Hardware Implementation of CMAC Type Neural Network on FPGA for Command Surface Approximation

Sándor Tihamér Brassai, László Bakó

Sapientia - Hungarian University of Transilvania, Faculty of Technical and Human Sciences, Târgu–Mureş, Romania, 547367 Corunca, Şos. Sighişoarei 1C

E-mail: tiha@ms.sapientia.ro, lbako@ms.sapientia.ro

Abstract: The hardware implementation of neural networks is a new step in the evolution and use of neural networks in practical applications. The CMAC cerebellar model articulation controller is intended especially for hardware implementation, and this type of network is used successfully in the areas of robotics and control, where the real time capabilities of the network are of particular importance. The implementation of neural networks on FPGA's has several benefits, with emphasis on parallelism and the real time capabilities.This paper discusses the hardware implementation of the CMAC type neural network, the architecture and parameters and the functional modules of the hardware implemented neuro-processor.

Keywords: Neural networks, CMAC, Neural networks hardware implementation, FPGA

1 Introduction

Great interest has been manifested lately for the utilization of adaptive modeling and control, based on biological structures and learning algorithms. Control systems need to have high dynamic performance and robust behavior. These controllers are expected to cope with complex [1], uncertain and nonlinear dynamic processes. It is difficult to obtain a mathematical representation of uncertain and nonlinear dynamic processes that impose an intelligent modeling and control. For static system modeling one can use feed-forward static networks like Multi-Layer Perceptrons (MLP), Radial Basis Function (RBF). For dynamic system modeling, neural networks that show temporal behaviour can be used.

The major disadvantage of MLPs and the MLP-based dynamic networks is their slow training algorithm. This drawback may be an obstacle to apply them for real-time adaptive modeling problems.

Using networks with only a single trainable layer, the learning speed can be significantly increased.

CMAC (Cerebellar Model Articulation Controller) and RBF (Radial Basis Function) are networks with a single trainable layer and have better capabilities than multi-layer perceptrons. The most important properties of the CMAC type controller are the fast learning capability and the special architecture that allows digital hardware implementation.

2 CMAC Network

Cerebellar Model Articulation Controller networks play an important role in non-linear function approximation and system modeling. The main advantages of CMAC type networks compared to MLP, are their extremely fast learning and the possibility of low-cost digital implementation.

The CMAC network can be represented as a three layer system (Figure 1) with a normalized input space, basis functions, and output, and can be considered as an associative memory, which realizes two subsequent mappings.

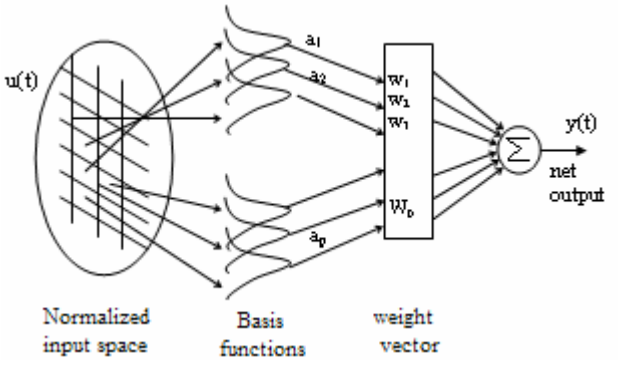

Figure 1
CMAC network as a three layer system

The first one - which is a non-linear mapping - projects an input space point **u** into a binary association vector **a.**

The association vectors always have C active elements, which means that C bits of an association vector have the value ‘1’ and the others have the value ‘0’. C is an important parameter of the CMAC network and it is much less than the length of the association vector (Figure 2).

In practical applications the two mappings are implemented by a two-layer network. The first layer is responsible for mapping the input points to the association vectors; this mapping is fixed (can be wired in hardware).

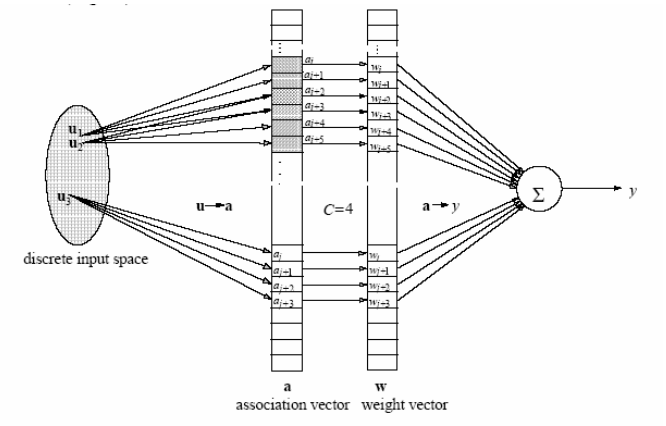

Figure 2
The mapping of a CMAC The trajectory tracking

The second layer is trainable and realizes a linear combination of the weight vector and the association basis function vector. The input variables are divided into overlapped regions and every region is subdivided into quantization intervals. The output value for an input point can be considered as a weighted sum of selected basis functions [2]. The resolution of the network and the shift positions of the overlapping regions are determined by this quantization.

A given value of an input variable activates all the regions where the input is within a quantization interval of a region.

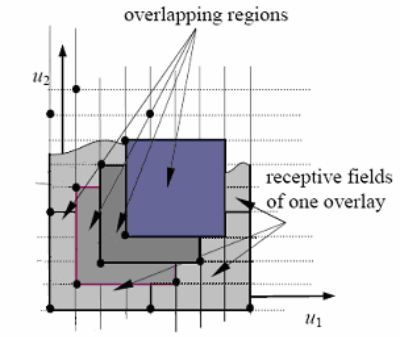

Figure 3
The receptive fields of a two-variable Albus CMAC

Every element in the association vector corresponds to a basis function. Each basis function has a so-called receptive field.

The shaded regions in Figure 3 are receptive fields of different basis functions [3]. If an input point is within a receptive field, the corresponding basis function is selected. The basis functions are grouped into overlays. One overlay contains basis functions with non-overlapping supports, but the union of the supports covers the whole input space. The different overlays have the same structure; they consist of similar basis functions in shifted positions. The positions of the overlays and the basis functions of one overlay can be represented by definite points.

In the original Albus scheme the overlay-representing points are in the main diagonal of the input space, while the basis function positions are represented by the sub-diagonal points as it is shown in Figure 3. The overlay representing points can be described as displacement vectors the elements of which are the coordinates of the definite points. Every input data will select C basis functions, each of them on a different overlay, so in an overlay one and only one basis function will be active for every input point. As every selected basis function will be multiplied by a weight value, the size of the weight memory is equal to the total number of basis functions or to the length of the association vector. As an element of the association vector can be considered as the value of a basis function for a given input, the output of the binary basis function is one if an input is in its receptive field and zero elsewhere:

$$a_i(u) = \begin{cases} 1 & if\ u\ is\ the\ receptive\ field \\ of\ the\ i-th\ basis\ function & \\ 0 & otherwise \end{cases} \tag{1}$$

The application of such functions means that if the basis functions are selected, the sum of the corresponding weights will form the network's output value independently of the exact position of the input point in the receptive fields of the basis functions:

$$y(u) = \sum_{i=1}^{C} a_i(u) w_i \tag{2}$$

The output of the network can be obtained without the need of any multiplication in the case of the binary basis function. As the output layer is the only trainable layer of the CMAC, and as this output layer performs linear operations, the simple LMS algorithm can be applied to train the network:

$$w(k+1) = w(k) + \mu\varepsilon(k)a(k) \tag{3}$$

where k is the discrete time index, ε(k) is the error of the network and a(k) is the association vector at step k. As it is a sparse binary vector only those weights will be modified during the training process which take part in the forming of the output value.

3 Parallel Implementations of Neural Networks

Fast implementations of neural network applications are useful because of the very high number of required arithmetic operations. Such implementations might use massively parallel computers as well as digital or analog hardware designs. This section briefly discusses the use of the various possible parallel devices.

• General purpose parallel computers. Fine-grain parallel implementations on massively parallel suffer from the connectivity of standard neural models which results in costly information exchanges. Coarse grain parallel implementations are mainly applied to neural learning, so that their efficiency suffers from the sequentiallity of standard learning algorithms such as stochastic gradient descent.

• Dedicated parallel computers. Neuro-computers are parallel systems dedicated to neural computing. They are based on computing devices such as DSPs (digital signal processors), or neuroprocessors. Their use suffers from their cost and their development time: they rapidly become out-of-date, compared to the most recent sequential processors. Most well-known neurocomputers are described in [5, 6].

• Analog ASICs. Many analog hardware implementations have been realized. They are very fast, dense and low-power, but they introduce specific problems, such as precision, data storage, robustness. On-chip learning is difficult.

• Digital ASICs. Many digital integrated circuits have also been designed for neural networks. Compared to analog chips, they provide more accuracy, they are more robust, and they can handle any standard neural computation. They usually implement limited parts of neural networks, so as to be included in neuro-computer systems ([4, 8]).

• The FPGA solution. The appearance of programmable hardware devices, algorithms may be implemented on very fast integrated circuits with software-like design principles, whereas usual VLSI designs lead to very high performances at the price of very long production times (up to 6 months).

FPGAs, such as Xilinx FPGA ([9]), are based on a matrix of configurable logic blocks (CLBs). Each CLB contains several logic cells that are able to implement small logical functions (4 or 5 inputs) with a few elementary memory devices (flip-flops or latches) and some multiplexers. CLBs can be connected thanks to a configurable routing structure. In Xilinx FPGAs, CLBs can be efficiently connected to neighbouring CLBs as well as CLBs in the same row or column. The configurable communication structure can connect external CLBs to input/output blocks (IOBs) that drive the input/output pads of the chip.

An FPGA approach simply adapts to the handled application, whereas a usual VLSI implementation requires costly rebuildings of the whole circuit when changing some characteristics. A design on FPGA requires the description of several operating blocks. Then the control and the communication schemes are

added to the description, and an automatic 'compiling' tool maps the described circuit onto the chip. Therefore configurable hardware appears as well- adapted to obtain efficient and flexible neural network implementations.

Neural Networks on FPGAs: Specific Assets

As stated above, FPGAs offer a cheap, easy and flexible choice for hardware implementations. They also have several specific advantages for neural implementations:

- Reprogrammable FPGAs permit prototyping
- FPGAs may be used for embedded applications, when the robustness and the simplicity of neural computations is most needed, even for lowscale productions.

4 Implemented CMAC Network Block Modules

The most important characteristics are the number of inputs, inputs' limits, number of internal points, generalization parameters, the basis function's type. For a better use of the capacity of the FPGA instrument the above mentioned parameters are a multiple or exponent of two.

