carried out in [14]. In the above equation ρ is the air density, U is the free stream velocity, c_{l_α} and c_{m_α} respectively, are lift and moment coefficients per angle of attack, and c_{l_β} and c_{m_β}, respectively are lift and moment coefficients per control surface deflection, and a is non-dimensional distance from the mid-chord to the elastic axis. β is the control surface deflection.

Several classes of non-linear stiffness contributions $k_\alpha(\alpha)$ have been studied in papers treating the open-loop dynamics of aeroelastic systems [19, 20, 21, 22]. We now introduce the use of non-linear stiffness term $k_\alpha(\alpha)$ as obtained by curve-fitting on the measured displacement-moment data for non-linear spring as [23]:

$$k_\alpha(\alpha) = 2.82(1 - 22.1\alpha + 1315.5\alpha^2 + 8580\alpha^3 + 17289.7\alpha^4).$$

The equations of motion, derived above, exhibit limit cycle oscillation, as well as other non-linear response regimes including chaotic response [20, 21, 23]. The system parameters to be used in this paper are given in the Appendix and are obtained from experimental models described in full detail in works [12, 23].

With the flow velocity $u = 15(m/s)$ and the initial conditions of $\alpha = 0.1(rad)$ and $h = 0.01(m)$, the resulting time response of the non-linear system exhibits limit cycle oscillation, in good qualitative agreement with the behaviour expected in this class of systems. Papers [15, 23] have shown the relations between limit cycle oscillation, magnitudes and initial conditions or flow velocities.

Let the equations (7) and (8) be combined and reformulated into state-space

model form:

$$\mathbf{x}(t) = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} h \\ \alpha \\ \dot{h} \\ \dot{\alpha} \end{pmatrix} \quad \text{and} \quad \mathbf{u}(t) = \beta.$$

Then we have:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\mathbf{p}(t))\mathbf{x}(t) + \mathbf{B}(\mathbf{p}(t))\mathbf{u}(t) = \mathbf{S}(\mathbf{p}(t)) \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix}, \tag{9}$$

where

$$\mathbf{A}(\mathbf{p}(t)) = \begin{pmatrix} x_3 \\ x_4 \\ -k_1 x_1 - (k_2 U^2 + p(x_2))x_2 - c_1 x_3 - c_2 x_4 \\ -k_3 x_1 - (k_4 U^2 + q(x_2))x_2 - c_3 x_3 - c_4 x_4 \end{pmatrix}$$

$$\mathbf{B}(\mathbf{p}(t)) = \begin{pmatrix} 0 \\ 0 \\ g_3 U^2 \\ g_4 U^2 \end{pmatrix},$$

where $\mathbf{p}(t) \in \mathbb{R}^{N=2}$ contains values x_2 and U. The new variables are given in the Appendix. One should note that, the equations of motion are also dependent upon the elastic axis location a.

5 Observer design

The recently proposed very powerful numerical methods (and associated theory) for *convex optimization* involving Linear Matrix Inequalities (LMI) help us with the analysis and the design issues of dynamic systems models in acceptable computational time [24]. One direction of these analysis and design methods is based on LMI's under the PDC design framework [1]. In this paper we apply the TP model transformation in combination with the PDC based observer design technique to derive viable observer methodologies for the prototypical aeroelastic wing section defined in the previous section. The key idea of the proposed design method is that the TP model transformation is utilized to represent the model (9) in convex TP model form with specific characteristics, whereupon PDC controller design techniques can immediately be executed. The following sections introduces the observer design:

5.1 TP model form of the prototypical aeroelastic wing section

5.1.1 TP model transformation

The goal of the TP model transformation is to transform a given state-space model (1) into convex TP model [2, 3, 6], in which the LTI systems form a tight convex

hull. Namely, the TP model transformation results in (4) with conditions (5) and (6), and searches the LTI systems as a points of a tight convex hull of $\mathbf{S}(\mathbf{p}(t))$.

The detailed description of the TP model transformation is discussed in [2, 3, 6]. In the followings only the main steps are briefly presented. The TP model transformation is a numerical method and has three key steps. The first step is the discreatisation of the given $\mathbf{S}(\mathbf{p}(t))$ via the sampling of $\mathbf{S}(\mathbf{p}(t))$ over a huge number of points $\mathbf{p} \in \Omega$, where Ω is the transformation space. The sampling points are defined by a dense hyper rectangular grid. In order to loose minimal information during the discretisation we apply as dense grid as possible. The second step extracts the LTI vertex systems from the sampled systems. This step is specialized to find the minimal number of LTI vertex systems, as the vertex points of the tight convex hull of the sampled systems. The third step constructs the TP model based on the LTI vertex systems obtained in the second step. It defines the continuous weighting functions to the LTI vertex systems.

5.2 Determination of the convex TP model form of the aeroelastic model

We execute the TP model transformation on the model (9). First of all, according to the three steps of the TP model transformation, let us define the transformation space Ω. We are interested in the interval $U \in [14,25](m/s)$ and we presume that, the interval $\alpha \in [-0.1,0.1](rad)$ is sufficiently large enough. Therefore, let: $\Omega : [14,25] \times [-0.1,0.1]$ in the present example (note that these intervals can arbitrarily be defined). Let the grid density be defined as $M_1 \times M_2$, $M_1 = 100$ and $M_2 = 100$. Step 2 of the TP model transformation yields 6 vertex LTI systems (singular values are: first dimension: 16808, 1442 and 2; second dimension: 13040 and 7970):

$$\mathbf{A}_{1,1} = 10^3 \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.2314 & -0.0095 & -0.0034 & -0.0001 \\ 0.2780 & -1.1036 & 0.0071 & -0.0000 \end{pmatrix} \quad \mathbf{B}_{1,1} = \begin{pmatrix} 0 \\ 0 \\ -8.6 \\ -32.4 \end{pmatrix}$$

$$\mathbf{A}_{2,1} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -231.3804 & -46.3063 & -4.3776 & -0.2573 \\ 277.9906 & -966.7931 & 10.6520 & 0.4104 \end{pmatrix} \quad \mathbf{B}_{2,1} = \begin{pmatrix} 0 \\ 0 \\ -27.3677 \\ -103.4344 \end{pmatrix}$$

$$\mathbf{A}_{3,1} = 10^3 \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.2314 & -0.0227 & -0.0039 & -0.0002 \\ 0.2780 & -1.0543 & 0.0089 & 0.0002 \end{pmatrix} \quad \mathbf{B}_{3,1} = \begin{pmatrix} 0 \\ 0 \\ -15.4 \\ -58 \end{pmatrix}$$

$$\mathbf{A}_{1,2}=\begin{pmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ -231.3804 & -16.5786 & -3.4333 & -0.1425\\ 277.9906 & 23.0842 & 7.1447 & -0.0157\end{pmatrix}\quad \mathbf{B}_{1,2}=\begin{pmatrix}0\\ 0\\ -8.5825\\ -32.4370\end{pmatrix}$$

$$\mathbf{A}_{2,2}=\begin{pmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ -231.3804 & -53.4094 & -4.3776 & -0.2573\\ 277.9906 & 159.8695 & 10.6520 & 0.4104\end{pmatrix}\quad \mathbf{B}_{2,2}=\begin{pmatrix}0\\ 0\\ -27.3677\\ -103.4344\end{pmatrix}$$

$$\mathbf{A}_{3,2}=\begin{pmatrix}0 & 0 & 1.0000 & 0\\ 0 & 0 & 0 & 1.0000\\ -231.3804 & -29.8524 & -3.9054 & -0.1999\\ 277.9906 & 72.3823 & 8.8983 & 0.1974\end{pmatrix}\quad \mathbf{B}_{3,2}=\begin{pmatrix}0\\ 0\\ -15.3526\\ -58.0244\end{pmatrix}$$

The third step results in weighting functions $w_{1,i}(U)$ and $w_{2,j}(\alpha)$ depicted in Figure 2. When we numerically check the error between the model (9) and the resulting TP model, we find that the error is about 10^{-11} that is caused by the numerical computation.

In conclusion, the aeroelastic model (9) can be described exactly in finite convex TP form of 6 vertex LTI models, also see [6]. Note that, one may try to derive the weighting functions analytically from (9). The weighting functions of α can be extracted from $k_\alpha(\alpha)$. Finding the weighting functions of U, however, seems to be rather complicated. In spite of this, the computation of the TP model transformation takes a few seconds.

6 Observer design to the prototypical aeroelastic wing section

6.1 Method for observer design under PDC framework

In reality not all the state variables are readily available in most cases. Unavailable state variables should be estimated in the case of state-feedback control strategy. Under these circumstances, the question arises whether it is possible to determine the state from the system response to some input over some period of time. Namely, the observer is required to satisfy:

$$\mathbf{x}(t)-\hat{\mathbf{x}}(t)\to 0 \quad \text{as} \quad t\to\infty,$$

where $\hat{\mathbf{x}}(t)$ denotes the state vector estimated by the observer. This condition guaranties that the steady-state error between $\mathbf{x}(t)$ and $\hat{\mathbf{x}}(t)$ converges to 0. We use the following observer structure:

$$\dot{\hat{\mathbf{x}}}(t)=\mathbf{A}(\mathbf{p}(t))\hat{\mathbf{x}}(t)+\mathbf{B}(\mathbf{p}(t))\mathbf{u}(t)+\mathbf{K}(\mathbf{p}(t))(\mathbf{y}(t)-\hat{\mathbf{y}}(t))$$

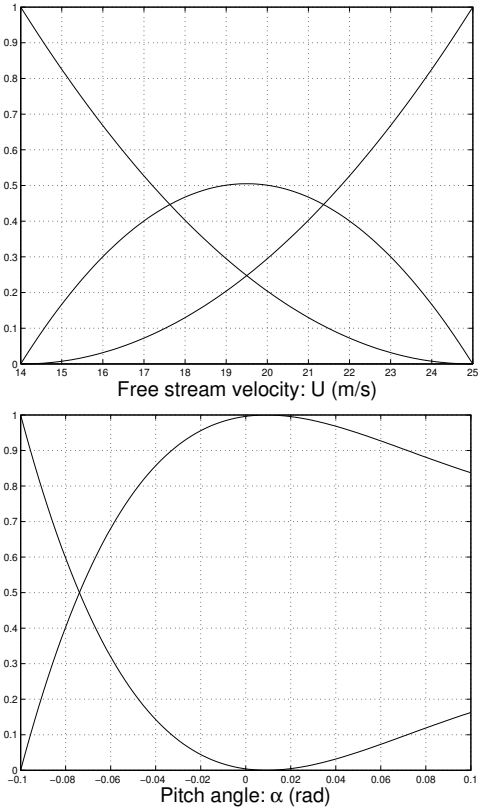


Figure 2: Weighting functions on the dimensions U and α.

$$\hat{\mathbf{y}}(t) = \mathbf{C}(\mathbf{p}(t))\hat{\mathbf{x}}(t),$$

That is in TP model form:

$$\begin{aligned}\dot{\hat{\mathbf{x}}}(t) = \mathcal{A} \underset{n}{\otimes} \mathbf{w}(p_n(t))\hat{\mathbf{x}}(t) + \mathcal{B} \underset{n}{\otimes} \mathbf{w}_n(p_n(t))\mathbf{u}(t) + \\ + \mathcal{K} \underset{n}{\otimes} \mathbf{w}(p_n(t))(\mathbf{y}(t) - \hat{\mathbf{y}}(t)) \\ \hat{\mathbf{y}}(t) = \mathcal{C} \underset{n}{\otimes} \mathbf{w}(p_n(t))\hat{\mathbf{x}}(t).\end{aligned} \tag{10}$$

At this point, we should emphasize that in our example the vector $\mathbf{p}(t)$ does not contain values form the estimated state-vector $\hat{\mathbf{x}}(t)$, since $p_1(t)$ equals U and $p_2(t)$ equals the pitch angle ($x_2(t)$). These variables are observable. We estimate only state-values $x_3(t)$ and $x_4(t)$. Consequently, the goal in the present case, is to determine gains in tensor $\mathcal{K}$ for (10). For this goal, the following LMI theorem can be find in [1]. Before dealing with this LMI theorem, we introduce a simple indexing technique, in order, to have direct link between the TP model form (4) and the typical form of LMI formulations:

Method 1 *(*Index transformation) *Let*

$$\mathbf{S}_r = \begin{pmatrix} \mathbf{A}_r & \mathbf{B}_r \\ \mathbf{C}_r & \mathbf{D}_r \end{pmatrix} = \mathbf{S}_{i_1,i_2,..,i_N},$$

where $r = ordering(i_1, i_2, .., i_N)$ ($r = 1..R = \prod_n I_n$). The function "ordering" results in the linear index equivalent of an N dimensional array's index $i_1, i_2, .., i_N$, when the size of the array is $I_1 \times I_2 \times .. \times I_N$. Let the weighting functions be defined according to the sequence of r:

$$w_r(\mathbf{p}(t)) = \prod_n w_{n,i_n}(p_n(t)).$$

Theorem 1 *(*Globally and asymptotically stable observer)

In order to ensure

$$\mathbf{x}(t) - \hat{\mathbf{x}}(t) \to 0 \quad as \quad t \to \infty,$$

in the observer strategy (10), find $\mathbf{P} > 0$ and $\mathbf{N}_r$ satisfying the following LMI's.

$$-\mathbf{A}_r^T\mathbf{P} - \mathbf{P}\mathbf{A}_r + \mathbf{C}_r^T\mathbf{N}_r^T + \mathbf{N}_r\mathbf{C}_r > 0 \tag{11}$$

for all r and

$$\begin{aligned}-\mathbf{A}_r^T\mathbf{P} - \mathbf{P}\mathbf{A}_r - \mathbf{A}_s^T\mathbf{P} - \mathbf{P}\mathbf{A}_s + \\ + \mathbf{C}_r^T\mathbf{N}_s^T + \mathbf{N}_s\mathbf{C}_r + \mathbf{C}_s^T\mathbf{N}_r^T + \mathbf{N}_r\mathbf{C}_s > 0.\end{aligned} \tag{12}$$

for $r < s \leq R$, except the pairs (r,s) such that $w_r(\mathbf{p}(t))w_s(\mathbf{p}(t)) = 0, \forall \mathbf{p}(t)$.

Since the above equations are LMI's, with respect to variables $\mathbf{P}$ and $\mathbf{N}_r$, we can find a positive definite matrix $\mathbf{P}$ and matrix $\mathbf{N}_r$ or determine that no such matrices exist. This is a convex feasibility problem. Numerically, this problem can be solved very efficiently by means of the most powerful tools available in the mathematical programming literature e.g. MATLAB-LMI toolbox [24].

The observer gains can then be obtained as:

$$\mathbf{K}_r = \mathbf{P}^{-1}\mathbf{N}_r. \tag{13}$$

Finally, by the help of $r = ordering(i_1, i_2, .., i_N)$ in Method 1 one can define $\mathbf{K}_{i_1,i_2,..,i_N}$ from $\mathbf{K}_r$ obtained in (13) and store into tensor $\mathcal{K}$ of (10).