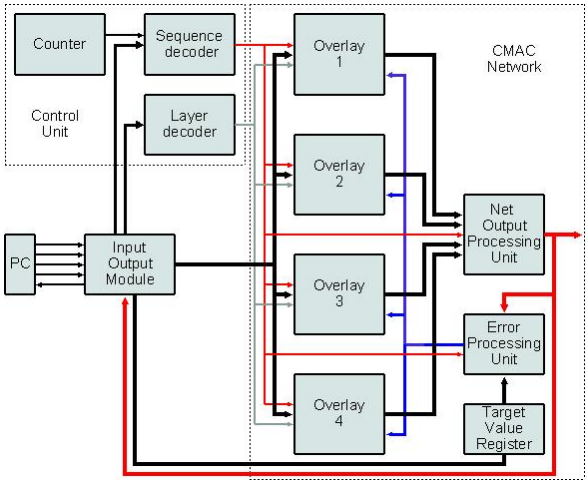

Figure 4
Implemented CMAC network block schemtic

The implemented network consists of the following main modules: the input/output module, the control module, and the network itself. The input output module assures the weight initialization, the data introduction and extraction. The control unit drives the network both in the output elaboration and the training

phase. The network itself is composed of several functional subunits, which were separately designed in VHDL. The modular design assures a network with a high flexibility and easy manageability. Figure 4 shows the implemented CMAC network block's schematic.

4.1 The Input/Output Module

The input/output module: Due to the fact that, by construction, the utilized FPGA development board uses the PC parallel port for downloading the configuration bit-stream, the easiest and most forthcoming way to exchange data with the CMAC network also is to use the same port. The issue that arises by doing so, is the limited number of bits available, which requires a custom serial protocol to be put in place. Hence, a synchronous, full-duplex serial communication module, the input/output module has been implemented. It manages the delivery of the initial weight values, the network input and the retrieval of the network output.

4.2 The Control Unit

The control unit is composed of a binary counter a sequence decoder and a layer decoder. This unit elaborates the different signals to control the network in different phases.

For each training point a time step is composed of seven cycles. Four cycles are used to compute the neural network output and three to calculate and update the new weight values (Figure 5). The computation cycles take place after the input and target values have been uploaded.

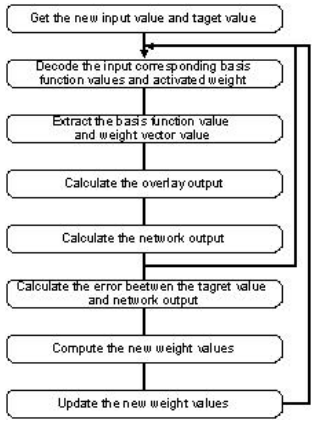

Figure 5
Flowchart of the neural network computation

4.3 CMAC Network Overlay Module

Two versions of the network have been developed. In the first version the weight values were stored in internal registers and the used triangular basis functions were implemented using logical elements (Figure 6). The main drawback of this approach is the fact that the most of the flip-flop type resources of the FPGA have been consumed by the weight storage. As a result it was impossible to implement large networks by this means.

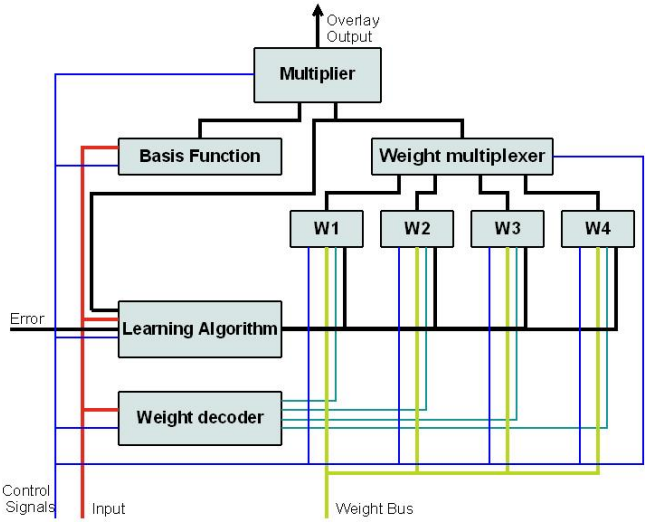

Figure 6
Overlay module structure I

In the second version of the network architecture we used Block RAMs to store weights and the basis function values (Figure 7).

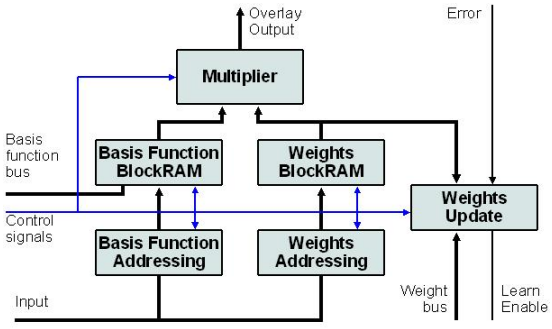

Figure 7
Overlay module structure II

By applying this change of FPGA resource utilization strategy, several advantages have emerged. The implementation has become more flexible as several basis function types can be now easily programmed.

The weight address and basis function address can be easily computed by integer division and by the rest of the integer division. In our Block RAM implementation the above mentioned operation can smoothly be implemented by selecting the higher or the lower bits of network input value.

In the case of multiple inputs the weight addressing is accomplished by simple mapping of the weights addresses on higher and lower memory address positions.

5 Experimental Results

These modules were developed in VHDL description language without using any schematics. The implemented networks were parameterized. Any other network can be generated by modifying the network parameters defined in the top level module.

For an easy development and tests an interface was created in Matlab with a driver implemented in Visual C++. The Visual C++ module contains the following functions: basis function upload, weights' initial value upload, inputs and target value uploads, the network output download to PC which, were accessed from Matlab.

Multiple tests and multiple networks with different parameters were tested. Table 1 contains the network parameters used in the performed experiments. The figures in the following section present some measurement results. These figures were recorded as results of a network with two input variables.

Table 1

Network parameters

Parameter name	Parameter value
Number of inputs	2
Number of bits per input variable	8
Basis functions receptive filed dimension	32x32
Number of overlays	4
Number of bits for weight value	6
Number of bits for basis function values	4

In the next figures 3D plots are presented for two target surfaces (Figure 8 and Figure 13), with a two variable function approximation. The subsequent figures contain a few samples of the learning process, starting with the response given using randomly initialized weights (Figure 9 respectevly Figure 14). As one can easily see, how the network gradually learns its tasks (in Figures 10, 11 for the

first surface and in Figures 15, 16 for the second surface) and how the error decreases (Figures 12 and 17). It should be mentioned that all parameters use integer representation. Obviously better accuracy could be achieved by using floating point or fixed point arithmetics.

Table 2
Resource utilization

Nr of resources	Used	Available	Used %
Slices	132	7680	1.7
Slice Flip Flops	107	15360	0.7
4 input LUTs	232	15360	1.5
Bonded IOBs	65	173	37
BRAMs	8	24	33
MULT18X18s	4	24	16.5
GCLKs	3	8	37

As it can be seen from Table 2, the resource utilization of the current implementation is very low, a version with higher precision would comfortably fit on the FPGA circuit.

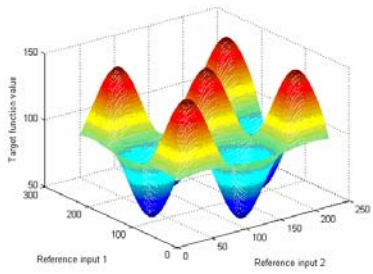

Figure 8
Reference surface

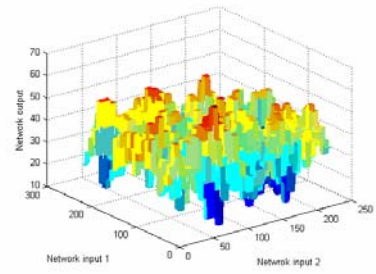

Figure 9
Initial form of surface

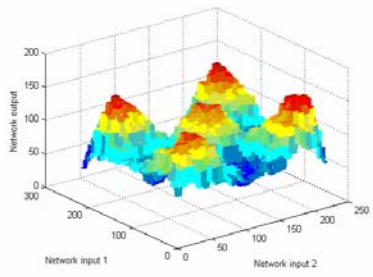

Figure 10
Learned surface after 20 epochs

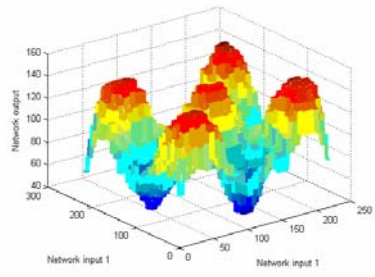

Figure 11
Learned surface after 40 epochs

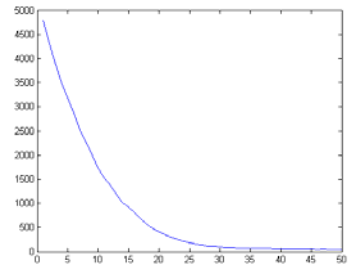

Figure 12
Squared aproximation error

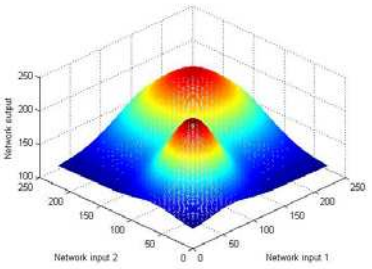

Figure 13
Reference surface

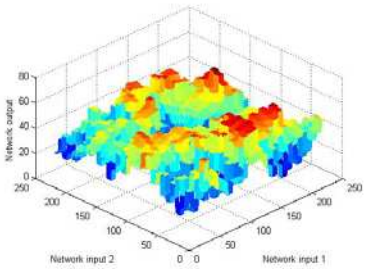

Figure 14
Initial form of surface

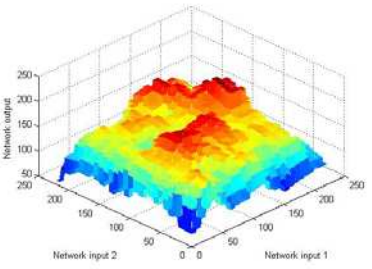

Figure 15
Learned surface after 40 epochs

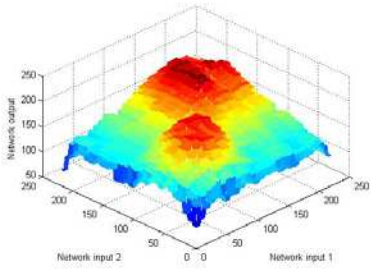

Figure 16
Learned surface after 100 epochs

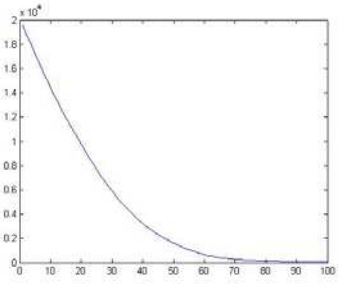

Figure 17
The trajectory tracking

Conclusions

A CMAC type hardware implemented network with one and two inputs has been developed. Due to the nature of the platform (FPGA), a very flexible architecture took shape, where most of the parameters can be modified. In the hardware implemented CMAC controller the follow error can be decreased by increasing the number of bits used for parameter representation, and for input coding.

Using an FPGA with more resources, the presented controller can easily be modified to use more than two inputs. The developed network is very fast, in 8 clock cycles it can obtain the network output. One of the main novelties of this implementation is that on-chip dynamic learning is performed without significant loss of efficiency and precision while maintaining reasonably low FPGA resource utilization.

References

[1] J. S. Albus, "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC)," Transaction of the ASME, Sep, 1975, pp. 220-227

[2] Horváth Gábor, Neuralis hálozatok és műszaki alkalmazások, Műegyetemi Kiadó, Budapest

[3] Horváth, G. Szabó, T. “Kernel CMAC With Improved Capability”, IEEE Transactions on Systems, Man and Cybernetics, Part B, Volume: 37, Issue: 1, 2007, pp. 124-138, ISSN: 1083-4419

[4] W. Eppler, T. Fisher, H. Gelnmeke, T. Becher, G. Kock. High Speed Neural Network Chip on PCI-Board. In Proc. MicroNeuro, 1997, pp. 9-17

[5] T. Nordstrem, B. Svensson. Using and Designing Massively Parallel Computers for Artificial Neural Networks. Journal of Parallel and Distributed Computing, 14(3), 1992, pp. 260-285

[6] M. Schaefer, T. Schoenauer, C. Wo I ff, G. Hartmann, H. Klar, U. Ruckert. Simulation of Spiking Neural Networks - Architectures and Implementations. Neurocomputing, 2002, (48), pp. 647-679

[7] Brassai Sándor Tihamér, Dávid László, Bakó László, Hardware Implementation of CMAC-based Artificial Network with Process Control Application, Timişoara, Transaction on Electronics and communication, Scientific buletin of the „Politehnica” University of Timisoara, 2004, pp. 209-213, ISSN 1583-3380

[8] J. Wawrzynek, K. Asanovi~, N. Morgan. The Design of a Neuromicroprocessor. IEEE Trans. on Neural Networks, 1993, 4(3):394-399

[9] Xilinx, editor. The Programmable Logic Data Book. Xilinx, 2002

Virtual Space with Enhanced Communication and Knowledge Capabilities

László Horváth, Imre J. Rudas

Institute of Intelligent Engineering Systems, John von Neumann Faculty of Informatics, Budapest Tech
Bécsi út 96/B, H-1034 Budapest, Hungary
horvath.laszlo@nik.bmf.hu, rudas@bmf.hu

Simona Vaivoda, Zsuzsa Preitl
Dept. of Automation and Applied Informatics, Fac. of Automation and Computer Science, "Politehnica" University of Timisoara
Bd. V. Parvan no. 2, RO-300223 Timisoara, Romania
Gs1961@aut.utt.ro, zsuzsap@aut.utt.ro

Abstract: Engineering for development, production, and other product related company activities are being organized in virtual systems for lifecycle management of product data. This new age of engineering is a result of the continuous development for step-by-step moving of product design, analysis and other production engineering activities from conventional environments to modeling environments during the eighties and nineties. However, more developments were needed in knowledge based decision assistance in order to realize modeling of human intent, increasingly complex product structures, and efficient communication of product data. In recent years, a great change of engineering methodology and software produced program products in knowledge technology and advanced local and global communication. Group work of engineers is increasingly organized around special portals for engineering on the Internet. Although intensive research activities produced outstanding methods in knowledge-based solutions for engineering, these methods have not widespread in the industrial modeling practice. This paper attempts to evaluate the possibility that recent communication and knowledge intensive engineering modeling can be developed into communication and knowledge intensive virtual space technology. Paper starts with a discussion on integrated application of different groups of product modeling techniques. Following this, methods are evaluated for the management of product data (PDM). Next section emphasizes aspects, contexts, and intents as primary issues in modeling for relationships in product data and decisions by engineers. Finally, methods for the management of engineering activities for lifecycle of products are summarized, considering communities of engineers around Internet portals.

1 Introduction

By the beginning of the 21st Century, information technology for engineering had been developed into integrated modeling of structural, mechanical, electrical and electronic elements in products. Engineering applications utilize Internet technology to organize remote individuals and groups on different hardware, software, and modeling platforms into project-based communities of engineers.

The new situation in engineering generated a need for integrated understanding of modeling techniques. The following issues have primary importance.

- Description of engineering objects.
- Management of product information.
- Embedding corporate knowledge in product model.
- Communication of engineers in wide area computer environments.
- Features of Internet mediated computer systems.

A study by the authors focused on interrelations of modeled objects, description of highly associative products, communication between engineers and modeling procedures, and application of stored experience in engineering. High number of coordinated modeling and problem solving techniques were considered from the current advanced engineering practice. Authors published the results in [2].

An integrated experimental system for modeling of engineering objects and product lifetime management (PLM) was established in the Laboratory of Intelligent Engineering Systems at the Institute of Intelligent Engineering Systems, John von Neumann Faculty of Informatics, Budapest Tech. The purpose of this new laboratory is verification of results of research projects in knowledge and communication intensive engineering. Based on comprehensive and robust professional engineering software, this installation comprises recent advanced CAD/CAM, human-computer, collaborative, product data management, Internet portal, and intelligent computing software.

This paper gives an evaluation about the development of recent communication and knowledge intensive engineering modeling into communication and knowledge intensive virtual space technology. Its contribution is a new approach to integration of current modeling procedures regarding the following essential issues.

- Integrated shape centered descriptions.
- Scenario of product data management.
- Characteristics of human intent.
- Knowledge in product modeling.
- Communities of engineers.

Paper starts with a discussion on integrated application of different groups of product modeling techniques. Following this, methods are evaluated for the management of product data (PDM). Next section emphasizes aspects, contexts, and intents as primary issues in modeling for relationships in product data and decisions by engineers. Finally, methods for the management of engineering activities for lifecycle of products are summarized, considering communities of engineers around Internet portals.

2 Integration in the Context of Product Development

By the middle eighties, extensive but non-integrated information in models could not serve the changed demands against product engineering in the industry any more. Therefore, development of CAD/CAM systems was concentrated on integration of separate model descriptions. As a general solution, development of comprehensive standard for integrated product information model (IPIM) was bought into foreground. This work was coordinated by the International Organization for Standardization (ISO) within the STEP (ISO 10303) project [1].

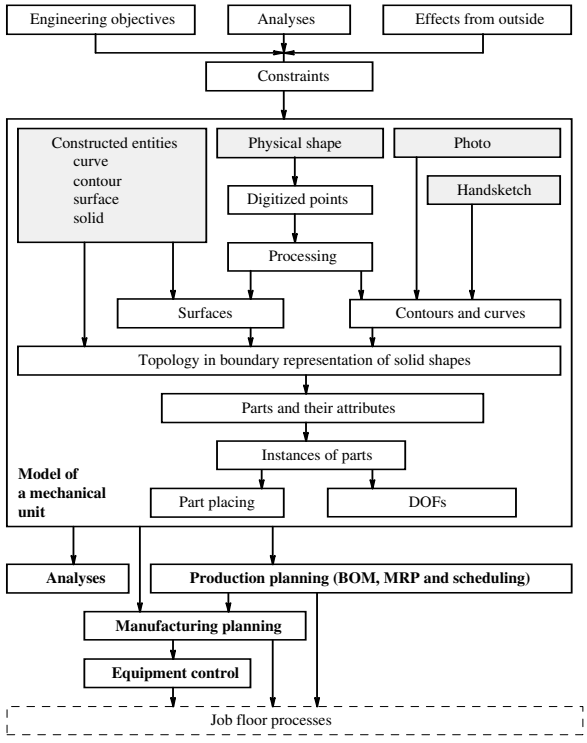

Figure 1
Integrated description of a mechanical unit

What does mean integration in the context of product development? Integrated solution for product modeling is demanded to offer software tools for creation, storage, retrieval, and application of arbitrary product related information during the lifecycle of a product, from the initial specification to recycling. Interdependencies of engineering activities are emphasized. For example, recycling must be considered at selection of material for parts. Other typical area of interdependency is definition part design in the context of manufacturing. Eventually, lifecycle management and assessment of products are considered, as they are discussed in [9] regarding approaches and visions towards sustainable manufacturing.

Integration assures that output information by any program can be used as input information for any relevant program. An important benefit of integration is increased chance for the consistent product description as enforced by modeling procedures. Regarding the related processes, one of the important issues is workflow. In [10], a new mathematical model has been introduced for workflow system synthesis. This model is partly based on a methodology formerly developed for processing network synthesis, the P-graph framework. This framework includes a specific modeling techniques and effective algorithms for network synthesis.

Fig. 1 explains integrated definition of shapes and their relationships for consistent information at design of mechanical units. Model describes information for shapes, technical specification, and relationships of shapes. Shape of a part is described by its boundary. By using of the well-approved boundary representation, curves and surfaces constituting the boundary are connected by Euclidean topology in a solid shape. During processing of geometry of a solid, correct information about surfaces and their intersection curves in the boundary is necessary to accomplish correct geometric operations. For example, when a solid is cut by a surface in three dimensions, high number of operations refreshes geometry and topology for changed shape of the solid. Solid shape is completed by part attributes to achieve part model. For repeated application of a part model in the same or different units, parts are instanced.

Relationships are defined between pairs of parts in a mechanical unit by placing and allowed movement information. Placing is defined by constraints while movements are allowed according to degrees of freedom (DOF) of relative movements. It can be concluded that associative engineering objects are described and that higher level engineering objects are described in the context of lower level objects.

Primary sources of shape information are constructed or built-in shapes in model spaces in the form of curves, contours, surfaces, and solids. Other sources of the same importance are described as follows.

- Planar or spatial hand sketches of contours for parts and assemblies.

- Curve and contour information captured from a drawing or a photo.
- Digitized physical shapes in the form of clouds and arrays of points are processed into surface models.