6.2 Observer design to the prototypical aeroelastic wing section

This section applies Theorem 1 to the TP model of the aeroelastic wing section. We define matrix $\mathbf{C}$ for all r from:

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t),$$

that is in present case:

$$\mathbf{C}_r = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

The LMIs of Theorem 1, applied to the result of the TP model transformation, are feasible:

$$\mathbf{N}_1 = 10^7 \begin{pmatrix} 3.2142 & 0 & 0 & 0 \\ 0 & 3.2142 & 0 & 0 \\ 0 & 0 & 3.2142 & 0 \\ 0 & 0 & 0 & 3.2142 \end{pmatrix}$$

$$\mathbf{N}_2 = 10^8 \begin{pmatrix} 3.3743 & 0.1523 & 0.0358 & 0.0020 \\ 0.1523 & 1.4305 & -0.0233 & -0.0031 \\ 0.0358 & -0.0233 & 0.0196 & 0.0010 \\ 0.0020 & -0.0031 & 0.0010 & 0.0034 \end{pmatrix}$$

Thus, equ. (13) yields 6 observer feedbacks:

$$\mathbf{K}_{1,1} = \begin{pmatrix} 0.3691 & 0.6921 \\ -0.0027 & 0.7410 \\ -46.1240 & -21.6020 \\ 253.9914 & -676.5871 \end{pmatrix} \quad \mathbf{K}_{2,1} = \begin{pmatrix} 0.2796 & 0.9673 \\ 0.0664 & 0.6824 \\ -38.2972 & -57.1578 \\ 251.2960 & -542.4373 \end{pmatrix}$$

$$\mathbf{K}_{3,1} = \begin{pmatrix} 0.3234 & 0.7934 \\ 0.0405 & 0.7144 \\ -41.9448 & -34.5500 \\ 249.9595 & -628.0852 \end{pmatrix} \quad \mathbf{K}_{1,2} = \begin{pmatrix} 0.3449 & 0.0771 \\ -0.0336 & 1.2358 \\ -44.0427 & -31.0104 \\ 264.3622 & 448.0143 \end{pmatrix}$$

$$\mathbf{K}_{2,2} = \begin{pmatrix} 0.3006 & 0.3599 \\ 0.0197 & 1.0976 \\ -39.6387 & -64.5575 \\ 252.4420 & 585.8035 \end{pmatrix} \quad \mathbf{K}_{3,2} = \begin{pmatrix} 0.3169 & 0.1815 \\ 0.0008 & 1.1785 \\ -41.1822 & -43.3241 \\ 256.5618 & 498.5251 \end{pmatrix}.$$

In conclusion the state values $x_3(t)$ and $x_4(t)$ are estimated by (10) as:

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{A}(\mathbf{p}(t))\hat{\mathbf{x}}(t) + \mathbf{B}(\mathbf{p}(t))u(t) + \left(\sum_{i=1}^{3} \sum_{j=1}^{2} w_{1,i}(U) w_{2,j}(\alpha) \mathbf{k}_{i,j} \right) (\mathbf{y}(t) - \hat{\mathbf{y}}(t)),$$

where

$$\mathbf{y}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} \quad \text{and} \quad \hat{\mathbf{y}}(t) = \begin{pmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \end{pmatrix} \quad \text{and} \quad \mathbf{p}(t) = \begin{pmatrix} U \\ \alpha \end{pmatrix},$$

($x_1(t) = h$, plunge, and $x_2(t) = \alpha$, pitch). In order to demonstrate the accuracy of the observer, numerical experiments are presented in the next section.

6.3 Simulation results

We simulate the observer for initials $\mathbf{x}(0) = \begin{pmatrix} 0.01 & 0.1 & 0.1 & 0.1 \end{pmatrix}^T$ and $\hat{\mathbf{x}}(0) = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}^T$, for the open loop case. Figure 3 shows how the observer is capable of converging to the unmeasurable state values $x_3(t)$ and $x_4(t)$ (dashed line is estimated by the observer).

7 Conclusion

The paper presents how to use the TP model transformation method can be used for observer design in uniform way for controller and observer design. The paper also shows how to determine observer for the prototypical aeroelastic wing section.

Appendix

System parameters

$b = 0.135m$; $span = 0.6m$; $k_h = 2844.4N/m$; $c_h = 27.43Ns/m$; $c_\alpha = 0.036Ns$; $\rho = 1.225kg/m^3$; $c_{l_\alpha} = 6.28$; $c_{l_\beta} = 3.358$; $c_{m_\alpha} = (0.5+a)c_{l_\alpha}$; $c_{m_\beta} = -0.635$; $m = 12.387kg$; $x_\alpha = -0.3533 - a$; $I_\alpha = 0.065kgm^2$; $c_\alpha = 0.036$;

System variables

$d = m(I_\alpha - m x_\alpha^2 b^2)$;
$k_1 = \frac{I_\alpha k_h}{d}$; $k_2 = \frac{I_\alpha \rho b c_{l_\alpha} + m x_\alpha b^3 \rho c_{m_\alpha}}{d}$;

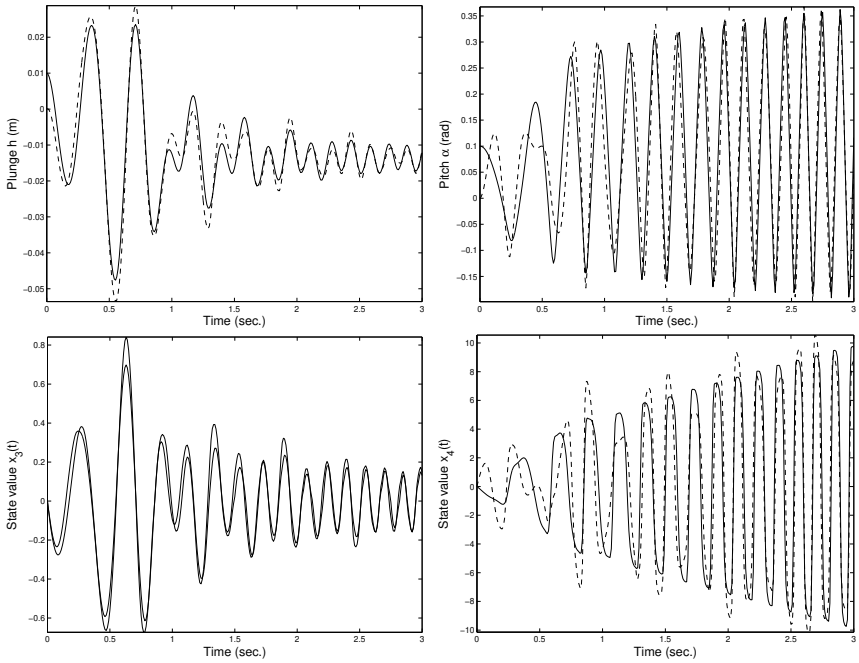


Figure 3: State values of $\mathbf{x}(t)$ (solid line) and the estimated values of $\hat{\mathbf{x}}(t)$ (dashed line) for open loop response. ($U = 20m/s$, $a = -0.4$, initials: $\mathbf{x}(0) = \begin{pmatrix} 0.01 & 0.1 & 0.1 & 0.1 \end{pmatrix}^T$, $\hat{\mathbf{x}}(0) = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}^T$)

$k_3 = \frac{-mx_\alpha bk_h}{d}$; $k_4 = \frac{-mx_\alpha b^2 \rho c_{l_\alpha} - m\rho b^2 c_{m_\alpha}}{d}$;
$p(\alpha) = \frac{-mx_\alpha b}{d} k_\alpha(\alpha)$; $q(\alpha) = \frac{m}{d} k_\alpha(\alpha)$;
$c_1(U) = \left(I_\alpha(c_h + \rho U b c_{l_\alpha}) + mx_\alpha \rho U^3 c_{m_\alpha}\right)/d$;
$c_2(U) = \left(I_\alpha \rho U b^2 c_{l_\alpha}(\frac{1}{2} - a) - mx_\alpha b c_\alpha + mx_\alpha \rho U b^4 c_{m_\alpha}(\frac{1}{2} - a)\right)/d$;
$c_3(U) = \left(-mx_\alpha b c_h - mx_\alpha \rho U b^2 c_{l_\alpha} - m\rho U b^2 c_{m_\alpha}\right)/d$;
$c_4(U) = \left(mc_\alpha - mx_\alpha \rho U b^3 c_{l_\alpha}(\frac{1}{2} - a) - m\rho U b^3 c_{m_\alpha}(\frac{1}{2} - a)\right)/d$;
$g_3 = (-I_\alpha \rho b c_{l_\beta} - mx_\alpha b^3 \rho c_{m_\beta})/d$;
$g_4 = (mx_\alpha b^2 \rho c_{l_\beta} + m\rho b^2 c_{m_\beta})/d$;

References

[1] K. Tanaka and H. O. Wang, *Fuzzy Control Systems Design and Analysis - A Linear Matrix Inequality Approach*, Hohn Wiley and Sons, Inc., 2001, 2001.

[2] P. Baranyi, D. Tikk, Y. Yam, and R. J. patton, "From differential equations to PDC controller design via numerical transformation," *Computers in Industry, Elsevier Science*, vol. 51, pp. 281–297, 2003.

[3] P. Baranyi, "TP model transformation as a way to LMI based controller design," *IEEE Transaction on Industrial Electronics*, vol. 51, no. 2, April 2004.

[4] I. J. Rudas and O. Kaynak, "Entropy-based operations on fuzzy sets," *IEEE Transactions on Fuzzy Systems*, vol. 6, 1998.

[5] I. J. Rudas and O. Kaynak, "Minimum and maximum fuzziness generalized operators," *Fuzzy Sets and Systems*, vol. 98, 1998.

[6] P. Baranyi, P. Korondi, R.J. Patton, and H. Hashimoto, "Global asymptotic stabilisation of the prototypical aeroelastic wing section via tp model transfromation," *Asian Journal of Control*, vol. (to be printed in Vol. 7, No. 2, 2004).

[7] P. Baranyi and R.J. Patton, "A numerical control design method for prototypical aeroelastic wing section with structural non-linearity," in *European Control Conference (ECC'03)*, University of Cambridge, UK, 2003.

[8] P. Baranyi, P. Michekberger, and A.R. Várkonyi-Kóczy, "Numerical control design for aeroelastic systems," in *2nd Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence (SAMI 2004)*, Herl'any, Slovakia, 16-17 January, 2004, pp. 43–50.

[9] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "A multi linear singular value decomposition," SIAM *Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[10] D. Tikk, P. Baranyi, and R.J.Patton, "Polytopic and TS models are nowere dense in the approximation model space," *IEEE Int. Conf. System Man and Cybernetics (SMC'02)*, 2002, Proc. on CD.

[11] D. Tikk, P. Baranyi, R.J. Patton, and J. Tar, "Approximation capability of tp model forms," *Australian Journal of Intelligent Information Processing Systems (accepted for publication)*, 2004.

[12] J. Ko, A. J. Kurdila, and T. W. Strganac, "Nonlinear control of a prototypical wing section with torsional nonlinearity," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 6, pp. 1181–1189, November–December 1997.

[13] J. Ko, A. J. Kridula, and T. W. Strganac, "Nonlinear control theory for a class of structural nonlinearities in a prototypical wing section," *Proc. of the 35th AIAA Aerospace Scinece Meeting and Exhibit, AIAA paper 97-0580*, 1997.

[14] J. J. Block and T. W. Strganac, "Applied active control for nonlinear aeroelastic structure," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 6, pp. 838–845, November–December 1998.

[15] T. O'Neil, H. C. Gilliat, and T. W. Strganac, "Investigatiosn of aeroelsatic response for a system with continuous structural nonlinearities," *Proc. of the 37th AIAA Structures, Structural Dynamics, and Matirials Conference, AIAA Paper 96-1390*, 1996.

[16] J. Ko, A. J. Kurdila, and T. W. Strganac, "Nonlinear dynamics and control for a structurally nonlinear aeroelastic system," *Proc. of the 38th AIAA Structures, Structural Dynamics, and Matirials Conference, AIAA Paper 97-1024*, 1997.

[17] Y. C. Fung, *An Introduction to the Theory of Aeroelasticity*, John Wiley and Sons, New York, 1955.

[18] E. H. Dowell (Editor), H. C. Jr. Curtiss, R. H. Scanlan, and F. Sisto, *A Modern Course in Aeroelasticity*, Stifthoff and Noordhoff, Alpen aan den Rijn, The Netherlands, 1978.

[19] D. M. Tang and E. H. Dowell, "Flutter and stall response of a helicopter blade with structural nonlinearity," *Journal of Aircraft*, vol. 29, pp. 953–960, 1992.

[20] L. C. Zhao and Z. C. Yang, "Chaotic motions of an airfoil with nonlinear stiffness in incompressible flow," *Journal of Sound and Vibration*, vol. 138, no. 2, pp. 245–254, 1990.

[21] E. H. Dowell, "Nonlinear aeroelasticity," *Proc. of the 31th AIAA Structures, Structural Dynamics, and Matirials Conference, AIAA Paper 97-1024*, pp. 1497–1509, 1990.

[22] Z. C. Yang and L. C. Zhao, "Analysis of limit cycle flutter of an airfoil in incompressible flow," *Journal of Sound and Vibration*, vol. 123, no. 1, pp. 1–13, 1988.

[23] T. O'Neil and T. W. Strganac, "An experimental investigation of nonlinear aeroelastic respons," *AIAA paper 95-1404, proc. 36th AIAA / ASME / ASCE/ AHS / ASC Structures, Structural Dynamics, and Materials Conference, New Orleans, Lousiana*, pp. 2043–2051, 1995.

[24] P. Gahinet, A. Nemirovski, A.J.Laub, and M.Chilali, LMI *Control Toolbox*, The MathWorks, Inc., 1995.

Computation of Boundary Layers

József Dénes, István Patkó

Budapest Tech
Doberdó út 6, H-1034 Budapest, Hungary, E-mail:
denes.jozsef@nik.bmf.hu
patko@bmf.hu

Abstract:This paper is the first part of a series of studies where we examine several methods for the solution of the boundary layer equation of the fluid mechanics. The first of these is the analytical or rather quasi analytical method due to Blasius. This method reduces a system of partial differential equations to a system of ordinary differential equations and these in turn are solved by numerical methods since no exact solution of the Blasius type equations is known. We determind all the Blasius equation neccessary for up to 11-th order approximation. Our further aim to study the finite difference numerical solutions of the boundary layer equation and some of the methods applying weighted residual principles and by comparing these with the "exact" solutions arrived at by Blasius method develop a quick reliable method for solving the boundary layer equation.

Keywords: Boundary Layer, Blasius Method, Boundary Layer equation

1 Boundary Layer

The motion of a fluid around a solid body according to Prandtl (1904) can be described by the Euler equation of the perfect (that is nonviscous) fluid motion except in a thin layer near the surface of the solid body where the speed of the motion increases from zero to the speed that would be in case if the fluid had no viscosity at all. Outside the boundary layer the fluid may be considered as nonviscous. This is the case when the velocity of the fluid in the direction of the flow around the body increases. When it decreases that is the pressure increases, often the fluid motion unable to follow the bodies' surface and it gets detached and the space between the surface of the solid and the detached fluid is filled with irregularly moving fluid. Prandt's theory of boundary layer, more precisly his equations describing the motion within the boundary layer can predict the point(s) of detachment accurately. The detachment begins where the curve of the velocity profile starts out perpendiclar to the surface of the solid. After this point a backward flow develops. The typical values used for describing the boundary layer are:

δ: Boundary layer thickness is the distance measured from the surface of the solid where the speed of the fluid is within 1% of the speed outside of the boundary layer.

δ_1: Displacement thickness $\delta_1 = \frac{1}{U}\int_0^\infty (U-u)dy$

δ_2: Impulseloss thickness $\delta_2 = \frac{1}{U^2}\int_0^\infty (U-u)udy$

δ^{**}: Energyloss thickness $\delta^{**} = \frac{1}{U^3}\int_0^\infty (U-u)^2 udy$

Profile parameter $H_{1,2} = \frac{\delta^{**}}{\delta_2}$

2 The Eqations of the Boundary Layer Flow

Inside the boundary layer that is in the vicinity of the body the forces due to viscosity are comparable in magnitude with the forces of inertia they can however be neglected outside of it. The pressure in the boundary layer could be taken as constant and its value equal to the pressure belonging to the corresponding perfect fluid flow, that is the pressure outside of the boundary layer. Without going into more details we give the equations of the boundary layer motion in case of two dimensional staionary incompressible fluid flow:

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = U\frac{dU}{dx} + \nu\frac{\partial^2 u}{\partial y}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

where u is the velocity of the fluid in the boundary layer parallel to the tangent of the surface of the solid v is perpendicular to it and U is the velocity outside of the boundary layer. u and v must also satisfy the boundary conditions: $y = 0$: $u = 0$ $v = 0$ and at $y = \infty$ $u = U(x)$.

3 Blasius' Method for Solving the Boundary Layer Equation

The boundary layer equations can be reduced to an infinite system of ordinary differential equations with the following method due to Blasius. By substituting for the velocity components $u = \frac{\partial \psi}{\partial y}$ and $v = -\frac{\partial \psi}{\partial x}$, where ψ is the stream function of Lagrange the second of the two boundary layer equations is automaticly satisfied and the first one becomes:

$$\psi_y \psi_{xy} - \psi_x \psi_{yy} = U \frac{dU}{dx} + \nu \psi_{yyy}.$$

This equation then can be reduced to a set of ordinary differential equation if for Ψ in case of symmetric bodies the following power series expansion is substituted:

$$\psi = \sqrt{\frac{\nu}{u_1}} \{u_1 x f_1(\eta) + 4u_3 x^3 f_3(\eta) + 6u_5 x^5 f_5(\eta) + 8u_7 x^7 f_7(\eta) +$$
$$+ 10u_9 x^9 f_9(\eta) + 12u_{11} x^{11} f_{11}(\eta) + ...$$

where $\eta = y\sqrt{\frac{u_1}{\nu}}$ .and $u_1, u_3, \cdots\cdots$ are the coefficients in the power series expansion of $U(x)$, that is:

$$U = u_1 x + u_3 x^3 + u_5 x^5 + u_7 x^7 + u_9 x^9 + u_{11} x^{11} + u_{13} x^{13} + u_{15} x^{15} +$$
$$+ u_{17} x^{17} + u_{19} x^{19} + ...$$

From the last two equations it follows that:

$$U \frac{dU}{dx} = u_1^2 x + 4u_1 u_3 x^3 + \left(6u_1 u_5 + 5u_3^2\right) x^5 + \left(8u_1 u_7 + 8u_3 u_5\right) x^7 +$$
$$\left(10u_1 u_9 + 10u_3 u_7 + 5u_5^2\right) x^9 + \left(12u_1 u_{11} + 12u_3 u_9 + 12u_5 u_7\right) x^{11} + ...$$

$$\psi_x = \sqrt{\frac{\nu}{u_1}}\{u_1 f_1(\eta) + 12u_3 x^2 f_3(\eta) + 30u_5 x^4 f_5(\eta) + 56u_7 x^6 f_7(\eta) + 90u_9 x^8 f_9(\eta) +$$
$$+ 132u_{11} x^{10} f_{11}(\eta) + ...\}.$$

$$\psi_y = u_1 x f_1^{'}(\eta) + 4u_3 x^3 f_3^{'}(\eta) + 6u_5 x^5 f_5^{'}(\eta) + 8u_7 x^7 f_7^{'}(\eta) + 10u_9 x^9 f_9^{'}(\eta) +$$
$$+ 12u_{11} x^{11} f_{11}^{'}(\eta) + ...$$

$$\psi_{xy} = u_1 f_1^{'}(\eta) + 12u_3 x^2 f_3^{'}(\eta) + 30u_5 x^4 f_5^{'}(\eta) + 56u_7 x^6 f_7^{'}(\eta) + 90u_9 x^8 f_9^{'}(\eta) +$$
$$+ 132u_{11} x^{10} f_{11}^{'}(\eta) + ...$$

$$\psi_{yy} = \sqrt{\frac{u_1}{\nu}}\{u_1 x f_1^{''}(\eta) + 4u_3 x^3 f_3^{''}(\eta) + 6u_5 x^5 f_5^{''}(\eta) + 8u_7 x^7 f_7^{''}(\eta) +$$
$$+ 10u_9 x^9 f_9^{''}(\eta) + 12u_{11} x^{11} f_{11}^{''}(\eta) + ...$$

$$\psi_{yyy} = \frac{u_1}{\nu}(u_1 x f_1^{'''}(\eta) + 4u_3 x^3 f_3^{'''}(\eta) + 6u_5 x^5 f_5^{'''}(\eta) + 8u_7 x^7 f_7^{'''}(\eta) +$$
$$+ 10u_9 x^9 f_9^{'''}(\eta) + 12u_{11} x^{11} f_{11}^{'''}(\eta) + ...)$$

and

$$\psi_y \psi_{xy} = u_1^2 x f_1^{'2} + 12u_1 u_3 f_1^{'} f_3^{'} x^3 + (36u_1 u_5 f_1^{'} f_5^{'} + 48u_3^2 f_3^{'2}) x^5 +$$
$$(64u_1 u_7 f_1^{'} f_7^{'} + 192u_3 u_5 f_3^{'} f_5^{'}) x^7 +$$
$$(100u_1 u_9 f_1^{'} f_9^{'} + 320u_3 u_7 f_3^{'} f_7^{'} + 180u_5^2 f_5^{'2}) x^9 +$$
$$(144u_1 u_{11} f_1^{'} f_{11}^{'} + 480u_3 u_9 f_3^{'} f_9^{'} + 576u_5 u_7 f_5^{'} f_7^{'}) x^{11} + ...$$

$$\psi_x \psi_{yy} = u_1^2 f_1 f_1^{''} x + (4u_1 u_3 f_1 f_3^{''} + 12u_1 u_3 f_3 f_3^{''}) x^3 +$$
$$(6u_1 u_5 f_1 f_5^{''} + 48u_3^2 f_3 f_3^{''} + 30u_1 u_5 f_5 f_1^{''}) x^5 +$$
$$(8u_1 u_7 f_1 f_7^{''} + 72u_3 u_5 f_3 f_5^{''} + 120u_3 u_5 f_5 f_3^{''} + 56u_1 u_7 f_7 f_1^{''}) x^7 +$$
$$(10u_1 u_9 f_1 f_9^{''} + 96u_3 u_7 f_3 f_7^{''} + 180u_5^2 f_5 f_5^{''} + 224u_3 u_7 f_7 f_3^{''} + 90u_1 u_9 f_9 f_1^{''}) x^9 +$$
$$(12u_1 u_{11} f_1 f_{11}^{''} + 120u_3 u_9 f_3 f_9^{''} + 240u_5 u_7 f_5 f_7^{''} + 336u_5 u_7 f_7 f_5^{''} +$$
$$360u_9 u_3 f_9 f_3^{''} + 132u_{11} u_1 f_9 f_1^{''}) x^{11} + ...$$

finally substituting these into the bondary layer equation:

$$\psi_y \psi_{xy} - \psi_x \psi_{yy} = U U_x + \psi_{yyy}$$

and comparing the coefficients of the powers of x we get the differential equations of Blasius:

Comparing the coefficients of x yields:

$$u_1^2 f_1'^2 - u_1^2 f_1 f_1'' = u_1^2 + u_1^2 f_{11}'''$$

and from here we get:

$$f_1'^2 - f_1 f_1'' = 1 + f_1'''$$

From the coefficients of x^3 we get:

$$12u_1u_3 f_1' f_3' - (4u_1u_3 f_1 f_3'' + 12u_1u_3 f_1'' f_3) = 4u_1u_3 + 4u_1u_3 f_3'''$$

that is:

$$3f_1' f_3' - f_1 f_3'' - 3f_1'' f_3 = 1 + f_3'''$$

As for x^5:

$$36u_1u_5 f_1' f_5' + 48u_3^2 f_3'^2 - (6u_1u_5 f_1 f_5'' + 48u_3^2 f_3 f_3'' + 30u_1u_5 f_1'' f_5) =$$
$$6u_1u_5 + 3u_3^2 + 6u_1u_5 f_5'''$$

Dividing by $6u_1u_5$:

$$6f_1' f_5' + 8\frac{u_3^2}{u_1u_5} f_3'^2 - (f_1 f_5'' + 8\frac{u_3^2}{u_1u_5} f_3 f_3'' + 5f_1'' f_5) = 1 + \frac{1}{2}\frac{u_3^2}{u_1u_5} + f_5'''$$

If we seek f_5 in the form

$f_5 = g_5 + \frac{u_3^2}{u_1u_5} h_5$ we get for g_5 and h_5 the following differentialequations:

$$6f_1' g_5' - f_1 g_5'' - 5f_1'' g_5 = 1 + g_5'''$$

$$6f_1' h_5' + 8f_3'^2 - f_1 h_5'' - 5f_1'' h_5 - 8f_3 f_3'' = \frac{1}{2} + h_5'''$$

Case x^7. This case is sufficiently complex to demonstrate the method of finding the Blasius type diffrential equations for the general case that is for x^n where n is an arbitrary odd integer. By comparing the coefficients of x^7 we get:

$$(64u_1u_7f_1'f_7' + 192u_3u_5f_3'f_5') -$$
$$(8u_1u_7f_1f_7''' + 72u_3u_5f_3f_5''' + 120u_5u_3f_5f_3''' + 56u_7u_1f_7f_1''') =$$
$$(8u_1u_7 + 8u_3u_5) + 8u_1u_7f_7''''$$

Dividing by $8u_1u_7$ gives:

$$(8f_1'f_7' + 24\frac{u_3u_5}{u_1u_7}f_3'f_5') - (f_1f_7''' + 9\frac{u_3u_5}{u_1u_7}f_3f_5''' + 15\frac{u_3u_5}{u_1u_7}f_5f_3''' + 7f_7f_1''') =$$
$$(1 + \frac{u_3u_5}{u_1u_7}) + f_7''''$$

Let us seek f_7 in the form $f_7 = g_7 + \frac{u_3u_5}{u_1u_7}\tilde{h}_7$. Substituting this into the last equation yields for g_7 the following ordinary differential equation:

$$8f_1'g_7' - f_1g_7''' - 7f_7f_1''' = 1 + g_7''''$$

and for $\tilde{h}_7$

$$8\frac{u_3u_5}{u_1u_7}f_1'\tilde{h}_7' + 24\frac{u_3u_5}{u_1u_7}f_3'(g_5' + \frac{u_3^2}{u_1u_5}h_5') - \frac{u_3u_5}{u_1u_7}f_1\tilde{h}_7''' -$$
$$9\frac{u_3u_5}{u_1u_7}f_3(g_5''' + \frac{u_3^2}{u_1u_5}h_5''') - 15\frac{u_3u_5}{u_1u_7}f_3'''(g_5 + \frac{u_3^2}{u_1u_5}h_5) - 7\tilde{h}_7f_1''' =$$
$$\frac{u_3u_5}{u_1u_7} + \frac{u_3u_5}{u_1u_7}\tilde{h}_7''''$$

Substituting for $\tilde{h}_7$ $\tilde{h}_7 = h_7 + \frac{u_3^2}{u_1u_7}k_7$ yields for h_7

$$8f_1'h_7' + 24f_3'g_5' - f_1h_7''' - 9f_3g_5''' - 15f_3'''g_5 - 7f_1'''h_7 = 1 + h_7''''$$

and for k_7

$$8f_1'k_7' + 24f_3'h_5' - f_1k_7''' - 9f_3h_5''' - 15f_3'''h_5 - 7f_1'''k_7 = k_7''''$$

With the same method we can arrive at the equations for all the f_n Blasius functions. With n increasing f_n has to be broken down into more and more terms. Here the forms of f_n and the corresponding differential equations are given up to the order of 11.

$$f_5 = g_5 + \frac{u_3^2}{u_1 u_5} h_5$$

$$f_7 = g_7 + \frac{u_3 u_5}{u_1 u_7} h_7 + \frac{u_3^2}{u_1^2 u_7} k_7$$

$$f_9 = g_9 + \frac{u_3 u_7}{u_1 u_9} h_9 + \frac{u_5^2}{u_1 u_9} k_9 + \frac{u_3^2 u_5}{u_1^2 u_9} j_9 + \frac{u_3^4}{u_1^3 u_9} q_9$$

$$f_{11} = g_{11} + \frac{u_3 u_9}{u_1 u_{11}} h_{11} + \frac{u_5 u_7}{u_1 u_{11}} k_{11} + \frac{u_3^2 u_7}{u_1^2 u_{11}} j_{11} + \frac{u_3 u_5^2}{u_1^2 u_{11}} q_{11} + \frac{u_3^3 u_5}{u_1^3 u_{11}} m_{11} + \frac{u_3^5}{u_1^4 u_{11}} n_{11}$$