Constraints for the definition of a mechanical unit are extracted from engineering objectives, results of analyses, and other outside effects such as earlier decisions, standards, agreements, legislations, etc. Directions of further integration of mechanical units are analysis, equipment control, manufacturing planning, and production planning. Other areas of integration may be marketing, customer services, and recycling.

Integrated modeling of a part and a mechanical unit are summarized in Fig. 2 and Fig. 3, respectively. In the example of Fig. 2, a part is initially defined as a solid box. Following this, the initial shape is modified by two form features, tabulated from *Contour 1* and *Contour 2* along *Vector 1* and *Vector 2*, respectively. These modifications result in the final shape of the *Part A*. Connections of surfaces at curves are described by topology constituting structural description for the boundary of the shape. In the topology, curves (lines) and surfaces are mapped to edges and faces, respectively. Vertices connect topological edges. Consistency of topology is assured by the application of appropriate Euler operators at its definition. Topology can be checked by topological rules [3]. Solid shape is completed by non-geometric information and stored as *Part A*.

Parts are assembled into units and therefore are constructed in the context of context of assembly, considering an assembly model space. Some contours in parts are copied from part to part not to be repeatedly constructed them.

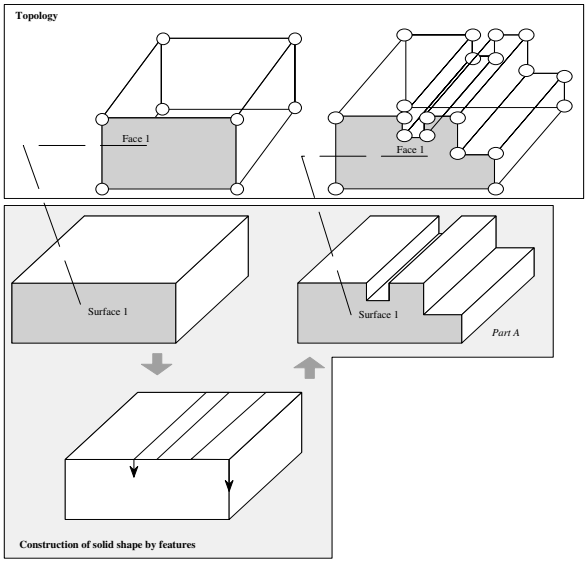

Figure 2
Integrated modeling of a part

Part A, Part B, and *Part C* constitute the *Unit 1* in Fig. 3. Placing of *Part B* and *Part C* on *Part C* is defined by *Contact* and *Coincide* constraints between parts. In order to allow its controlled relative movements *Part B* gets one translation (1T) degree of freedom to move along the direction denoted by arrow. As a complex operation on the shape of *Part A* cutting with *Surface 2* modifies shape of *Part A* so that shape of *Unit 1*. In the course of the cutting operation, a region of the *Surface 2* is defined by its trimming with new boundary lines and curves. Topology is also completed and new surface and curve entities are mapped to topological entities.

In the model of a mechanical unit, entities and their attributes are interrelated by bi-directional associativity definitions. This method is essential to save consistency of the construction at its modification and development. Non-associative and non-constrained shapes and dimensions are free to modify. However, in these operations, checking for consistency and collision is strongly recommended.

3 Management of Product Data Systems

During the nineties, model databases became more and more large and complex. Besides product structure based organization of product data, two additional demands were arisen. One was that models from different modeling systems were required to handle in the same product data system. The other was a strong demand for engineering process-oriented services. In order to fulfill these new demands, product data management (PDM) systems were established for product centric data base management.

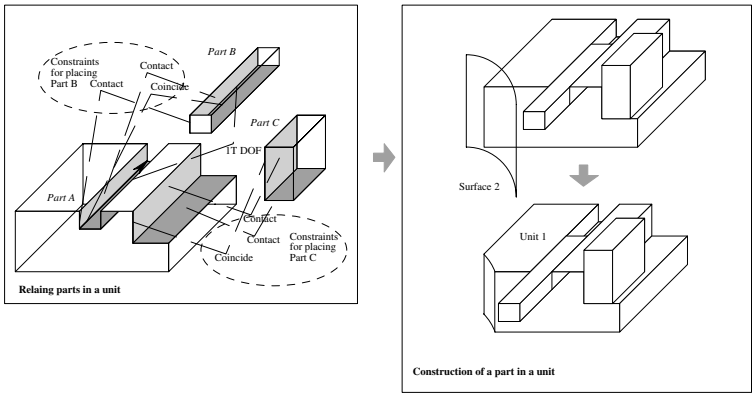

Figure 3
Integrated modeling of a mechanical unit

PDM systems were developed in integration with modeling systems. By now, they offer functionality for flexible product information handling, change management, control of engineering processes, and handling of product structure. Advanced PDM systems are integrated with other company activities. They are capable of efficient exploration of consistency related and other problems in a complete product information environment.

Essential PDM functions are surveyed in Fig. 4. Product data is handled for modeling systems $MS_1 - MS_n$. Engineers $ENG_1 - ENG_n$. are working in a group. Multiple projects and roles can be assigned for an engineer. Engineer can define, retrieve and maintain product structure subsets called as views. View can be selected by parameters, copied to other points in data structure of the same or other product. In this way, arbitrary context of model information can be handled as a unit.

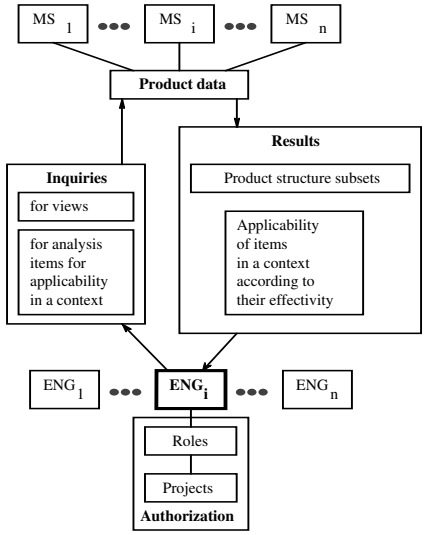

Figure 4
Scenario of product data management

Additional function of PDM systems is checking the applicability of a part or other item in a given context. Criteria of application are called as effectivity. This function is supported by specification of effectivity range for product items. Views can be selected according to effectivity. Project and role based authorizations are mapped to view, access and modify data sets in extended enterprises.

4 Contexts, Aspects, and Human Intents

Three concepts in the title of this part highlight the main difference between conventional and future style of definition of engineering objects in product modeling. Shapes and other entities are created in the context of existing entities.

Contextual connections of modeled product objects are applied at automatic definition of associative connections, assuring consistency of model during the entire lifecycle of product. Contextual links are specified as constraints. When an entity changes, the entities contextual with it are also changed. Contextual links can be defined for changes in one direction or in both of directions.

A typical contextual link is explained in Fig. 5. A swept surface is created in the context of a generator (c_g), a path (c_p), and a spine (c_s) curve, and a pivot point (P_j). The context is specified as follows.

- Generator curve is connected to the end of the path curve at the pivot point.
- Moving of the generator curve along the path curve creates the swept surface.
- Normal of plane of the generator curve coincides the tangent (t) of the spine curve at each point along the path curve.

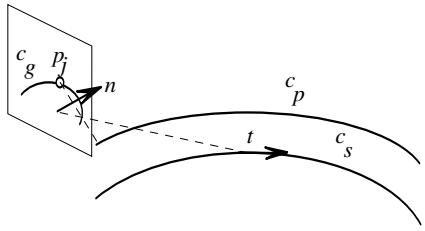

Figure 5
Definitions of a swept surface in context of three curves

Modeling in an aspect is a means of definition of modeled objects according to their function and application. In Fig. 6, three different form features are defined according to one manufacturing and two different construction aspects. Steps are merged into a single form feature consisting for planning of the control of production equipment. At the same time, individual steps have different functions in the construction. Extraction of form features from shape representations can be applied for intelligent dimensioning of mechanical parts [5].

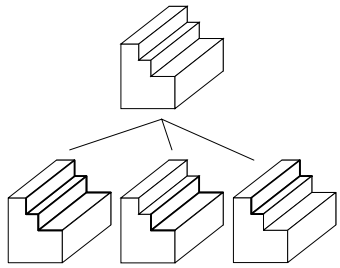

Figure 6
Aspects as form features

Typical engineering processes are controlled by intent of different humans. Intent of several humans must be considered at decision [4]. Description of intent characteristics in a product model would be especially beneficial at these multiple intent based decision. In order to establish intent modeling, as result of an analysis, a set of characteristics was concluded for design intent. Characteristics include type to categorize its content, status to describe its strength, and status of the human who is its source (Fig. 7). Technical content of intent is to be represented as knowledge.

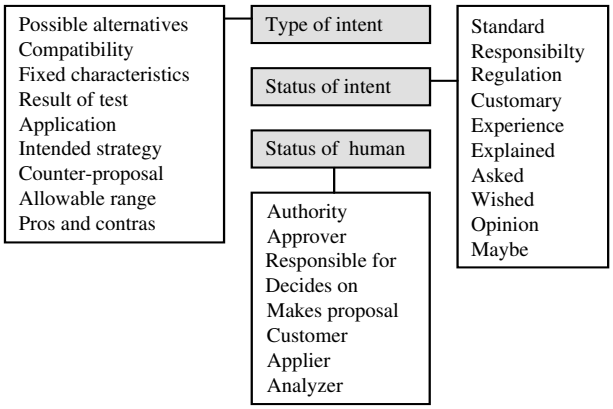

Figure 7
Characteristics of human intent

Typical intents and their representation in product model are given in Fig. 8. Engineering objectives determine behaviors of a product at given sets of circumstances called situations. Functions are modeled as functional associative links between pairs of product objects (see details in Fig. 11). Intent for consistent product structure is represented by structures as topology and constraints. Higher priority intents come from chief engineers, authorities, etc. Finally, engineers make decisions on parameters and their relationships then record them as constrained and unconstrained associative definitions between or parameter values for product objects.