The differential equatios that the functions f_1, f_3, g_5, h_5, g_7, h_7, k_7, g_9, h_9, k_9, j_9, q_9, g_{11}, h_{11}, k_{11}, j_{11}, q_{11}, m_{11}, n_{11} have to satisfy are:

$$f_1'^2 - f_1 f_1'' = 1 + f_1'''$$

$$3 f_1' f_3' - f_1 f_3'' - 3 f_1'' f_3 = 1 + f_3'''$$

$$6 f_1' g_5' - f_1 g_5'' - 5 f_1'' g_5 = 1 + g_5'''$$

$$6 f_1' h_5' + 8 f_3'^2 - f_1 h_5'' - 5 f_1'' h_5 - 8 f_3 f_3'' = \frac{1}{2} + h_5'''$$

$$8 f_1' g_7' - f_1 g_7'' - 7 f_7 f_1'' = 1 + g_7'''$$

$$8 f_1' h_7' + 24 f_3' g_5' - f_1 h_7'' - 9 f_3 g_5'' - 15 f_3'' g_5 - 7 f_1'' h_7 = 1 + h_7'''$$

$$8 f_1' k_7' + 24 f_3' h_5' - f_1 k_7'' - 9 f_3 h_5'' - 15 f_3'' h_5 - 7 f_1'' k_7 = k_7'''$$

$$10 f_1' g_9' - f_1 g_9'' - 9 f_1'' g_9 = 1 + g_9'''$$

$$10f_1'h_9' + 32f_3'g_7' - f_1h_9'' - 9{,}6f_3g_7'' - 22{,}4f_3''g_7 - 9f_1''h_9 = 1 + h_9'''$$

$$10f_1'k_9' + 18g_5'^2 - f_1k_9'' - 18g_5g_5'' - 9f_1''k_9 = \frac{1}{2} + k_9'''$$

$$10f_1'j_9' + 32f_3'h_7' + 36g_5'h_5' - f_1j_9'' - 9{,}6f_3h_7'' - 18g_5h_5'' - 18g_5''h_5 - 22{,}4f_3''h_7 - 9f_1''j_9 = j_9'''$$

$$10f_1'q_9' + 32f_3'k_7' + 18h_5'^2 - f_1q_9'' - 9{,}6f_3k_7'' - 18h_5h_5'' - 22{,}4f_3''k_7 - 9f_1''q_9 = q_9'''$$

$$12f_1'g_{11}' - f_1g_{11}'' - 11f_1''g_{11} = 1 + g_{11}'''$$

$$12f_1'h_{11}' + 40f_3'g_9' - f_1h_{11}'' - 10f_3g_9'' - 30f_3''g_9 - 11f_1''h_{11} = 1 + h_{11}'''$$

$$12f_1'k_{11}' + 48g_5g_7' - f_1k_{11}'' - 20g_5g_7'' - 28g_5''g_7 - 11f_1''k_{11} = 1 + k_{11}'''$$

$$12f_1'j_{11}' + 40f_3'h_9' + 48h_5'g_7' - f_1j_{11}'' - 10f_3h_9'' - 20h_5g_7'' - 28h_5''g_7 - 30f_3''h_9 - 11f_1''j_{11} = j_{11}'''$$

$$12f_1'q_{11}' + 40f_3'k_9' + 48g_5'h_7' - f_1q_{11}'' - 10f_3k_9'' - 20g_5h_7'' - 28g_5''h_7 - 30f_3''k_9 - 11f_1''q_{11} = q_{11}'''$$

$$12f_1'm_{11}' + 40f_3'j_9' + 48h_5'k_7' + 48g_5'h_7' - f_1m_{11}'' - 10f_3j_9'' - 20h_5h_7'' - 20g_5k_7'' - 28g_5''k_7 - 28h_5''h_7 - 30f_3''j_9 - 11f_1''m_{11} = m_{11}'''$$

$$12f_1'n_{11}' + 40f_3'q_9' - f_1n_{11}'' - 10f_3q_9'' - 20h_5k_7'' - 28h_5''k_7 - 30f_3''q_9 - 11f_1''n_{11} = n_{11}'''$$

The $u(x,y) = 0$ at $y = 0$ and $\lim_{y\to\infty} u(x,y) = U(x)$ boundary conditios in terms of f_1, f_3, g_5, h_5, g_7, h_7, k_7, g_9, h_9, k_9, j_9, q_9, g_{11}, h_{11}, k_{11}, j_{11}, q_{11}, m_{11}, n_{11} become

at $\eta = 0$

$$f_1 = f_1' = 0; \quad f_3 = f_3' = 0; \quad g_5 = g_5' = 0; \quad h_5 = h_5' = 0; \quad g_7 = g_7' = 0;$$
$$h_7 = h_7' = 0; \quad k_7 = k_7' = 0; \quad g_9 = g_9' = 0; \quad h_9 = h_9' = 0; \quad k_9 = k_9' = 0;$$

$j_9 = j_9^{'} = 0;\ q_9 = q_9^{'} = 0;\ \ g_{11} = g_{11}^{'} = 0;\ \ h_{11} = h_{11}^{'} = 0;\ \ k_{11} = k_{11}^{'} = 0;$
$j_{11} = j_{11}^{'} = 0;\ \ q_{11} = q_{11}^{'} = 0;\ \ m_{11} = m_{11}^{'} = 0;\ \ n_{11} = n_{11}^{'} = 0;$

and at $\eta = \infty$

$$f_1^{'} = 1;\ \ f_3^{'} = \frac{1}{4};\ \ g_5^{'} = \frac{1}{6};\ \ h_5^{'} = 0;\ \ g_7^{'} = \frac{1}{8};\ \ h_7^{'} = 0;\ \ k_7^{'} = 0;\ \ g_9^{'} = \frac{1}{10};$$
$$h_9^{'} = 0;\ \ k_9^{'} = 0;\ \ j_9^{'} = 0;\ q_9^{'} = 0;\ \ g_{11}^{'} = \frac{1}{12};\ \ h_{11}^{'} = 0;\ \ k_{11}^{'} = 0;\ \ j_{11}^{'} = 0;$$
$$q_{11}^{'} = 0;\ \ m_{11}^{'} = 0;\ \ n_{11}^{'} = 0;$$

Conclusion

We have derived the ordinary differential equations necessary to carry out numerical approximation to the solution of the boundary layer equation by 11-th order Blasius method.

References

[1] L. Prandtl, Über Flüssigkeitsbewegung bei sehr kleiner Reibung. Proceedings 3rd Intern. Math. Congr. Heidelberg 1904, 484-491. Reprinted in Coll. Works II 575-584.

[2] H. Blasius, Grenzschichten in Flüssigkeiten mit kleiner Reibung. Z. Math. u. Phys. 56, 1-37 (1908).

[3] L. Prandtl, Über Flüssigkeitsbewegung bei sehr kleiner Reibung. Proceedings 3rd Intern. Math. Congr. Heidelberg 1904, 484-491. Reprinted in Coll. Works II 575-584.

[4] H. Blasius, Grenzschichten in Flüssigkeiten mit kleiner Reibung. Z. Math. u. Phys. 56, 1-37 (1908).

On the Complexity of the Channel Routing Problem in the Dogleg-free Multilayer Manhattan Model

Kornélia Ambrus Somogyi

Budapest Tech, email: ambrusne.somogyi.kornelia@nik.bmf.hu

András Recski[1]

Budapest University of Technology and Economics, email: recski@cs.bme.hu

Abstract:Let l_H *and* l_V *denote the number of layers reserved for horizontal (vertical) wire segments in the multilayer dogleg-free Manhattan model.* $[d/l_H]$ *is a lower bound for the minimum width for routing a channel of density* d *where* $[x]$ *denotes the upper integer part of* x*. A greedy interval packing algorithm realizes every channel routing problem with width* $[d/(l_V-1)]$ *in linear time if* $l_V \geq 2$*.*

In case $l_V=l_H+1$ *the resulting width is* $[d/l_H]$*, hence it is best possible.*

The second author has shown that in case $l_V=l_H=k>1$*, it is NP-complete to decide whether a channel routing problem can be solved with width* $[d/k]$ *in the dogleg-free* $2k$*-layer Manhattan model.*

Here we turn to the remaining case. In the special case $l_V=1$ *and* $l_H=2$ *we show its relation to the NP-complete problem for the usual 2-layers Manhattan model and we point out a relation of the routing problem to a 2-processor job sheduling problem.*

Keywords:VLSI, detailed routing,channel routing, Manhattan model, multilayer routing.

1 Basic Definitions

A channel is a rectangular grid of rows and columns.

- **n** denotes the length of the channel (number of the columns),
- **w** denotes the width of the channel, the width of a certain routing (number of the rows).

[1] Research partially supported by the Hungarian National Science Fund (Grant No. OTKA T 42559)

The rows of the routing will be called tracks.

The principal direction of the channel will be called horizontal, this means, the pins of the devices of the electric equipment are placed on the Northern and Southern boundaries of the channel. These pins are called terminals (or nodes) and the important task of the routing: to interconnect some terminals by wires. A subset of terminals to be interconnected will be called a net. In the examples terminals of the same net are denoted by the same number.

If $X_{i,l}$ and $X_{i,r}$ denote the X-coordinates of the leftmost and the rightmost terminal of net N_i then the interval $[X_{i,l}, X_{i,r}]$ will be called the interval of the net N_i If a net has only two opposite terminals then the interval reduces to a single point. Such a net is called trivial like N_5 in Figure 1.

A routing is dogleg-free if the realization of each nontrivial net contains a single horizontal wire segment only. (For example net N_1 of Figure 1 is dogleg-free, N_3 has a dogleg.) The congestion of a vertical line is the number of nets, whose intervals intersect the line. The maximum congestion is called the density of the channel routing problem and is denoted by ***d***.

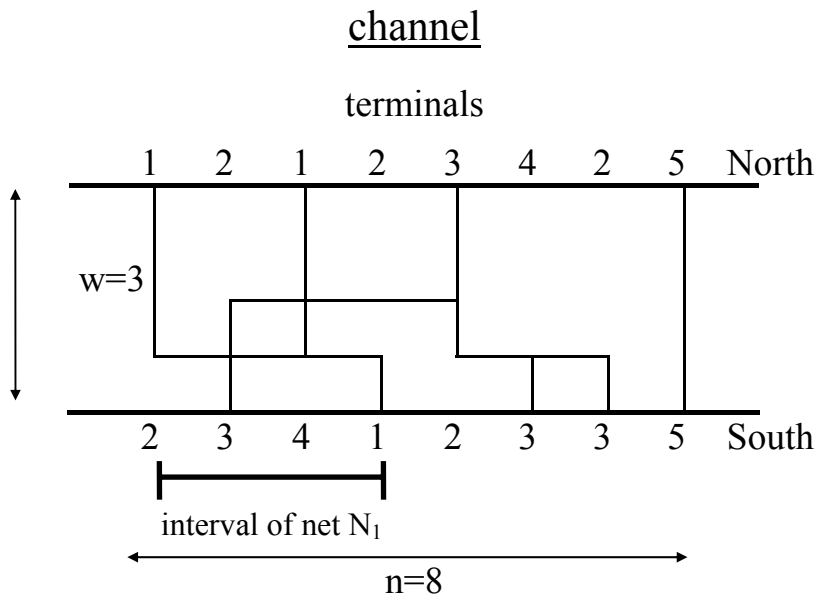


Figure 1

Let ***l*** denote the number of layers in the routing, and we consider the „Manhattan model". The „usual" Manhattan model has two layers for the routing, one layer for the horizontal and one layer for the vertical wire segments. The multilayer Manhattan model has ***l*** layers, let l_H denote the number of layers reserved for

horizontal and l_V denote the number of layers reserved for vertical wire segments. Consecutive layers in the Manhattan model may contain wire segments of different direction only.

A gridpoint, where the wire turns from horizontal to vertical direction (or vice versa), and the wire must leave a layer for another adjacent one is called via. In case $l=2$ one layer is reserved for horizontal, and one for vertical wire segments. If l is even, the number of horizontal and vertical layers are equal. If l is odd, the number of horizontal and vertical layers are different. For example in case $l=3$ we can distinguish between **HVH** and **VHV** models. In general for any odd number l one has two types of l -layer- Manhattan models.

In the multilayer Manhattan model we have a lower bound for the minimum width for routing a channel of density d: $[d/l_H]$, where $[x]$ denotes the upper integer part of x. We also have an upper bound for $l_V \geq 2$: a greedy interval packing algorithm gives a solution with width $[d/(l_V - 1)]$ in linear time. It is the best possible if $l_V = l_H + 1$.

2 Gallai's Algorithm

First let us suppose that all the terminals of the nets are on the Northern boundary of the channel. This case is the so called single row routing problem [10]. In this case the intervals of the nets can be packed into d horizontal lines, so called tracks, by a greedy interval packing algorithm:

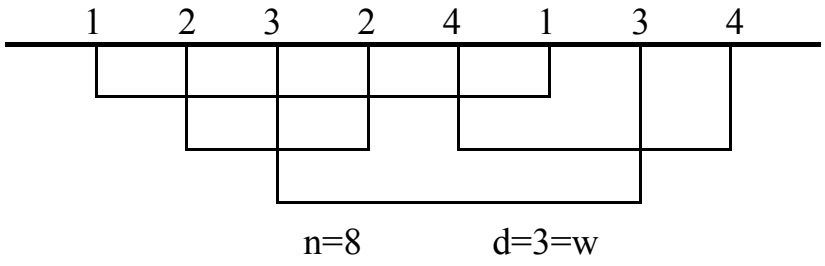


Figure 2

Let us consider the list **L** of the interval of the nets.

Step 1 If the list **L** is empty, stop. Otherwise consider the interval with minimum left end coordinate, place it to a new track, denote its right end coordinate by **X**; and delete the interval from **L**.

Step 2 Consider those intervals whose left end coordinate is greater than **X**. If there are none, go to step **1**. Otherwise choose the one with minimum

left end coordinate, place it to the actual track, denote its right end coordinate by **X**, delete this interval from **L** and go to step **2**.

Based on Tibor Gallai's research in the fifties [7], this algorithm gives the following classical result:

Theorem

- Every single row routing problem can be solved in linear time in the 2-layer Manhattan model.
- The resulting width equals the density, hence it is the best possible.
- The routing is dogleg-free, and the width could not be reduced if doglegs were permitted.

If we return to the general channel routing problem, the lower bound $w \geq d$ clearly remains valid in the 2-layer model, and in general, $w \geq \lceil d/l_H \rceil$ is valid for every solution, where d is the density and l_H is the number of horizontal layers.

But the statements of Theorem do not remain true. Let us show some examples.

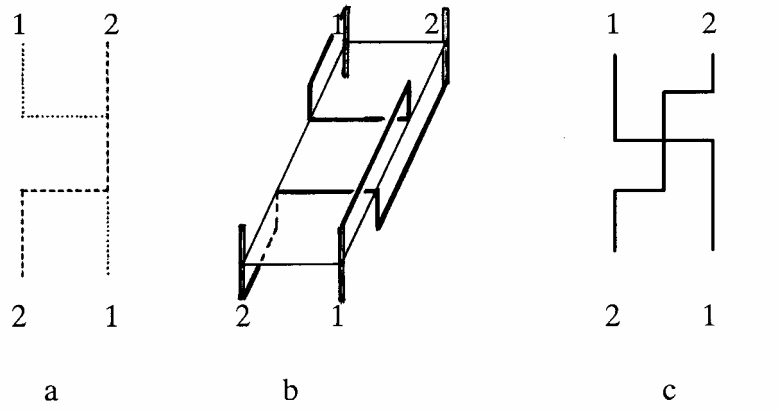


Figure 3

1. 3a is unsolvable in the 2-layer dogleg-free Manhattan model, but it has a solution in VHV model (see 3b), and it is solvable in the 2-layer Manhattan model, if doglegs are permitted (see 3c).

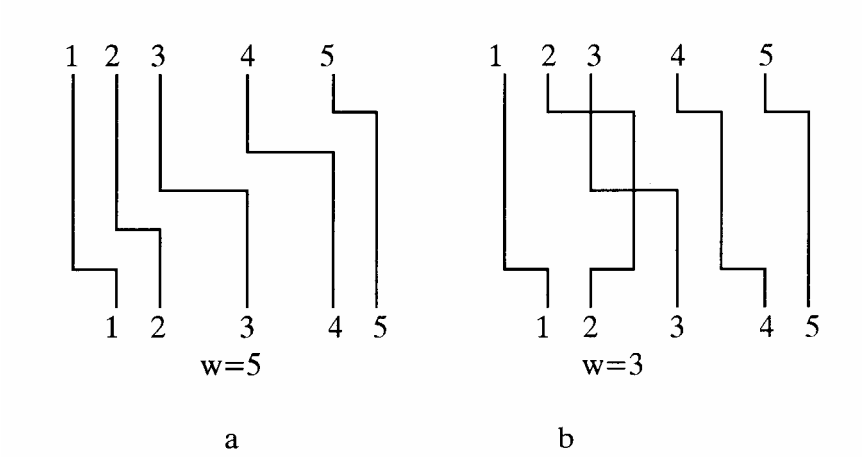


Figure 4

2. We have a solution in Figure 4a with width 5, and the same channel-routing problem has a solution with width 3, if doglegs are permitted (see 4b).

It is interesting to consider what can we say in general about channel-routing in the dogleg-free multilayer Manhattan model.

1. Every channel routing problem can be solved in linear time in the (VHVH....V) type Manhattan model, and the resulting width is $\lfloor d/l_H \rfloor$, hence it is the best possible. The proof in the VHV case comes from the above theorem [3] and the generalization for more layers is obvious [1], [2], [6].
2. In case $l_V=l_H=k$ it is NP complete to decide whether a channel routing problem can be solved with width $\lceil d/k \rceil$ in the dogleg-free 2k-layer Manhattan model. (The case $k=1$ has been well known for many years [8], [12], the more recent result [11] refers to the case $k>1$.)

Now we are going to study the remaining case.

3 The Case $l_V=1$ and $l_H=2$

In this case we have a trivial lower bound, but only some trivial upper bounds can be formulated. In fact, the dogleg-free realizability of a channel routing problem in this model is equivalent to that in the 2-layer Manhattan model and actual width requirements in then two models are closely related, as shown by the following observations [11]:

Lemma 1

If a channel routing problem can be solved with width **w** in the dogleg-free way in the HVH model – using one vertical and two horizontal layers – , then it has a dogleg-free solution with width at most **2w** in the „usual" Manhattan model (using one vertical and one horizontal layer).

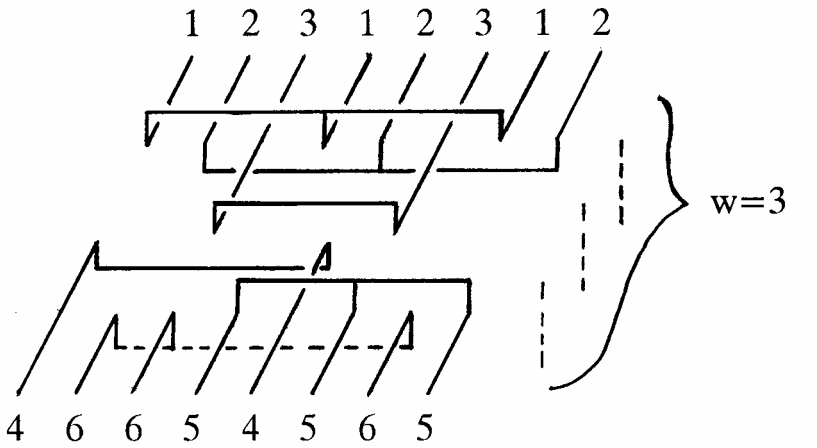


Figure 5a

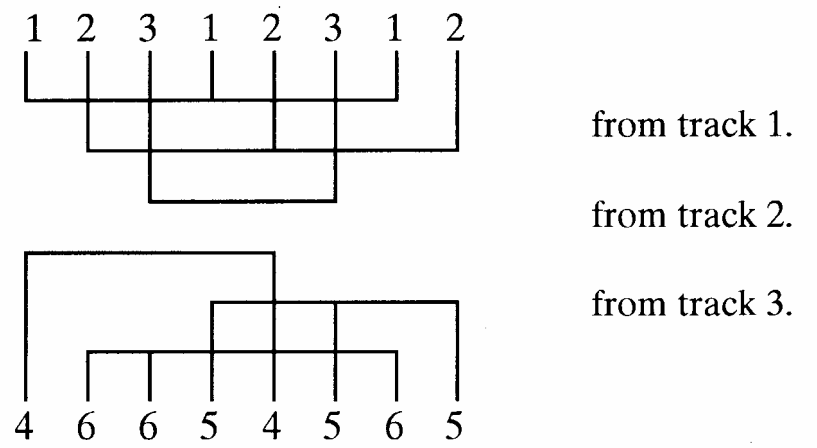


Figure 5b

Proof: Let us consider Figure 5a. This is a channel routing with length **8**, with width **3** in HVH Manhattan model. Let us proceed from North to South and realize the lines of the two horizontal layers in an alternating way (see Figure 5b) leading to two tracks of the two layer routing from each track of the original HVH routing. No conflict can arise, since we have a single vertical layer anyhow. This procedure transforms any HVH model into a „usual" 2-layer Manhattan solution [4].

Lemma 2

If a channel routing problem can be solved with width **w** in the dogleg-free 2-layer Manhattan model, it has a dogleg-free solution in the HVH model with width **w'**, where **w'** satisfies $\mathbf{w/2 \leq w' \leq w}$.

The proof is obvious.

It is very important to see, however, that the transformation from HVH to „usual" Manhattan model cannot always be inverted. If a channel routing problem has a solution with width **w**, it does not always have a solution in HVH model with width **w/2**.

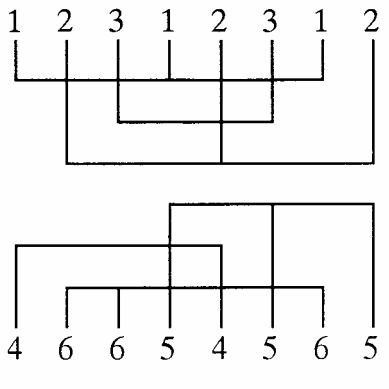


Figure 6a

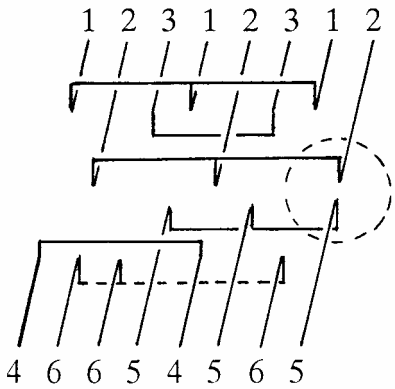


Figure 6b

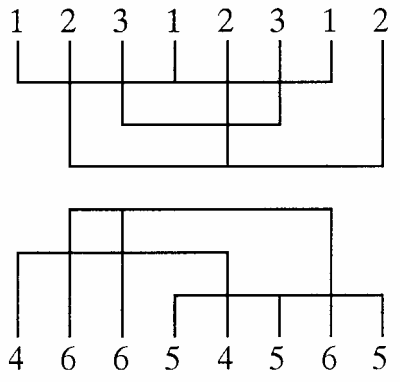


Figure 6c

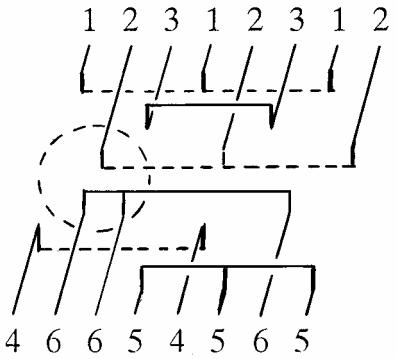


Figure 6d

For example, Figure 6a and 6c show two further solutions of the same channel routing problem as in Figure 5b, also with width **6**. However these solutions cannot be „folded back" like in Figure 5a (see 6b and 6d). More precisely, these solutions can be inverted to the HVH solution only with width **4** and not with width **3**. (In fact, one can enumerate that the actual problem can be routed in 40 different ways with width **6** in the usual Manhattan model and only 8 of them can be transformed back to the HVH model with width **3)**.

Let a dogleg-free channel-routing with width **w** in the usual Manhattan model and **w'** satisfies $\mathbf{w/2 \leq w' \leq w}$. The problem of transforming the routing into a HVH routing with width **w'** (if it is possible at all) can be formulated as a 2-processor job scheduling problem [5].

References

[1] Brady, M. L. and D. J. Brown, 1984. Optimal multilayer channel routing with overlap, *in*: C. E. Leierson, ed. *Advanced Research in VLSI Theory,* MIT Press, Cambridge, MA 281-296.

[2] Brady, M. L. and D. J. Brown, 1991. Optimal multilayer channel routing with overlap, Algorithmica *6*, 83-101.

[3] Chan, Wan S., 1983. A new channel routing algorithm, *in*:.Bryant, ed. *3rd Caltech Conference on VLSI*, Comp.Sci. Press, Rockville, 117-139.

[4] Chen, H. and M. Liu, 1984. Three-layer channel routing, *IEEE Trans. Computer-Aided Design of Integrated Circ. Syst.* CAD-3, 136-163.

[5] Cong, J., D. F. Wong and C. L. Liu, 1987. A new approach to the three layer channel routing problem, *ICCAD-87 (IEEE Internat. Conf. on Computer-Aided Design)* 378-381.

[6] Enbody, R. J. and H. C. Du, 1986. Near-optimal n-layer channel routing, *Proc. 23rd ACM-IEEE Design Automation Conf.* 708-714.

[7] Gallai T. His unpublished results have been announced in A. Hajnal and J. Surányi, 1958. Über die Auflösung von Graphen in vollständige Teilgraphen, *Ann. Univ. Sci. Budapest Eötvös Sect. Math.*1, 115-123.

[8] La Paugh, A. S. 1980. A polinomial time algorithm for optimal routing around a rectangle, *Proc. 21st FOCS Symp.* 282-293.

[9] Recski A. 1989. *Matroid Theory and its Applications in Electric Network Theory and in Statics*, Springer, Berlin and Akadémiai Kiadó, Budapest

[10] Recski A. 1992. Minimax results and polynomial algorithms in VLSI routing, *in*: M. Fiedler and J. Nesetril eds. *Combinatorics, Graphs, Complexity*, Ann. Discrete Math, 261-273.

[11] Recski A. 1997. Channel routing in the dogleg-free multilayer Manhattan model, *Proc. ECCTD' 97, Budapest,* I. 39-43.

[12] Szymanski, T. G. 1985. Dogleg channel routing is NP-complete, *IEEE Trans. Computer-Aided Design of Integrated Circ. Syst. CAD-4*, 31-41.

Appropriate Mathematical Model of DC Servo Motors Applied in SCARA Robots

Attila L. Bencsik

Bánki Donát Faculty of Mechanical Engineering, Budapest Tech
Népszínház utca 8, H-1081 Budapest, Hungary
bencsik.attila@bgk.bmf.hu

Abstract: In the first part of the presentation detailed description of the modular technical system built up of electric components and end-effectors is given. Each of these components was developed at different industrial companies separately. The particular mechatronic unit under consideration was constructed by the use of the appropriate mathematical model of these units. The aim of this presentation is to publish the results achieved by the use of a mathematical modeling technique invented and applied in the development of different mechatronic units as drives and actuators. The unified model describing the whole system was developed with the integration of the models valid to the particular components. In the phase of testing the models a program approximating typical realistic situations in terms of work-loads and physical state of the system during operation was developed and applied. The main innovation here presented consists in integrating the conclusions of professional experiences the developers gained during their former R&D activity in different professional environments. The control system is constructed on the basis of classical methods, therefore the results of the model investigations can immediately be utilized by the developer of the whole complex system, which for instance may be an industrial robot.

Keywords: mechatronic, system investigation, industrial robot, DC servo motor, PID control

1 Model of the Servo Motor

The first step was to design the model of a PM motor in order to perform the analysis. The system model has two main parts and several smaller units.

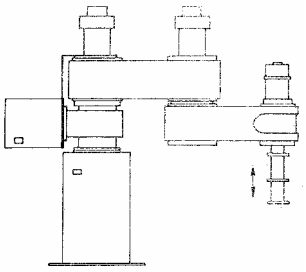


Figure 1

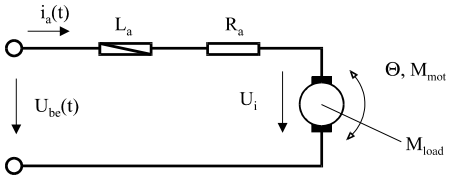


Figure 2

As a following step the control circuit was developed using the previous results.

$$U_{be}(t) = U_{R_a} + U_{L_a} + U_i \tag{1}$$

$$U_{be}(t) = i_a R_a + L_a \frac{di_a}{dt} + K_1 \tag{2}$$

$$U_i = K_i \Omega \tag{3}$$

$$M_{mot} - M''_{load} = M_{accelerator} = \Theta \frac{d\Omega}{dt} \tag{4}$$

$$K_2 la - M_{load} - B_V \Omega = \Theta \frac{d\Omega}{dt} \tag{5}$$

Loop of the circle:

$$U_{be}(t) = i_a - R_a + L_a \frac{di_a}{dt} + K_1 \Omega \tag{6}$$

$$K_2 i_a = \Theta \frac{d\Omega}{dt} + M_{load} + B_V \Omega \tag{7}$$

The system-model:

$$L_a \frac{d\, m_a}{dt} = U_{be}(t) - i_a R_a - K_1 \Omega \tag{8}$$

$$\Theta \frac{d\Omega}{dt} = K_2 i_a - M_{load} + B_V \Omega \tag{9}$$

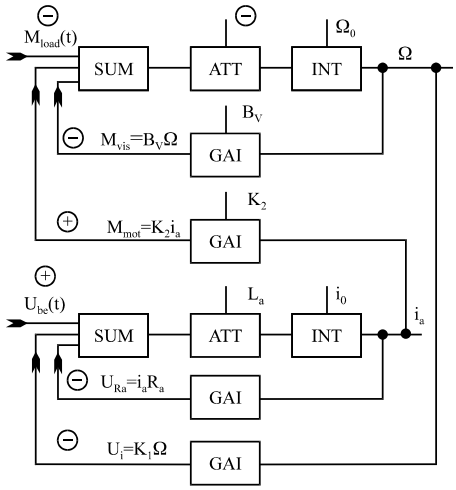
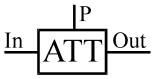


Figure 3

Interpretation of the blocks in system description

$$Out = In_1 + In_2$$



$$Out = \frac{In}{P}$$

$$Out = In \cdot P$$

$$Out = \int In \cdot dt$$

Motor given by the system-model can be regarded as only unit

Parameters of the motor used in simulation 80W

$R_a = 0{,}360$ [W] $L_a = 0.14 \cdot 10^{-3}$ [H]

$Q_{rotor} = 1{,}22 \cdot 10^{-4}$ [kg·m^2]

$K_1 = 50{,}1 \cdot 10^{-3}$ [Vs]

$M_n = 0{,}301$ [Nm]

$K_2 = 50{,}1 \cdot 10^{-3}$ [Nm/A]

$U_{be} = U_n = 15$ V

B_v clamping coefficient (viscosity) is not given in the cataloge. Let take the value of B_v so that in slow running, no-load current flows in the motor, $I_0 = 324$ mA.