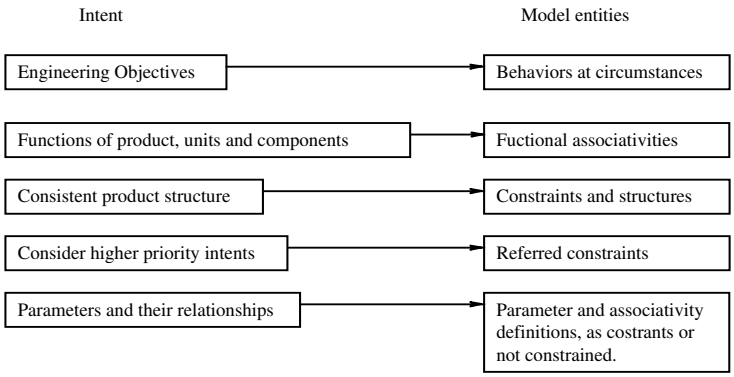

Figure 8
Typical intents and their description in product model

Knowledge engineering produced methods for emulation of activities and intelligence of engineers during eighties and nineties. In that time, efforts to achieve effective methods to embed knowledge in product modeling procedures and product models were mostly failed. One of the causes of difficulties at application of knowledge-based methods in engineering is that knowledge must be acceptable by responsible engineers. Application of non-verified knowledge makes supervision of engineering activities difficult and it can be dangerous.

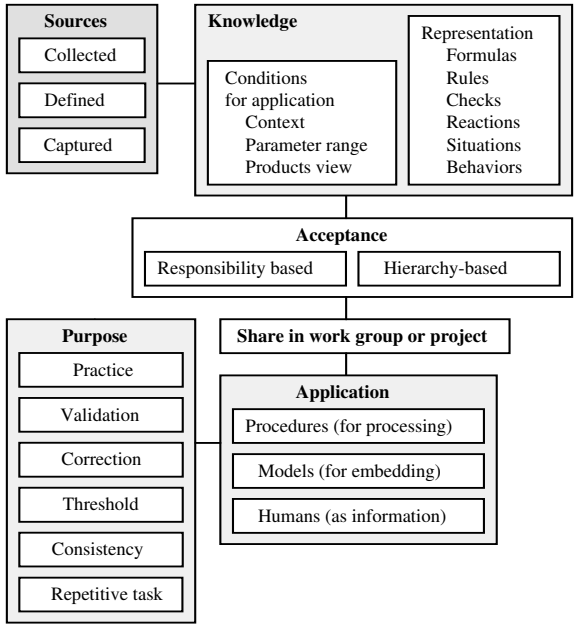

Figure 9
Knowledge in product modeling

Sources, content, applications, and purposes of knowledge are outlined in Fig. 9. Knowledge is best to capture in work groups and projects by direct definition by humans, extraction from successful practice, verification, and experience. It is shared amongst participants of work groups and projects. Engineers embed knowledge in models by responsibility and hierarchy-based acceptance. Circumstances as parameter ranges, products or their views, and contexts can specify conditions for application of knowledge.

Simple, efficient and engineer understandable forms of knowledge description in current in present engineering modeling are formulas, rules, checks, and reactions. Formula is a record of an associative link. Rule predicts while check verifies actual value of parameters. Reaction is an action programmed for events of given parameter values. As an example, an application tool is shown in [7]. This tool utilizes knowledge-based-engineering environment in a PLM system. It uses design rules for aerospace structures to add details to a conceptual design.

Product and related engineering objects are defined by sets of parameters. Modeling procedures reveal combinations, select suitable combinations, and choose optimal combinations of parameters (Fig. 10). Analysis of interactions between parameters and selection of the most influential parameter are requires support by knowledge representations and associative parameter definitions. Design objectives, limitations, and other criteria can be applied for the creation of set of optimal parameters for product objects.

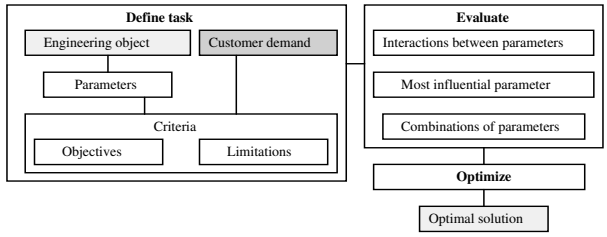

Figure 10
Handling object parameters

Model description of functions and functional relations of product objects is relative new. Functions are attached to shape oriented product model in the form of functional associative definitions for parts, functional subsystems, and product variants (Fig. 11).

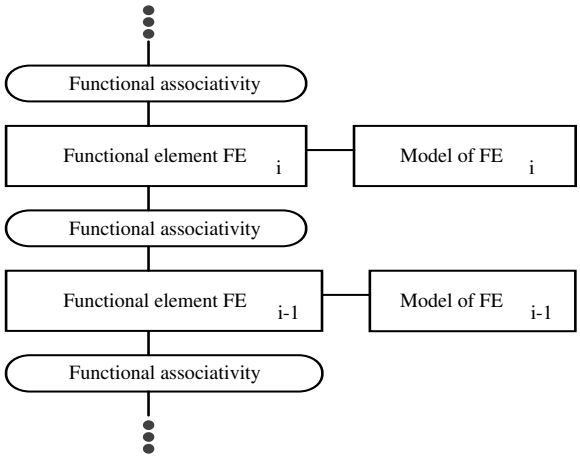

Figure 11
Modeling of functions of a product

5 Communication Related Issues

Despite high level of communication between modeling procedures communication of humans has not been replaced by communication of computer procedures. Moreover, one of the most important and effective developments in engineering could be experienced in communication of humans during the last decade. Humans control the processing of high amount information and make thousands of decisions with their personal responsibility. At the same time, integrating cooperative procedures and embedding human intent and proven knowledge descriptions in product models a great change in automated engineering during the forthcoming years.

Communities of engineers in integrated modeling of products are moving into standard Internet environments. Substantial developments support this tendency in recent PLM technology. In [8], collaborative virtual prototyping of product assemblies is introduced for the Internet. Paper [6] describes an integrated product and process modeling (IPPM) framework for collaborative product design through the Internet. Hierarchical and heterarchical dependencies are applied between decomposed smaller design problems.

Software functionalities are available for human communication purposes according to product related objectives (Fig. 12). Product lifecycle collaboration is provided for remote engineers and groups. Participants can use different hardware and software platforms and modeling systems, regardless of geographical location.

Using recent achievements in information technology, engineers work in well-organized, powerful, professional, economic, dynamic and secure environments.

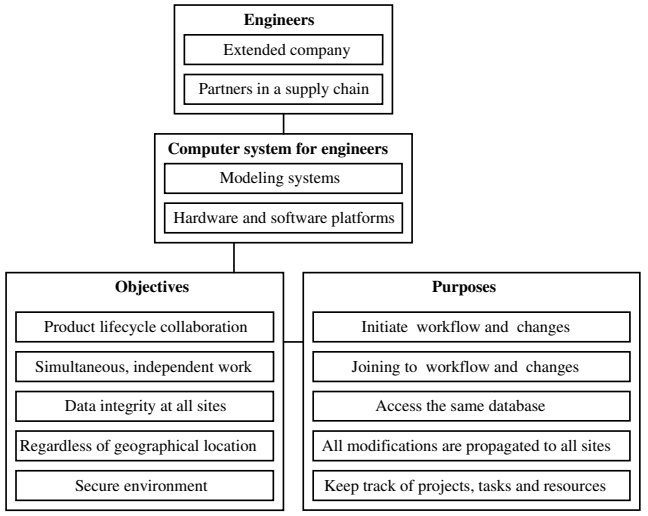

Figure 12
Communities of engineers

All authorized group member initiate workflows and product changes, access the same database, share the same resources, and keep track of projects, tasks and resources at any time. The product data is identical for all participants. All modifications are propagated to all workstations to maintain integrity of product data. Independent workstations are active simultaneously, sometimes regardless of the network connection.

Conclusions

This paper focuses on evaluation of capabilities of PLM systems to accommodate communication and knowledge centered intelligent computing methods. For that reason, it contributes with a new approach to integration of current modeling procedures regarding the following issues.

- Integrated description of a mechanical unit.
- Scenario of product data management.
- Characteristics of human intent.
- Knowledge in product modeling.
- Communities of engineers.

Lifecycle solutions for product modeling provide assistance for engineers in order to assure consistency at definition of new engineering objects in product model.

Product centric data base management in PDM systems is integrated with product modeling other company activities. Modeling activities of engineers are organized around portals on the Internet establishing communication intensive collaboration.

This paper evaluates effects of the new technology and environment on engineering modeling in order to prepare the development of knowledge and communication intensive intelligent procedures for current PLM systems with the following characteristics.

- Integrated solutions for product modeling.
- Product centric database management.
- Possibility for description of contexts, aspects, and intents in product model.
- Advanced definition of engineering objects.
- Handling of interrelated product object parameters.
- Modeling of product functions.
- Advanced communication within groups of engineers.

Engineering decisions are typically controlled by intent of more than a single human. Saving human intent in product model is preferred in one form of engineer friendly representations. Personal responsibility for decisions requires controlled knowledge and makes application of intelligent computing at engineering difficult.

Acknowledgements

The authors gratefully acknowledge the grant provided by the National Office of Research and Technology (Hungary) and the Agency for Research Fund Management and Research Exploitation (KPI, Hungary) within the Hungarian – Romanian Intergovernmental Science & Technology Cooperation Programmes. Project number is RO-51/05. The authors also gratefully acknowledge the grant provided by the OTKA Fund for Research of the Hungarian Government. Project number is K 68029.

References

[1] Zha, X. F., Du, H., "A PDES/STEP-based Model and System for Concurrent Integrated Design and Assembly Planning," in *Computer-Aided Design*, Vol. 34, 2002, pp. 1087-1110

[2] L. Horváth, I. J. Rudas, *Modeling and Problem Solving Methods for Engineer*s, Elsevier, Academic Press, Amstaerdam, New York, 2004

[3] L. Horváth, "Emerging Intelligent Technologies in Computer Aided Engineering," *in Proceedings of the 3rd IEEE International Conference on*

Intelligent Engineering Systems, INES'99, Stara Lesná, Slovakia, 1999, pp. 427-436

[4] L. Horváth, I. J. Rudas, C. Couto, "Integration 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

[5] Chen, K.-Z., Feng, X.-A., Lu, Q.-S., "Intelligent Dimensioning for Mechanical Parts Based on Feature Extraction," in *Computer-Aided Design,* Vol. 33, No. 13, (2001): pp. 949-965

[6] Yoon-Eui Nahm, Haruo Ishikawa, "Integrated Product and Process Modeling for Collaborative Design Environment," in *Concurrent Engineering*, Vol. 12, No. 1, pp. 5-23, 2004

[7] Jin-Woo Choi, Donald Kelly, John Raju, Carl Reidsema, "Knowledge Based Engineering System to Estimate Manufacturing Cost for Composite Structures," in *Journal of Aircraft*, Vol. 42, No. 6, pp. 1396-1386, 2005

[8] Shyamsundar, R. Gadh, "Collaborative Virtual Prototyping of Product Assemblies over the Internet," in *Computer-Aided Design*, Volume 34, Issue 10, 1 September 2002, pp. 755-768, 2002

[9] E Westkämper, L Alting, G Arndt, "Life Cycle Management and Assessment: Approaches and Visions towards Sustainable Manufacturing," in *Journal of Engineering Manufacture*, Professional Engineering Publishing 0954-4054, Volume 215, Number 5 / 2001, pp. 599-626

[10] J. Tick, Z. Kovács, F. Friedler, "Synthesis of Optimal Workflow Structure," in *Journal of Universal Computer Science*, Vol. 12, 2006, No. 9, pp. 1385-1392

An Introduction on Cognition System Design

Claudiu Pozna

Departament of Product Design and Robotics, University Transilvania of Brasov,
Bd. Eroilor 28, 500036 Brasov, Romania
Phone: +40-268-418967, Fax: +40-268-418967, E-mail: cp@unitbv.ro

Abstract: Present work is an introduction on cognition system design. This work is structured in two parts: the first consists on a phenomenological analyze of Artificial Intelligence collocation which will generate seven questions needed in cognition system design; the second part represents possible answers for the first questions. The last part represents also is the opportunity to present the plausible reasoning theory and to solve two examples.