$W_0 = 286{,}78$ rad/s. If $I_0 = 310$ mA then $M_{vis} = 0{,}05 \cdot M_n$! Then $B_v = (0{,}05.M_n)/W_0 = 5{,}23 \cdot 10^{-5}$ [Nms]

Time constants of the motor: $T_{el} = \frac{L_a}{R_a} = 3{,}8 \cdot 10^{-4}$ s

$$T_{emech} = \frac{R_a \cdot \Theta}{K^2} = 1{,}7 \cdot 10^2 \text{ s}$$

For evaluation of the results presented in diagram.

Stepping the curves (leaving them as the results):

On the left side of the diagram is the sclaling. (Ω, α, $I_{current}$, U_{switch}, $I_{current}$) The first number indicates the value belonging to the lower level of the rectangle which is divided into parts by sections, the second number indicates the changing value belonging to the upper level.

(In the case of those variables where the absolute value of these numbers is the same, sing is differnt, the **0** axis can be foumd on the centre line of the rectangle.)

Time on the horizontal axis can be seen on each diagram in [s].

Voltage signals can be found in the lower field of the diagram. Current signals situated symmetrically compared to the centre line (revolutions are the same exept running-up curves).

The basic signal for the position can also be seen in the circles for the regulation of the position. The torque signal (when it is not nul) there next to the diagram can be found in a small coordinate-system.

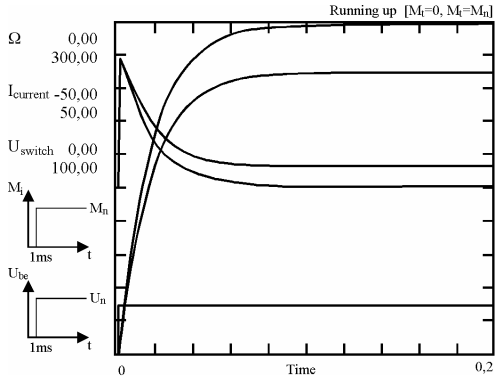


Figure 4

a) Load M_t=0; t=200 ms; I_{ac}=0,3105 A; U_{be}=15 V; W_0 = 297,1 rad/s

b) Load M_t=M_n=0,3 Nm; t=200 ms; I_a=6,253 A; U_{be}=15 V; W_n=254,46 rad/s

2 The Rotation Control Circuit

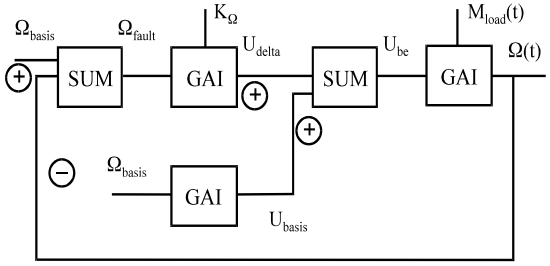


Figure 5

$$\Omega_{fault}=\Omega_{basis}-\Omega(t) \tag{10}$$

$$U_{delta}=K_{\Omega}\cdot\Omega_{fault} \tag{11}$$

As in the steady state oparation (M_t=at constant)

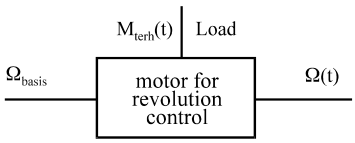
$$U_{be}/basis=U_i/\Omega_{basis}+I_a\cdot R_a/\Omega_{basis} \tag{12}$$

and $U_i>>I_a\cdot R_a$ inner voltage descrease is expedient, if

$$U_{basis}=K_1\cdot\Omega_{basis}=U_i/\Omega_{basis} \tag{13}$$

K-propertional coefficient of amplification influences the accuracy of control when K_{Ω} is increased the Ω_{fault} converges to 0.

Hereinafter the circle for revolutions control can be considered as an only unit.



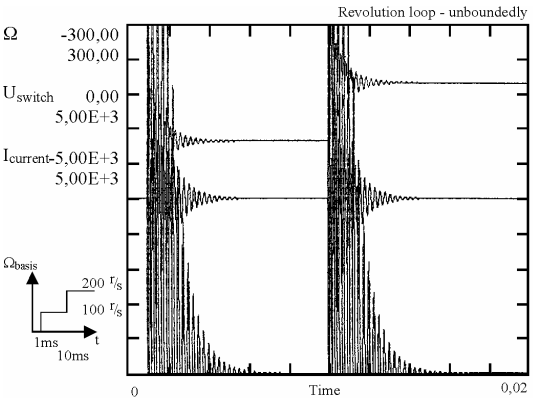


Figure 6

M_t=0; K_Ω=200; t=9.99 ms; Ω=100.001 r/s; t=20 ms Ω=200.00 r/s

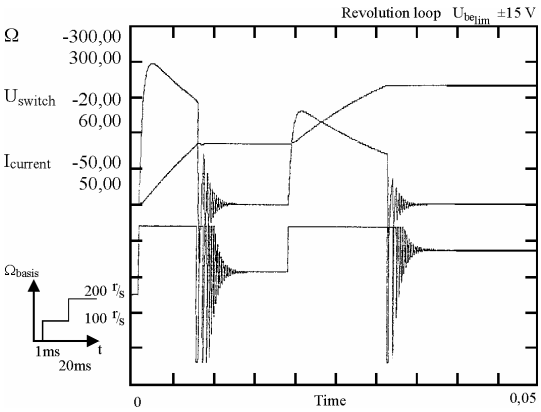


Figure 7

M_0=0; $U_{be_{lim}}$ ±15 V; K_Ω=200; t=50 ms; Ω=200.00 r/s

In Fig can be seen that following the unit jump the system wants to intervene faster and in the result of it more kV (i.e. much more current) is taken up. With real supply unit it is impossible to achieve. For rated voltage with unbounded supply voltage.

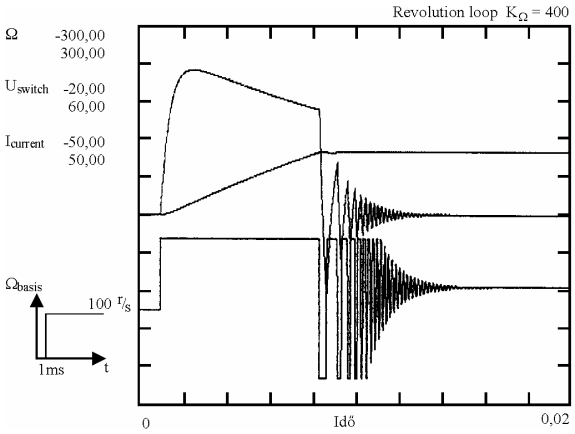


Figure 8

M_t=0; $U_{be_{lim}}$ = ±15 V; t=20 ms; Ω = 99.999 r/s; I_a=104.6 mA; U_{be}=5.047 V; K_Ω = 400

Under the same conditions in what has gone before K_Ω=1000. Non-damping vibrations set in. By increasing the proportional coefficient of amplification the accuracy of regulation can be increased, but problems of stability can arise.

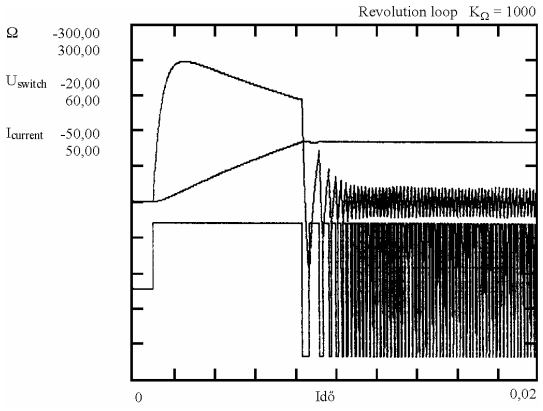


Figure 9

Effect of drive inertia. Reduction on motor-axis.

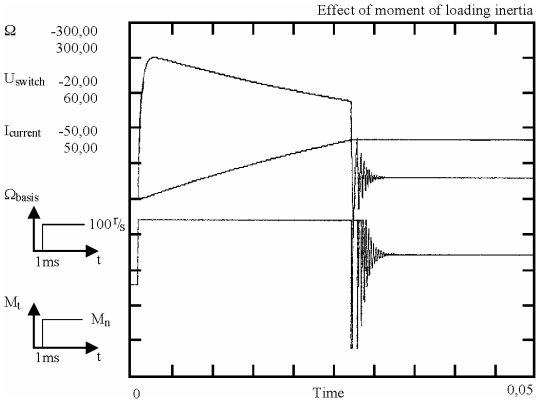


Figure 10

at t=50 ms; U_{be}=7.209 V; I_a=6.092 A; Ω=99.994 r/s; $\Theta=3\Theta_{rotor}$

On the basis of capacity:

$$M_r \cdot n_m = M_{load} \cdot n_{load} \cdot \frac{1}{\eta} \tag{14}$$

η - transmission efficiency

$$M_r = M_{load} \cdot \frac{n_{load}}{n_{motor} \cdot \eta} = M_{load} \cdot \frac{1}{a \cdot \eta} \tag{15}$$

Kinetic energy:

$$\frac{1}{2}\Theta_{load} \cdot \Omega_{load}^2 = \frac{1}{2}\Theta_r \cdot \Omega_m^2 \cdot \eta \tag{16}$$

$$\Theta_r = \Theta_{load} \cdot \frac{1}{\eta \cdot a^2} \tag{17}$$

In accordance with the relationship follows that in the case of reducers inertia of the driven arm has less influence. If the reduced inertia can be compared with the rotor inertia as it seen in Fig 6, regulation slows down (T increases) and the current load of the armature considerably increases.

Building up the regulating circle for the position.

3 The Position Control

Feedback of the proportional regulating loop for the position is performed by potentiometer.

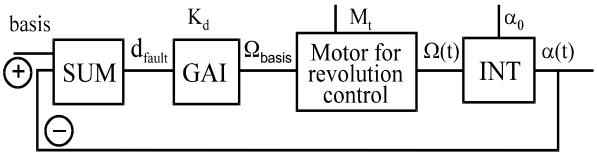


Figure 11

K_α proportional coefficient of amplification influences accuracy and stability of regulation. (Fig. 7, 8, 9)

As it seen from the diagrams that the proportional coefficient of amplification is sensitive to the scale of amplitude in the base signal. The inertia of the driven arm consideratly impackt on the quality of control, too.

The Whole circuit design Was concluded by the solution of the position control.

Funktion between α_{fault} (s) and Ω_{basis} (s):

$$K \cdot \left(\frac{1}{s} T_i + \frac{1 + s \cdot T_d}{1 + a \cdot s \cdot T_d} \right) \tag{18}$$

where T_i integration
T_d differential time constant
K amplification factor
at a=1 or T_d=0 PI regulation can be achieved

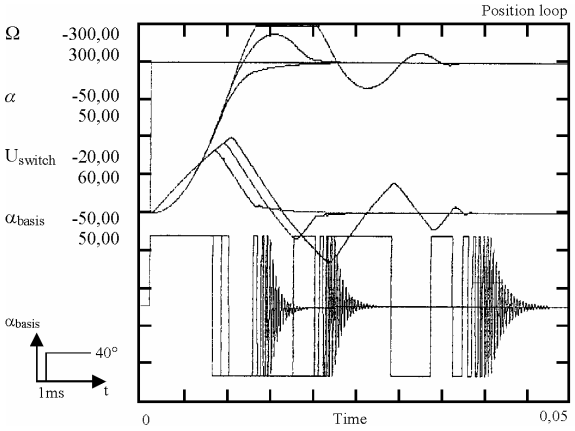


Figure 12

M_t =0 influence of the proportional coefficient of amplification K_α=300 aperiodic K_α=500 aperiodic with swing-off K_α=1000 periodic, with damped vibration.

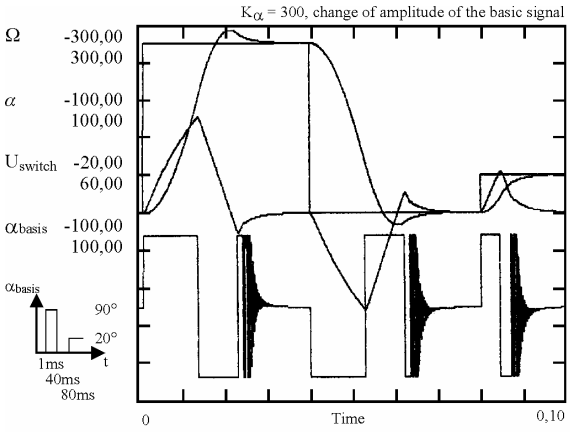


Figure 13

M_t=0 at t=100 ms; α=19.899; U_{be}=-0.009 V; I_a=-0.386 A

K_α proportional factor of amplification should be the function of the basic signal amplitude. K_α=300 set in to the 40° junp at 90° jump results swing-off, by decreasing of it the accuracy depreciates.

$\Theta = \Theta_{rotor}$ aperiodic swing-off free set in of the rotor.

$\Theta = 3\Theta_{rotor}$ aperiodic set in of the rotor with swing-off

at t=50 ms; U_{be}=0.028V; α=40.008°; I_a=0.095 A; Ω = -0.043 r/s

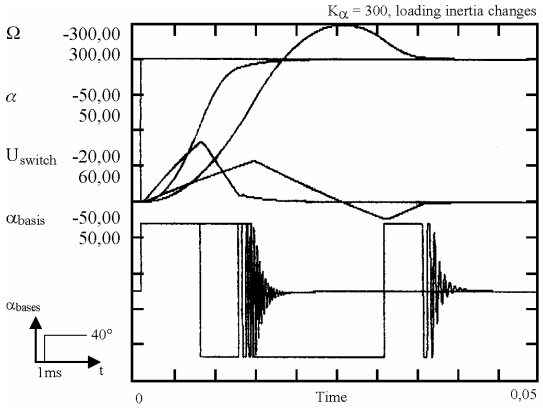


Figure 14

If the inertia of the arm can be compared with the inertia of the rotor - so in that case it has a great influence on the position regulating circle, too. The quality of range changes under the same contitions and so does the range time.

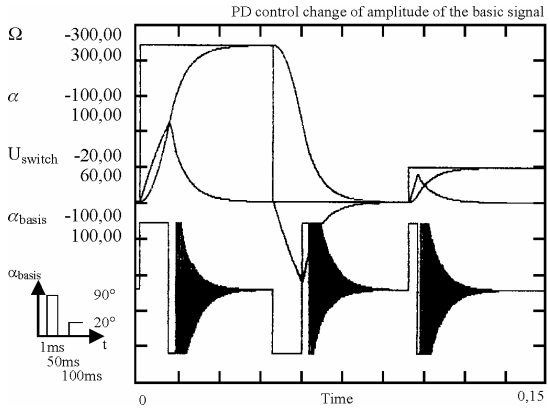


Figure 15

Parameters: K=100; $T_d=10^{-3}$; a=0.05; $M_t=0$; t=150 ms; I_a=-0.006 A; U_{be} = 0.0028V; $\alpha = 19.998°$; $\Omega = 0.0018$ r/s

$\Theta = \Theta_{rotor}$ her initial run-up takes place faster

$\Theta = 3\Theta_{rotor}$, then run-up is slower but the set-in (has the) same quality.