Keywords: plausible reasoning, Bayesian theory, model

1 Introduction

In 2002 by IST – 2002 2.3 2.4 (*published in the Official Journal of the European Union*) the European Union decided that the researches in cognitive systems design are considered strategically objective. The cognitive systems are defined by [1] like a system which understands, learn and are developing by social or individual interaction. These interactions include subsystems for perceptions and actions, which are already known like robots. The third element of the cognition system is the reasoning subsystem which will manage the first two. This is the biggest challenge of the cognition system design because it implies the modeling of human reasoning.

Modeling the human reasoning is an interdisciplinary problem because involves also philosophical and psychological knowledge [11-15]. The Artificial Intelligence (AI) domain covers a part of these problems [2, 3, 4]. This is the reason why we intend to start our study on cognitive system by a better understanding of (AI). We think that it is important to reveal – here in a phenomenological way – what we aspect from the science which is named Artificial Intelligent. Usually the first step of such analyses is to find the appropriate questions which will make deeper the phenomenon understanding. The results of the phenomenological researches on artificial intelligent (AI) collocation are seven questions which can drive to intelligent product

construction. The second part of the paper intends to answer the first questions: *Are there known theories that have as object the human knowledge?* and *How can we use them in order to develop a human knowledge model?* One possible answer could be 'The Laplace model of commune sense' or 'The Bayesian theory'. The named theory is based on reverend Thomas Bayes and on mathematician Pierre Simone Laplace results [5, 6] and was developed in [7].

The backgrounds of the present work are E. T Jayne's probability theory [7] and also the related works of Cox and E. T. Jayne, where the rules of the mentioned theory are presented. We will mention also the work of E. Yudkowsky [8] where an epistemology based on Thomas Bayesian result is presented and also J. Pearl work on causal reasoning [9]. The bridge between the Bayesian plausible reasoning and mobile robots has been inspired by the work of C. Pradalier, where the navigation of a mobile robot is controlled using Bayesian's filters [10].

Our intention is to transform the rules of 'Laplace model of commune sense' into postulates, and to present theoretical results which are obtained from these postulates (the Bayesian theorem [7] and the Bayesian filter [10]). In the end we will present two examples of plausible reasoning solutions.

2 The Phenomenological Analyses of AI Syntagma

2.1 Definition of AI

The Artificial Intelligence is a syntagma composed by two terms (intelligence and artificial) that through their nature generate an interior stress, because the term of intelligence is in the ontic acceptation bound by the human or at the most by the living being and the *artificial* attribute comes to underline the fact that we have in mind a human creation or more precisely a product achieved by the human being. In this way the Artificial Intelligence becomes a human product that imitates the intelligence features (human, eventually naturals). We must recognize that from a psychological point of view this comment amplifies the mentioned stress. In our word the intelligence has become a fetish and has generated, in this way, a psychological complex. We will remember that all the people wish to prove intelligence even many of them don't know exactly what the intelligence is. Because of that behavior, we accept hardly that intelligence can be associated with an object.

After all this considerations if we have accepted that AI is a product that copies the human intelligence, then we must understand what means *intelligence* and what means *to copy*.

The Intelligence

The intelligence is defined in several ways, from this richness we will start with the following work definition: *The intelligence is the capacity of understanding the experience and the capacity to take benefit from this understanding*.

The enunciated definition articulate causally two attributes: the experience understanding and the benefit of this understanding. If we focus at the *experience understanding* we will discover that is a tautological expression, because the experience assumes a certain understanding. For example, the experience in the Kantian sense it is more then a sensations assembly, including a certain base of knowledge.

Because of this reason will replace the *experience understanding* through a term more comprehensive that transforms the work definition in: *The Intelligence is the capacity of knowledge and the capacity to take benefit from this understanding.*

If we wish to analyze now what *take benefit* means, we have to admit that this collocation assumes ethical approaches. Because in this moment we intend to avoid such ethical approaches we will reduce the significations of the benefit and will replace this expression with: *the facility of knowledge accessing* (inclusive the ones that mention the possibility of benefit).

In this way the work definition has become: *the intelligence is the knowledge capacity and the facility of access these knowledge*. According to this definition an intelligent human being is the one that can know easily and can use this knowledge (fruitfully).

The Imitation

We will return at the *artificial* term content in the AI syntagma. We intend to copy the intelligence features of the human being and for that is important to understand what means *to copy* .To copy in an ontic sense is the operation in which the original is transposed with approximation into a product. Then, when I copy, I don't claim to perform an identical, but only to transpose certain features that I consider to be essentials. I'll give up, in this way, at all that seems to be accidental and I will perform a representation accepted by the original object concept.

To imitate is an activity that it's bounding by the knowledge because I don't imitate the object himself but I copy my knowledge regarding this object. Furthermore, when I imitate I decide that certain notions are important and other don't, and these decisions are based on my knowledge.

After that, to imitate means the approach of a certain technology. The technology assumes the knowledge of some procedures, the existence of some tools and objects (materials) where I will implement my copy. So in the knowledge process imitation I must identify all these elements.

Therefore we can conclude that when we mention the AI syntagma we refer at the copy of our knowledge about knowledge and about the access of this knowledge.

2.2 Opinions about Knowledge

The above analyze has underlined our capacity to know about knowledge process. We must mention from the beginning that the human knowledge sources are of various forms: mythical, religious, artistically, scientifically etc. and we must specify our position regarding this problem. Therefore bellow when we mention the cognition about knowledge we will understand the scientifically cognition of the human knowledge in her generality. Now it is natural to analyze what we understand through scientific cognition. The subject vastness and the space allowed to this article will be balanced trough the *opinion* term used in the following description. Also we will formulate certain opinions on the subject (see Figure 1):

- Scientific knowledge divides the reality in quasi independent domains (the systematic vision);
- For a certain domain it's start from a minimum number of fundamental troughs. When these principles ore axioms are proposed we desire that they are independent and in minimum number. The principles are carried out inductively, this process is induced by the experience and are after that adjusted through the theory results that they generate;
- Based on the principles, through deductions, theories are constructed. A theory represents the knowledge that can explain the phenomena from a certain domain of the reality;
- Based on a certain theory, a particular phenomenon is represented through the model. The model is a peculiar knowledge assembly, obtained by approximation process, that aspire to become operational;
- Scientific knowledge must be validated continuously by the experiment;
- It has as aim the a priori knowledge, more precisely we wish to know how will ensuing the phenomena before the experimentation (*voire pour prevoire*).

Conclusively the scientific knowledge has as operational element the model. The model is defined as being an approximation of the phenomenon which is constructed starting from a theory by elude the *non important* from the *important* of the phenomenon. This process (the separation in important and non important) is a subjective decision (depends on the subject – human – that know), but we have the hope that the experiment will infirm the bad decisions. We have mentioned that the model is operational, this means that the model can be used directly for obtaining the mentioned purpose: the a priori knowledge.

We can describe the phenomenon of model using in two ways: first if the model is simple we can use it directly (mental experiment), but if the model is too complicated to be used directly we must use technologies in order to obtain results. This technology contains methods (mostly mathematical), tools (mostly

computers) and support objects (paper, computer screen, etc.). We have named this operation as simulation.

It is important to approach the fact that when we imitate, we will not imitate the subject himself or the phenomena but the imagined model.

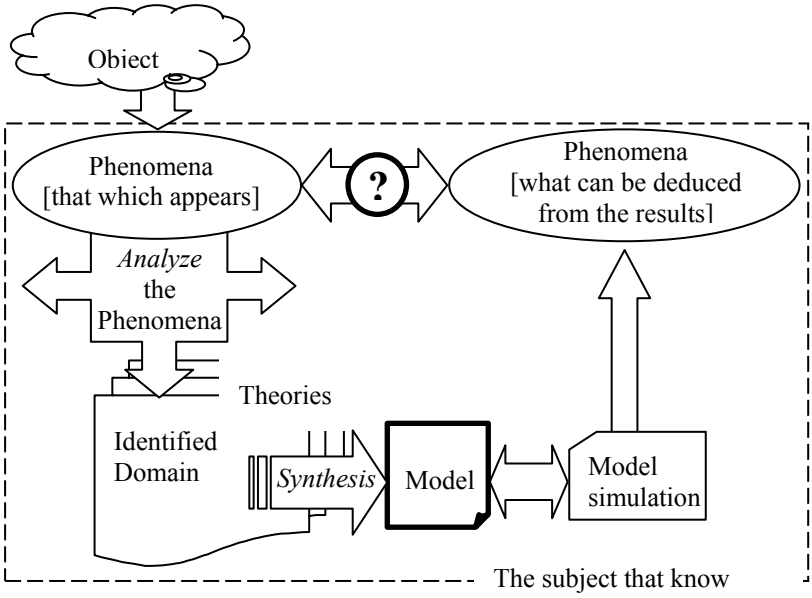

Figure 1
The model creation

2.3 Results of Phenomenological Analysis

If we will resume the previous results we can say that when we mention the AI syntagma we expect to find a science which contains the technology to copy the model's of human knowledge and the access to this knowledge. In order to construct such models this science must be linked to cognition theories.

Starting from these conclusions we can find now the appropriate questions which allows the possibility to deep the understanding of the AI and which drives to intelligent product construction.

1 Are they known theories that have as object the human knowledge?

2 How can we use them in order to develop a human knowledge model?

3 How can we simulate this model and how can we improve it?

4 What is the technology – the methods and the tools – which can be used in order to copy the model?

5 What are the properties of the object that can be transformed in intelligent object?

6 How can we experiment the intelligent object?

7 What are the ethical aspects of the intelligent object construction?

3 The Theory of Plausible Reasoning

The second part of the present paper intend to answer to the first questions: *Are they known theories that have as object the human knowledge?* and *How can we use them in order to develop a human knowledge model?*. One possible answer could be 'The Laplace model of commune sense' [2]. The background of a particular theory consists on principles or axioms. The difference between these two concepts consists on the fact that the axioms are self evident fundamental and the principles are accepted fundamental reason. This is the reason why we have chosen to name the next fundamental reasons principles.

The Principles of Plausible Reasoning

1 The representation of degree of plausibility is given by the plausibility function:

$$p:\Phi \to [0 \quad 1];\ p(A \mid X) = y \tag{1}$$

where:

Θ is a set of sentences

$p(A \mid X)$ is a continuous and monotonic function which associates a particularly degree of truth for the sentence A in the condition that sentence X is true;

2 The consistence of the commune sense requires the following property for the *p* function

$$p(AB \mid X) = p(A \mid X)p(B \mid AX) \tag{2}$$

$$p(A \mid X) + p(\neg A \mid B) = 1 \tag{3}$$

$$p(A + B \mid X) = p(A \mid X) + p(B \mid X) - p(AB \mid X) \tag{4}$$

$$p(A_i \mid X) = \frac{1}{n} \qquad i = 1...n \tag{5}$$

where $\{A_i\}|_{i=1...n}$ is a complete set of mutual excusive sentence.

Some comments are necessaries:

- by consistence we mean:
  - every possible way of reasoning a sentence must lead to the same result;
  - the equivalent sentences have an equal degree of plausibility;
- in order to obtain the degree of plausibility for a sentence we must take into account all the evidence available;
  - $p(AB \mid X)$ means the plausibility of sentence A **and** B in the condition that sentence X is true;
  - $\neg A$ means **non A;**
  - $p(A+B \mid X)$ means the plausibility of sentence A **or** B in the condition that sentence X is true.

Theoretical Results

Analyzing the mentioned postulates, theoretical results can be deduced. From the beginning we will mention that because the probability function has the same properties (1…5) it can be accepted that the plausibility function is synonymous with the probability function. This is the only reasons that theoretical results from probability theory can be transferred to the theory of plausible reasoning [7].

It is obvious that we do not intend to present exhaustive theoretical results. We will resume presenting the Bayesian theorem which can easily deduce from (1-5). If we name by d the evidence of an experiment and by $h_{i=1...n}$ a set of mutual exclusive hypotheses the Bayesian theorem tells us that the plausibility of hypothesis h_i in the condition of evidence d is equal with the plausibility of hypothesis h_i multiplied by the plausibility of evidence d in the condition that hypothesis h_i is trough and divided by the sum of the same product made for all the hypotheses of the set.

$$p(h_i \mid d) = p(h_i)\frac{p(d \mid h_i)}{\sum_{k=1...n} p(h_k)p(d \mid h_k)} \tag{6}$$

The plausibility of hypothesis h_i in the condition of evidence d is named the a posteriori knowledge, the plausibility of hypothesis h_i is named the a priori knowledge and the plausibility of evidence d in the condition that hypothesis h_i is true is named the likelihood. The sum from the denominator is named the marginalization sum.

In order to converge to the model construction we will link this theoretical result to the Bayesian filter [10]. A Bayesian filter allows to estimate the state X_t for a Markovian system in condition of knowing the observation $Z_1,..Z_t$. In order to solve this problem several steps are necessary:

- variable definition:

  $\{X_i\}_{0\le i\le t}$ the system states; $\{Z_i\}_{0\le i\le t}$ observations;

- decomposition

$$p(X_0...X_t, Z_0...Z_t) = \prod_{i=0}^{t} p(X_i \mid X_{i-1}) p(Z_i \mid X_i) \tag{7}$$

- initial knowledge:
  - the initial state distribution;

$$p(X_0) \tag{8}$$

  - the transition **model** from state i-1 to state i

$$p(X_i \mid X_{i-1}) \tag{9}$$

  - the sensor **model**;

$$p(Z_i \mid X_i) \tag{10}$$

- the question

$$p(X_t \mid Z_t...Z_0) \tag{11}$$

4 The First Case Study

In order to exemplify the mentioned theoretical results we will consider the case of a mobile robot which modifies his state (position) and – from time to time – make observations (measure his position), see Figure 2.

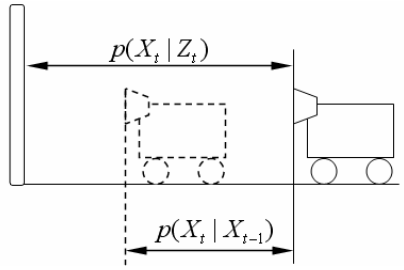

Figure 2
The mobile robot

The dynamic model of the robot is very simple (the robot has a constant speed):

$$x_k = x_{k-1} + \Delta \tag{12}$$

where: x_k is the position of the robot.

We know that this model is only an approximation of the reality and from moment two developments – knowledge improvements – are possible:

- developing our model eventual by adaptation: adjust the apropriate value of Δ or introduce new parameters;
- constructing the Bayesian filter over this model.

We have chosen the second possibility which can be mathematical described by the following equations

$$x_k^{est} = x_k + \pi^{est} \tag{13}$$

where: x_k^{est} is the outputs estimations; x_k is the model output; π^{est} is the model perturbations.

We don't know a priori the model perturbation but we can obtain, by experiments, the statistical distribution of π^{es}: $p(\pi^{est})$. This distribution accomplishes (1) so we can define the estimation plausibility like the degree of truth for the following sentence: *'the estimated output k for our model is* x_k^{est} *'*.

From (13) we have:

$$p(\pi^{est}) = p(x_k^{est} - x_k) \tag{14}$$

We must note that using the model we will obtain the state *k* from state *k-1* so we can rewrite (14)

$$p(\pi^{est}) = p(x_k^{est} - x_k) \equiv p(x_k^{est} \mid x_{k-1}) \tag{15}$$

Using the Bayesian rule (6) we can write:

$$p(x_k^{est}) \propto \sum_{x_{k-1}} p(x_{k-1}) p(x_k^{est} \mid x_{k-1}) \tag{16}$$

where: $p(x_k^{est})$ is the plausibility of the output estimation;

$p(x_{k-1})$ is the plausibility of state x_{k-1};

$p(x_k^{est} \mid x_{k-1})$ is the plausibility of the estimation when we know the state x_{k-1};

$\propto$ means proportional.

If during locomotion we measure (make observations). We can describe this process in the following mathematical form:

$$x_k^{mes} = x_k^{est} + \pi^{mea} \tag{17}$$

where: x_k^{mes} is the output measurement;

π^{mea} is the measurement perturbation.

Once again we don't know a priori the value of the measurement perturbation but if we experiment our sensor we can obtain a statistical distribution of these values. We can write:

$$p(\pi^{mes}) = p(x_k^{mes} - x_k^{est}) \equiv p(x_k^{mes} \mid x_k^{est}) \tag{18}$$

Using (6) we obtain:

$$p(x_k^{mes}) \propto p(x_k^{est}) p(x_k^{mes} \mid x_k^{est}) \tag{19}$$

If we use normalized distribution we can transform (16) and (19) in equations.

For the purpose of the Bayesian filter constructing we will return to relation (7-11):

- variable definition:

$\{x_k\}_{k\in\{0,\dots n\}}$ the system states are the position of the robot (see Figure 1a); $\{x_k^{mes}\}_{k\in\{0,\dots n\}}$ we will measure the position;

- decomposition

$$p(x_1^{est},\dots x_n^{est}, x_1^{mea},\dots x_n^{mea}) = \prod_{i=0}^{t} p(x_k^{est} \mid x_{k-1}) p(x_k^{mes} \mid x_k^{est}) \tag{19}$$

- initial knowledge:
  - the initial state distribution, is obtained after experiments, in this case we have chosen the following Gaussian distribution;

$$p(x_0) \propto \exp\left(-\frac{(x-x_0)^2}{2\cdot 0.1^2}\right) \tag{20}$$

  - the transition **model** from state k-1 to state k, is presented in (12), the mathematical form of this distribution can be obtained from experimental measurement, once again we have chosen a Gaussian distribution:

$$p(\pi^{est}) = p(x_k^{est} \mid x_{k-1}) \propto \exp\left(-\frac{(\pi^{est})^2}{2\cdot 0.2^2}\right) \tag{21}$$

  - the sensor **model**: the mathematical form of this distribution can be obtained from experimental measurement, once again we have chosen a Gaussian distribution

$$p(\pi^{mes}) = p(x_k^{mes} \mid x_k^{est}) \propto \exp\left(-\frac{(\pi^{mes})}{2 \cdot 0.05^2}\right) \tag{22}$$

- the question is the plausibility of the each state when we know the transition plausibility and the measurement (sensor) plausibility; in order to compute this results we have used (16) and (19):

$$p(x_k^{mes}) \propto \left(\sum_{x_{k-1}} p(x_{k-1}) p(x_k^{est} \mid x_{k-1})\right) \cdot p(x_k^{mes} \mid x_k^{est}) \tag{23}$$

The question response is a distribution for each *k=0…n*. This distribution has a maximum value which is the most plausibile answer to the question. More precisly, each itteration we obtaine a 2 component information: the most plausible answer (the robot position) and the value of its plausibility.

Even the initial data are not crispy because we must admit that we don't know with precision this data (see Figure 3).

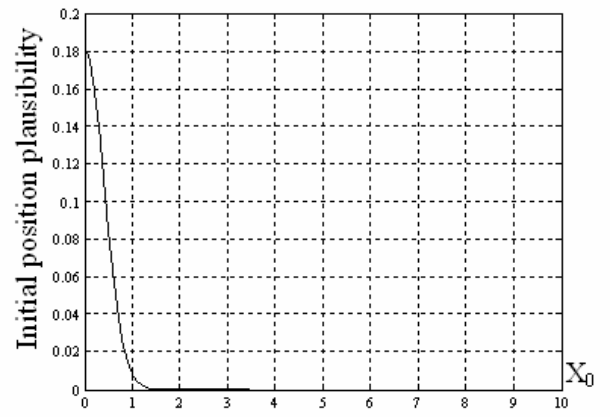

Figure 3
The initial state

The First Situation

The robot has several state transition and no observations are made during this transitions. Simulation results are presented in Figure 4. If we analyze this result the main conclusion is that even the translation value – according to (12) – remains constant, the degree of plausibility has decreased continuously from translation to translation. This means that the degree of trust decrees continuously. This is an obvious situation, because a scientist has already the feeling that using repeatedly a model the degree of confidence will decrease. In this case the benefit is that we can compute this decreasing and of course we can take decisions after these results.