$\Theta = 3\Theta_{rotor}$ at t=50 ms; U_{be}=-0.025V; α=39.976°; Ω=0.068 r/s

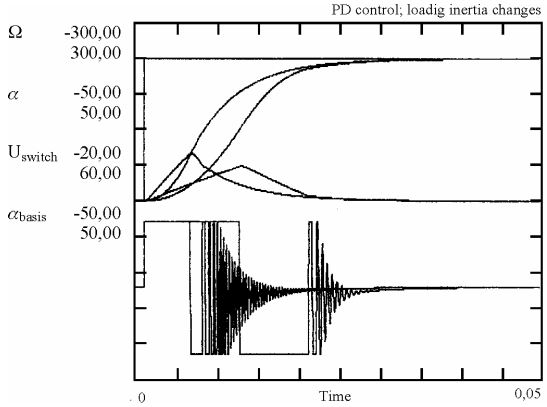


Figure 16

Fig. 16 repetation of simulations Fig 14. with PD control

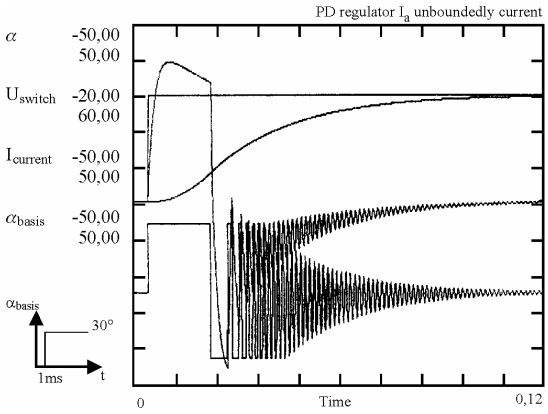


Figure 17

As it seen that in most part of reset time great amount of current falls to the armature. At starting without current limit, cca 39 A peak current appears. Amplification factor of PID member is: K=200

M_t=0 Amplification factor as the PID member should be decreased by K=80 for stabilitys sake. Setting time increases signifcantly.

In the case of real load control into the positions station can be very quick, so in every starting much current load to be reckoned wicht.

The motor in unable to bear in a long run. Current belonging to its nominal moment is 6,1 A but at starting it is 5-6 times more.

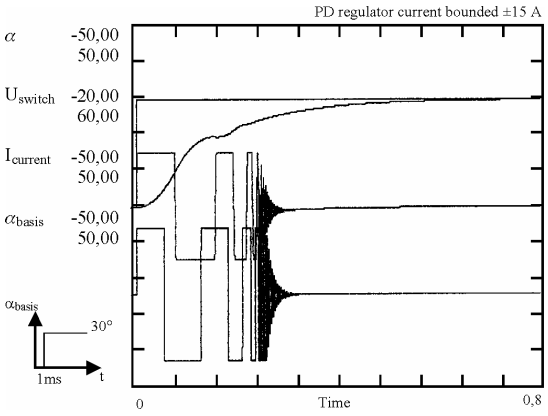


Figure 18

Conclusions

In accordance with these principles the presentation is rich in illustrations selected from the ample set of particular investigations. Though the limited size of this paper does not make it possible to give the full description of the verification of

the models applied, it is worthy of note, that via completing wide-spread series of simulations and measurements the appropriate models have been verified.

The main aim of the here published development was to bring about an open motion control system for a robot used for educational purposes.

References

[1] I. J. Rudas, L. Horváth: Fault Detection in Industrial Robots Using a Learning Control Based Approach. *Proc. of the 32nd SICE Annual Conference. Kanazawa,* Japan, 1993.

[2] Dr. Attila Bencsik-György Suszter: Investigation of Robot Arm Ridigity *Mechatronics* Vol. 3. No. 2. pp. 185-203 1993.

[3] A. Bencsik, I. J. Rudas: Simulation Tool for Interactive Design of Force Reflecting Master Arms. *Bulletins for Applied Mathematics* 947/94. The 25th Year's Jubilee Meeting in London, United Kingdom. pp. 99-105.

[4] Givon C. K. Yan, Clarence W.de Silva and George X. Wang Simulation and experiments on Intelligent control of a Wooddryng Kiln. *IFAC Symposion on Artifical Intelligence in Real Time Control (AIRTC-2000) Budapest, Hungary, Preprints pp.75-79.*

[5] Ales Hace, Karel Jezernik :Robust Position Tracking Control for Direct Drive Robot *Proceedings of the International Conference on Itelligent Engineering Systems (INES 2000),* Portoroz, Slovenija, pp. 121-124.

Semantical Equivalence of Process Functional and Imperative Programs

Ján Kollár

Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Letná 9, 042 00 Košice, Slovakia
Jan.Kollar@tuke.sk

Valerie Novitzká

Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Letná 9, 042 00 Košice, Slovakia
Valerie.Novitzka@tuke.sk

Abstract. Source-to-source transformations play crucial role in weaving multiple aspects of computation in aspect languages. Except that expressing imperative programs in the uniform form of expressions simplifies these transformations, this form is useful from the viewpoint of recognizing different aspects of computation at any level of program structure. In this paper we present the relation between imperative language and PFL – a process functional language, which manipulate environment variables in a side-effect manner, still preserving a purely functional principle based on evaluating expressions. Using an example of an imperative structured program, we will show the semantical equivalence of process functional and imperative programs. As a result, fine grained PFL form for picking out potential join points in imperative programs is obtained.

Keywords. Programming paradigms, programming languages, side effects, process functional language, aspect oriented programming.

1 Introduction

Purely functional programs [4] support equational reasoning – the program synthesis and the proof of program correctness [4, 16, 17]. However, the complex systems are not functional [20]. They are executed using I/O, exceptions, interrupt

handling, channel communication, etc. depending on the application to which they are proposed. The stateful computation/algorithm is such in which the state is significant. Since the state plays significant role in systems, an imperative language seems to be the best alternative for describing their functionality. On the other hand, using functional programming paradigm, a program is more tightly bound to the use of mathematical methods and more appropriate to transformations needed when weaving multiple aspects in aspect languages [2,3,7,15,23]. Let us introduce three approaches used in functional languages, able to express stateful computation. Using a functional language, the mechanism for updating a set of memory cells is required.

In Standard ML [5], the variable environment is used. For example, *v* is a memory cell in SML definition **val** *v* = **ref** 5. The value of variable *v* is accessed using operation ! : *a* **ref** $\rightarrow$ *a* in the form !*v*. The assignment *v* := !*v* + 1 increments the value of cell *v* by one. In SML, assignments are expressions of unit type and they may be used elsewhere in expressions by a programmer, i.e. explicitly as it is in an imperative language.

A pure, lazy functional language Clean [1] uses linear types again to perform the stateful computation, like Haskell. The asterisk in a type **World* designates that the type World is linear type. Since each function (process) may be of the type **World* $\rightarrow$ **World*, no single abstraction of monad is necessary to perform stateful computation.

No assignments are available to a programmer in Glasgow Haskell [19] – a purely functional language. They are hidden in application of processes, called state transformers [20]. The single abstraction of monads – mutable abstract type [22] is used to update the values of linear types, i.e. such that are accessed by a single pointer.

A monad is a triple (*M, unitM, bindM*), where *M* is a linear type, and the operations *unitM* and *bindM* are of the type:

$$unitM :: a \rightarrow M\ a \qquad \text{and} \qquad bindM :: M\ a \rightarrow (a \rightarrow M\ b) \rightarrow M\ b$$

In contrast to SML, Haskell is not environment-based language. The disadvantage of monadic approach is (at least from the viewpoint of software engineering) that memory cells are invisible to a user.

Seemingly, we may decide either to hide variable environment not using assignments, or, making environment visible, we must use assignments explicitly. However, process functional paradigm is based on preventing assignments, at the same time making variable environment visible to a programmer. PFL – an experimental process functional language [8, 9, 10, 11, 12, 13, 14], which we have developed in the past, is closely related to process functional paradigm.

Syntactically, PFL is a reduced subset of Haskell language, extended in a uniform way to support object programming. Currently we have two generators from PFL -

a generator to Java and to Haskell languages. Originally, our aim was to develop a programming language, that is safe, in the sense that it manipulates the state by application of processes instead of assignments and, at the same time, it provides well-defined spatial information about memory data organization to a programmer.

Considering aspect oriented programming methodology [2,3,7,15,23], it seems that PFL may used as a general implementation bridge for any target imperative language, supporting this multi-aspect approach to programming. In this paper, we present the relation between imperative structures of a simple but representative imperative language and process functional expressions. Providing the translation scheme ***P***, and a simple imperative program expressing it in PFL, we prove informally the equivalence of process functional and imperative languages.

The essence of PFL is introduced in section 1. A representative subset of an imperative language (omitting procedures and functions) is defined in section 2. In section 3 we present the translation scheme ***P*** which maps imperative programs to PFL form. An example of the simple imperative program, as well as its equivalent form in PFL, is introduced in section 4.

Finally, in Conclusion, we summarize our results and briefly comment the directions of further research. We will use Bird‘s mathematical notation for PFL programs in this paper.

2 The Essence of PFL

In PFL, the assignments are performed by a process application implicitly [8]. The source form of a process definition (an equation designated by =) is seemingly purely functional. On the other hand, a variable environment is visible to a programmer in a process type definition (an equation designated by ::). Environment variables – memory cells – may be shared by two or more processes defined in the same scope. They are introduced as the attributes of the types of process arguments. There are no reference types in PFL. Each argument type (*v T*) introduces the variable *v* of the type *T* to the variable environment. Each function comprising at least one argument type in the form (*v T*) is a process. Except that, processes are functions defined in terms of unit types for arguments and/or values. Primitive and algebraic types are designated by identifiers starting with uppercase letters and the environment variables (memory cells) by identifiers starting with lowercase letters. It is just one exception from this rule, when the type *T* is a type variable. Then it is designated by single lowercase letter. For example (*a a*) means, that the first *a* is the environment variable and the second *a* is the type variable. The computational model for PFL based on control-driven data flow can be found in [9].

As an introductory example, let process *p* is defined as follows:

$$p :: a\ Int \rightarrow b\ Int \rightarrow Int$$

$$p\ x\ y = x + y$$

Then, the value of expression (*p* 2 3) is 5. Like the side effect of evaluation, the values of cells *a* and *b* (environment variables) will be $a = 2$ and $b = 3$.

Now, suppose that the values of *a*, *b* are $a = 2$ and $b = 3$.

Hence, the value of (*p* ()()) is 5 again, and the state of *a* and *b* ($a = 2$ and $b = 3$) remains unchanged. The arguments () are control (unit) values of unit types ().

Provided that an argument is of process type (*v T*), then applying the process on a control value (or an expression that yields control value), the process value is evaluated using the current value having been stored in *v* before the process is applied. Control values do not affect the function of computation directly, nevertheless, the order, in which the expressions of unit types are evaluated, affects the state. Notice, that the processes (in contrast to functions) are evaluated eagerly, following the principle of causality: *The value of a process is evaluated after the arguments are evaluated*. The order, in which the arguments are evaluated, may affect the function of computation. That is why source process definitions are purely functional just seemingly. As an example, let us define the process *q* as follows:

$$q :: a\ Int \rightarrow a\ Int \rightarrow Int$$

$$q\ x\ y = x + y$$

Evaluating (*q* 2 3), the result is 5, but the value assigned to variable *a* is either 2 or 3, depending on whether *q* is applied first to 3 or to 2. If the arguments (that may be complex expressions not just simple constants, such 2 and 3 above) are evaluated in parallel, the state change is non-deterministic. Clearly, the definition of process *q* is purely functional just if we omit its type definition (marked by ::), otherwise not. On the other hand, possible nontransparency is evidently separated from the definition itself and it is shifted to the type definition.

Let process *r* is defined as follows:

$$r :: a\ Int \rightarrow ()$$

$$r\ x = ()$$

The application (*r* (4 + 5)) evaluates the argument 9, which is assigned to *a*. The result is a control value () which does not allocate the stack at all. It may be noticed that PFL expression application (*r* (4 + 5)) corresponds to the assignment $a := 4 + 5$ in an imperative language.

In the last introductory example, let process *s* is defined as follows:

$s :: () \rightarrow ()$

$s\ () = ()$

Process *s* can be applied to each expression of unit type, yielding control value, for example, such as (*s* (*r* (4+5))). Side effect is the same as for (*r* (4+5)), which yields control value (). This value is used as the argument of *s*. In PFL, the conception of data and control values is well balanced, not hiding them to a programmer. We will suppose evaluating PFL expressions in leftmost innermost order, to guarantee correct semantics for imperative control structures – statements of a sequential imperative language.

3 An Imperative Language

We will start with the syntactic domain of a simple imperative language, in which the program *pr* comprises variable declarations *vd* and block *bl* consisting of statement sequence, according to the Fig.1. A statement *st* may be assignment *as*, if statement *if* , or while statement *wh*.

pr ::= *vd bl*
vd ::= **var** $v_1 : T_1 ; \ldots v_n : T_n;$
bl ::= **begin** $st_1 ; \ldots ; st_m$ **end**
st ::= *as* | *if* | *wh*
as ::= *v* := *e*
if ::= **if** *e* **then** bl_T **else** bl_F
| **if** *e* **then** bl_T
wh ::= **while** *e* **do** *bl*

Fig. 1: An imperative language

The detailed syntax of expression *e* is not substantial for our purposes. In Fig.1, T_k are types, *n* is the number of variables ($n \geq 1$), *m* is the number of statements ($m \geq 1$) of a block, and *e* is an expression. Instead of extended BNF form $(st\ (; st)^*)$ we rather express non-empty statement sequence by $st_1 ; \ldots ; st_m$ ($m \geq 1$). Block bl_T of if statement is executed when the value of boolean expression *e* is true, and block bl_F is executed when *e* is false.

4 Expressing Imperative Programs in PFL

In this section we will present the scheme ***P*** for translation of imperative language in Fig. 1. For the purpose of simplicity, we will consider just variables declared in variable declarations *vd* are used in block *bl*.

Program *pr* is equivalent to PFL expression ***P*** $[\![\ vd\ bl\ ]\!]$, such that:

$$\boldsymbol{P}\ [\![\ vd\ bl\ ]\!]\ = \boldsymbol{P}\ [\![\ bl\ ]\!]\ \mathbf{A}$$

where $\mathbf{A} = \{ v_1 : T_1 ; \ldots v_n : T_n \}$ is the set of associations defined in variable declarations *vd*

such that v_k are used in *bl*.

For example, the set of associations for imperative program in Fig. 2 is as follows:

$$\mathbf{A} = \{x : Int, y : Int, s : Int\}$$

Expression *e* in an imperative language is equivalent to PFL expression $\boldsymbol{P}\ [\![\ e\ ]\!]\ \mathbf{A}_e$, as follows:

$$\boldsymbol{P}\ [\![\ e\ ]\!]\ \mathbf{A}_e = ep\ ()_1 \ldots ()_r$$

provided that $v_1, \ldots, v_r$ are (imperative) variables used in *e*, $\mathbf{A}_e = \{v_1 : T_1, \ldots, v_r : T_r\}$, such that $\mathbf{A}_e \subset \mathbf{A}$, and process *ep* is a new process defined as follows:

$$ep :: v_1\ T_1 \rightarrow \ldots \rightarrow v_r\ T_r \rightarrow T$$

$$ep\ x_1 \ldots x_r = e[x_1/v_1, \ldots, x_r/v_r]$$

where *T* is a type of expression *e*, and its form $e[x_1/v_1, \ldots, x_r/v_r]$ means that in this expression are variables v_k substituted by lambda variables x_k.

For example, let us consider the expression $s + x$ on the right hand side of assignment

$s := s + x$. This expression is translated using associations $\mathbf{A}_e = \{x : Int, s : Int\}$ into the application of a new PFL process, say *sxp*, in the form *sxp* () (), where that *sxp* is defined as follows:

$$sxp :: s\ Int \rightarrow x\ Int \rightarrow Int$$

$$sxp\ p\ q = p + q$$

Block *bl* is equivalent to PFL expression $\boldsymbol{P}\ [\![\ bl\ ]\!]\ \mathbf{A}$, as follows:

$$\boldsymbol{P}\ [\![\ \textbf{begin}\ st_1 ; \ldots ; st_m\ \textbf{end}\ ]\!]\ \mathbf{A}\ = blp\ \boldsymbol{P}\ [\![\ st_1\ ]\!]\ \mathbf{A}_1 \ldots \boldsymbol{P}\ [\![\ st_m\ ]\!]\ \mathbf{A}_m$$

where the variables of associations $\mathbf{A}_i$ are used in statements st_i. It holds $\mathbf{A} = \mathbf{A}_1 \cup \ldots \cup \mathbf{A}_m$. The new process *blp* applied in PFL expression above, is defined as follows:

$$blp :: ()_1 \rightarrow \ldots \rightarrow ()_m \rightarrow ()$$

$$blp\ ()_1 \ldots ()_m = ()$$

For example, if top-level block contains three assignments, one while statement and one write statement, then corresponding PFL definition of process *blp* is as follows:

$blp :: ()\rightarrow()\rightarrow() \rightarrow()\rightarrow()\rightarrow()$

$blp\ ()\ ()\ ()\ ()\ ()\ = ()$

Statement *st* is equivalent to PFL expression $\boldsymbol{P}\ [\![\ st\ ]\!]\ \mathbf{A}$, which is equal to PFL expressions for assignment, if statement, or while statement.

Assignment *as* is equivalent to PFL expression $\boldsymbol{P}\ [\![\ as\ ]\!]\ \mathbf{A}$ as follows:

$$\boldsymbol{P}\ [\![\ v := e\ ]\!]\ \mathbf{A} = asp\ \boldsymbol{P}\ [\![\ e\ ]\!]\ \mathbf{A}_e$$

where the variables of associations $\mathbf{A}_e$ are used in expression e. It holds $\mathbf{A} = \{v : T\} \cup \mathbf{A}_e$. The new process *asp* is defined as follows:

$asp :: v\ T \rightarrow ()$

$asp\ x = ()$

and $\boldsymbol{P}\ [\![\ e\ ]\!]\ \mathbf{A}_e$ is a PFL expression.

If statement *if* is equivalent to PFL expression $\boldsymbol{P}\ [\![\ if\ ]\!]\ \mathbf{A}$, as follows:

$$\boldsymbol{P}\ [\![\ \textbf{if}\ e\ \textbf{then}\ bl_T\ \textbf{else}\ bl_F\ ]\!]\ \mathbf{A} = ifp$$

The new process *ifp* is defined as follows:

$ifp :: ()$

$ifp \quad | \boldsymbol{P}\ [\![\ e\ ]\!]\ \mathbf{A}_e = \boldsymbol{P}\ [\![\ bl_T\ ]\!]\ \mathbf{A}_T$

$\quad\quad | \textbf{otherwise} \quad = \boldsymbol{P}\ [\![\ bl_F\ ]\!]\ \mathbf{A}_F$

where $\mathbf{A} = \mathbf{A}_e \cup \mathbf{A}_T \cup \mathbf{A}_F$.

If statement without bl_F block is expressed as follows:

$$\boldsymbol{P}\ [\![\ \textbf{if}\ e\ \textbf{then}\ bl_T\ ]\!]\ \mathbf{A} = ifp$$

In this case the new process *ifp* is defined as follows:

$ifp :: ()$

$ifp \quad | \boldsymbol{P}\ [\![\ e\ ]\!]\ \mathbf{A}_e = \boldsymbol{P}\ [\![\ bl_T\ ]\!]\ \mathbf{A}_T$

$\quad\quad | \textbf{otherwise} \quad = ()$

where $\mathbf{A} = \mathbf{A}_e \cup \mathbf{A}_T$.

While statement *wh* is equivalent to PFL expression $\boldsymbol{P}\ [\![\ wh\ ]\!]\ \mathbf{A}$, as follows:

$$\boldsymbol{P}\ [\![\ \textbf{while}\ e\ \textbf{do}\ bl\ ]\!]\ \mathbf{A}_e = whp\ ()$$

New process *whp* is defined as follows:

$$whp :: ()\rightarrow()$$

$$whp\ () \quad | \boldsymbol{P}\ [\![\ e\ ]\!]\ \mathbf{A}_e \quad = whp\ \boldsymbol{P}\ [\![\ bl\ ]\!]\ \mathbf{A}_w$$

$$| \textbf{otherwise} \quad = ()$$

where $\mathbf{A} = \mathbf{A}_e \cup \mathbf{A}_w$.

PFL form for boolean expression *e* in if and while statements is the same as for general expresions *e* being shown above.

5 An Example

As an example, let us consider a simple imperative program in Fig. 2, which reads two integers x and y, and computes the sum *s* of absolute values in the range

(x . . . y).

It is supposed, that primitive function **read** (referentially non-transparent) and primitive **write**, of the types **read** :: *Int*, and **write** :: *Int* → (), are built-in.

```
var x, y, s : integer;
begin
  x := read; y := read; s := 0;
  while x <= y do begin
    if x > 0 then begin
      s := s + x
    end else begin
      s := s – x
    end;
    x := x + 1
  end;
  write(s)
end
```

Fig. 2: Imperative program

As we will see, except that the basic association is $\mathbf{A} = \{x : Int, y : Int, s : Int\}$, after expressing the program in PFL form, each process uses a subset of this association, which is defined by its type definition.

Since the values of variables *s*, *x* and *y* are used in expressions, the accessing processes *s*, *x* and *y* (for the purpose of simplicity we use for them the same names as for variables) are identities, defined in Fig. 3a.

$$s :: s\ Int \rightarrow Int$$
$$s\ p = p$$

```
x :: x Int → Int
x p = p

y :: y Int → Int
y p = p
```

Fig. 3a: Accessing processes

As we will see below, the weakness of imperative programs is that the argument of accessing processes is just unit value (), not a complex expression evaluated to unit value. The values in environment variables *s*, *x*, and *y* are accessed using applications *s*(), *x*(), and *y*(), respectively.

The top level block of a program in Fig. 2 consists of the sequence of three assignments, followed by two statements (while and write). So, we can integrate the transformation of block, sequence and assignment combining all of them by the definition of single process *bsa* – block-sequence-assignment compound process. Since different compound process expresses the body of while statement they are designated by different names: *bsaA* for the top level block, and *bsaB* for the while block in Fig. 3b. On the other hand, both blocks in if statement consist of assignment to the same variable *s*. Such blocks can be integrated using one process *bsaC* for both blocks, as shown in Fig. 3b.

```
bsaA :: x Int → y Int → s Int → () → () → ()
bsaA  p q r ()() = ()

bsaB ::()→ x Int → ()
bsaB  () p = ()

bsaC :: s Int → ()
bsaC  p  = ()
```

Fig. 3b: Compound processes

Now we are ready to built up the structure of our program in Fig. 2 in the whole, defining *while* process using the scheme for while statement, *if* process using the scheme for if statement and finally, main as a constant expression – the application of *bsaA*, yielding unit value, as can be seen in Fig. 3c.

```
while :: () → ()
while ()  | x()<= y()          = while ( bsaB  if   (x() + 1))
                  | otherwise          = ()

if  :: ()
if          | x() > 0  = bsaC (s() + x())
            | otherwise          = bsaC (s() – x())

main  :: ()
main = bsaA read read 0 (whp ()) (write (s()) )
```

Fig. 3c: The structure of the program

The PFL program in Figures 3a, 3b, and 3c including built-in **read** and **write**, is semantically equivalent to the imperative program in Fig. 2; the evaluation of *main* is the same as the execution of program in Fig. 2.

Conclusions

Using PFL, a program is expressed without assignments and statement sequences, preserving at the same time the visibility of environment variables – memory cells. An imperative computation is performed by the evaluation of an expression with side effects.

We have illustrated structured imperative style of programming, building PFL program in botton-up manner. It is not to argue that process functional programming is better than when an imperative language is used. But we can see the following facts. First, exploiting the application dependency in imperative languages is poor, if any. Omitting functions, lambda variables (designated by *p*, *q*, and *r* in our example in Fig. 3a, 3b, and 3c) are not used in expressions at all. Second, the use of variables in imperative languages is far less disciplined, as when they are associated with PFL processes in their type definitions and shared by process applications.

The arguments of processes in this paper are supposed to be evaluated sequentially and eagerly. Essentially, this is the simpliest way how to guarrantee the required degree of determinism in computation. On the other hand, the application dependency is other alternative, exploited using monads [22]. Although this is over the scope of this paper, monadic programming style is not excluded using process functional language. Using monadic style in PFL, in contrast to Haskell, memory cells remain still visible, as we have shown in [12].

Considering multi-paradigmatic approaches, such as combining logic and functional programming [18, 21] or object oriented and logic programming in aspect programming methodology, the aim of both is to increase the semantical power of the language, using less or more uniform language syntax. Especially in aspect programming, the underlying language such as Java in AspectJ [7] determines the transparency of programs, because Java constructs are used in pointcut designators. Except that, renaming a metod in an original module after adding aspect module may affect the semantics the program inappropriately.

As has been shown, PFL form of programs allows (and requires) to designate the structures and substructures of a program systematically. Then we may think about more fine grained aspects, as those when an imperative language is used. In the past, we have PFL-to-Java and PFL-to-Haskell generators developed. The subject of our current research is integrating aspect and process functional paradigm of programming. In this framework, with respect of complexity of software systems and the need to precede their behavior [6], we are interested especially in methods for replacing the „programming style“ (in which PFL implementation structures are named) by the „specification style“ in which they

are derived [16, 17] and affected by new static and dynamic aspects, in the way that run-time can be still monitored, corrected, profiled and optimized according to user requirements.

Acknowledgement

This work was supported by VEGA Grant No. 1/1065/04 – Specification and implementation of aspects in programming.

References

[1] P. Achten, R. Plasmeijer: Interactive Functional Objects in Clean. In: Clack et al. (Ed.): IFL'97, LNCS 1467, 1998, pp. 304–321.

[2] J. H. Andrews: Process-algebraic foundations of aspect-oriented programming, In Proceedings of the Third International Conference on Metalevel Architectures and Separation of Crosscutting Concerns (Reflection 2001), LNCS, Springer-Verlag, 2001, Vol. 2192, pp. 187–209.

[3] E. Avdicausevic, M. Lenic, M. Mernik, and V. Zumer: AspectCOOL: An experiment in design and implementation of aspect-oriented language. ACM SIGPLAN not., December 2001, Vol. 36, No.12, pp. 84–94.

[4] R. S. Bird: Algebraic Identities for Program Calculation, The Computer Journal, Vol.32, No.2, 1989, pp. 122–126

[5] R. Harper, D. MacQueen, and R. Milner: Standard ML. ECS-LFCS-86-2, LFCS Report Series, University of Edinburgh, Department of Computer Science, 1986, 35pp.

[6] Š. Hudák, S. Šimoňák: FDT Interfacing, Analele Universitatei din Oradea, Romania, pp. 53-59.

[7] G. Kiczales, E. Hilsdale, J. Hugunin, M. Kersen, J. Palm, and W. G. Griswold: An overview of AspectJ, In Proceedings European Conference on Object-Oriented Programming, LNCS, Springer-Verlag, Vol. 2072, 2001, pp. 327–353

[8] J. Kollár: Process Functional Programming, In: Proc. of Int. Conf. MOSIS' 99 Conference, Rožnov pod Radhoštěm, Czech Republic, April 27–29, 1999, pp. 41–48.

[9] J. Kollár: Control-driven Data Flow, Journal of Electrical Engineering, Vol. 51. No. 3-4, 2000, pp. 67–74

[10] J. Kollár: Comprehending Loops in a Process Functional Programming Language, Computers and AI, Vol. 19, 2000, pp. 373–388

[11] J. Kollár: Object Modelling using Process Functional Paradigm, Proc. 34th Spring International Conference MOSIS 2000 - ISM 2000 Information Systems Modelling , Rožnov pod Radhoštěm, Czech Republic, May 2–4, 2000, ACTA MOSIS No. 80, pp. 203–208

[12] J. Kollár: Partial Monadic Approach in Process Functional Language, Acta Electrotechnica et Informatica, No. 1, Vol. 3, 2003, TU Košice, Slovakia, pp. 36–42

[13] J. Kollár: Unified Approach to Environments in a Process Functional Language, Computing and Informatics, Vol 22, 2003, pp. 439-456

[14] J. Kollár, P. Václavík, and J. Porubän: The Classification of Programming Environments, Acta Universitatis Matthiae Belii, 2003, 10, 2003, pp. 51-64

[15] M. Mernik, M. Lenic, E. Avdicausevic, and V. Zumer: A reusable object-oriented approach to formal specification of programming languages, L'Objet, 1998, Vol. 4, No. 3, pp. 273-306

[16] V. Novitzká: Structures of Algebraic Specification Languages, Proc. of the Int. Conf. ECI'98, Košice–Herl'any, Slovakia, October 8–9, pp. 1–6.

[17] V. Novitzká: Systems for Deriving Correct Implementations. In:Proc. of Int. Conf. MOSIS'99 Conference, Rožnov pod Radhoštěm, Czech Republic, April 27–29, 1999, pp. 201–207.

[18] M. Paralič: Mobile Agents Based on Concurrent Constraint Programming, Joint Modular Languages Conference, JMLC 2000, September 15 6-8, 2000, Zurich, Switzerland. In: Lecture Notes in Computer Science, 1897, pp. 62–75.

[19] J. Peterson, K. Hammond (Ed.): Report on the Programming Language Haskell: A Non-strict, Purely Functional Language Version 1.3. Yale University, May 1996. 164pp.

[20] S. L. Peyton Jones, P. Wadler: Imperative functional programming. In 20th Annual Symposium on Principles of Programming Languages, Charleston, South Carolina, January 1993, pp.71–84.

[21] G. Smolka: The Oz programming model, In Jan van Leeuwen, editor, Computer Science Today, Lecture Notes in Computer Science 1000, Springer-Verlag, Berlin, 1995, pp. 324–343.

[22] P. Wadler: The essence of functional programming. In: 19th Annual Symposium on Principles of Programming Languages, Santa Fe, New Mexico, January 1992, pp. 47-70.

[23] M. Wand: A semantics for advice and dynamic join points in aspect-oriented programming. Lecture Notes in Computer Science, 2001, 2196:45-57.