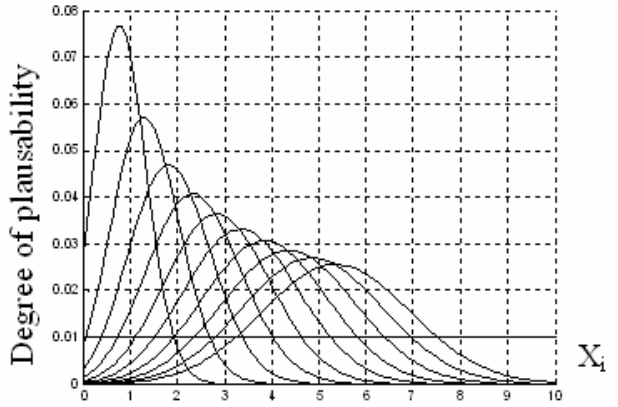

Figure 4
Transition without observation

The Second Situation

The robot performs several observations – without performing any transition. In Figures 5 and 6 where we have presented the results of this simulation we can see that the degree of plausibility increases continuously and converges to value 1 (absolute trust).

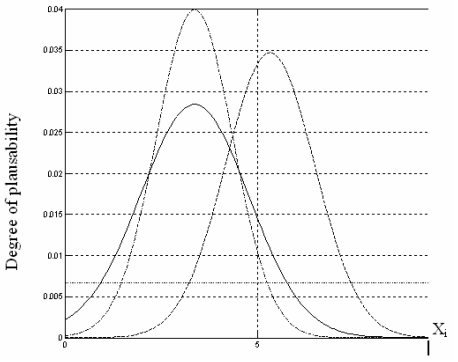

Figure 5
Two observation (.-.- and ---) which starts from the same state (-)

In Figure 5 two particular situations are compared. There are two observations which start from the same state. In the first case, when the observation reproduces the value of the state, we will obtain a bigger rising. At contrary, in the second case there is a difference between the observation and the state. This difference will rule to a smaller degree of true. If we realize several observations which have the same value the degree of true will increase continuously to one (Figure 6).

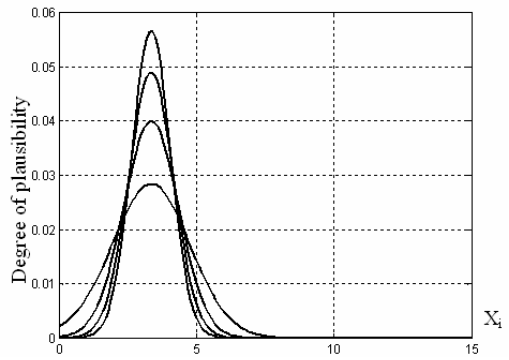

Figure 6
Increasing the plausibility by several observations

The Third Situation

After these results the conclusion is that we can impose a minimum value of trust and perform observations only if we are below of this value. This is a more realistic strategy which is presented in Figure 7.

In Figure 6 the minimum trust value is 0.1. We have started from 0.18 and after five transitions we are below this value. In this moment we have performed an observation which increased the confidence value to 0.22.

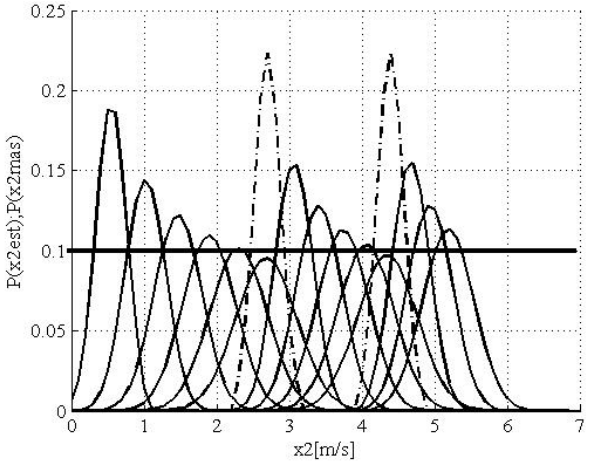

Figure 7
Transitions (-) followed by observations (-.-)

4 The Second Case Study

The second case study proves the ability of the Bayesian method geometry. The problem that we intend to solve is the following:

Problem: If ABC *is an isosceles triangle,* $M \in AC$ *and* $AM = MC$ *then* $BM \perp AC$, a priori we know that if ABC *is an isosceles triangle and* $BM, \neg\perp, AC$ *then* $AM \neq MC$.

If we will rewrite the problem by using the plausibility function we will obtain:

$$p(\perp|\Delta,=) = 1 \quad \text{if} \quad p(=|\Delta,\neg\perp) = 0 \tag{24}$$

where: $p(\perp|\Delta,=)$ is defined like the plausibility that $BM \perp AC$ when we know that ΔABC is isosceles and $AM = MC$;

$p(=|\Delta,\neg\perp)$ is defined like the plausibility that $AM = MC$ when we know that ΔABC is isosceles and $BM, \neg\perp, AC$.

From (2-5) we can write:

$$p(\perp|\Delta,=) = \frac{p(=,\Delta,\perp)}{p(=,\Delta)} = \frac{p(\perp)p(\Delta|\perp)p(=|\perp,\Delta)}{p(\perp)p(\Delta|\perp)p(=|\perp,\Delta) + p(\neg\perp)p(\Delta|\neg\perp)p(=|\Delta,\neg\perp)} = \frac{\alpha\beta\delta}{\alpha\beta\delta + (1-\alpha)\varphi\cdot 0} = 1$$

Some comments are necessary: usually in the first moment will consider that $\alpha, \beta, \delta, \varphi$ are 50%; but after a more careful examination we will realize that these plausibility are very smalls because there are many possibilities that are also plausible. The solution proves that these values are not important.

Conclusions

Present paper consists from two parts. In the first a phenomenological analyzes of AI collocation is performed. The result of this analyze are seven question which intend to deep the understanding of AI. In the second part we tray to answer to the first two questions by presenting the plausible reasoning theory. This theory is proposed in [4], but we have structured it from a new point of view which corresponds to the description from the previous analysis. We consider that the main advantage of this theory consists in fact that it allows epistemological model which contains both inductive and deductive process. The first presented example underlines this aspect. Increasing the plausibility of a sentence by performing observation means to perform the induction. We will underline also two aspects which have been obtained from simulation. We will mention firstly the diminution of the trust, during repeated use of a theoretical model and secondly the possibility to increase the plausibility by performing observations. The second example underlines the possibilities Bayesian method in deduction processes.

References

[1] http://fp6.cordis.lu/fp6

[2] Shinghal R. Formal Concepts in Artificial Inteligence, Chapman & Hall

[3] Rich, E. Artificial Intelligence McGraw – Hill, New York

[4] Nilsson, N., Principles of Artificial Intelligence, Morgan Kaufmann, Los Altos, California

[5] T. Bayes (1763/1958) Studies in the History of Probability and Statistics: IX. Thomas Bayes's Essay Towards Solving a Problem in the Doctrine of Chances. Biometrika 45:296-315

[6] Laplace, P. S. (1795/1951) A Philosophical Essay on Probabilities, Dover

[7] Jaynes, E. T., Probability Theory with Application in Science and Engineering, Washington University 1974

[8] http://yudkowsky.net/bayes/bayes.html Yudkowsky, E., An Intuitive Explanation of Bayesian Reasoning

[9] J. Pearl. Causality: Models, Reasoning, and Inference, Cambridge Univ. Press, New York, 2000

[10] Praladier, C., Navigation intentionnelle d'un robot mobile, Doctoral Thesis of L'INPG 2004

[11] Rescorla, R. A., Wagner, A. R. (1972). A Theory of Pavlovian Conditioning: Variations in the Effectiveness of Reinforcement and Nonreinforcement. In A. H. Black, W. F. Prokasy (Eds.), Classical Conditioning II: Current Theory and Research (pp. 64-99). New York: Appleton-Century-Crofts

[12] Rumelhart, D. E., McClelland, J. L. (1986). Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Cambridge, MA: MIT Press

[13] Elman, J. L., Bates, E. A., Johnson, M. H., Karmiloff-Smith, A. (1996). Rethinking Innateness: A Connectionist Perspective on Development. Cambridge, MA: MIT Press

[14] Schulz, L. E., Gopnik, A. (2004). Causal Learning Across Domains. Developmental Psychology, 40, 162-176

[15] Rogers, T., McClelland, J. (2004). Semantic Cognition: Aparallel Distributed Approach. Cambridge, MA: MIT Press

Improved High Dynamic Range Image Reproduction Method

András Rövid [1, 2], Takeshi Hashimoto [2]

[1] Department of Vehicles and Light-Weight Structure Analysis
Budapest University of Technology and Economics
Bertalan Lajos u. 2, H-1111 Budapest, Hungary
e-mail: rovid@kme.bme.hu

[1, 2] Department of Electrical and Electronics Engineering, Shizuoka University
5-1, 3-chome Johoku, Hamamatsu, 432-8561, Japan
e-mail: tethash@ipc.shizuoka.ac.jp

Abstract: High dynamic range (HDR) of illumination may cause serious distortions and other problems in viewing and further processing of digital images. This paper describes a new algorithm for HDR image creation based on merging images taken with different exposure time. There are many fields, in which HDR images can be used advantageously, with the help of them the accuracy, reliability and many other features of the certain image processing methods can be improved.

Keywords: high dynamic range, multiple exposures, segmentation

1 Introduction

Digital processing can often improve the visual quality of real world photographs, even if they have been taken with the best cameras by professional photographers in carefully controlled lighting conditions. This is because visual quality is not the same thing as accurate scene reproduction. In image processing most of the recently used methods apply a so called preprocessing procedure to obtain images which guarantees – from the point of view of the concrete method – better conditions for the processing. Eliminating noise from the images yields much better results as else.

There are many kinds of image properties to which the certain methods are more or less sensitive [1] [2]. Certain image regions have different features. The parameters of the processing methods in many cases are functions of the image features. The light intensity at a point in the image is the product of the reflectance at the corresponding object point and the intensity of illumination at that point.

The amount of light projected to the eyes (luminance) is determined by factors such as: the illumination that strikes visible surfaces, the proportion of light reflected from the surface and the amount of light absorbed, reflected or deflected by the prevailing atmospheric conditions such as haze or other partially transparent media [3]. An organism needs to know about meaningful world properties, such as color, size, shape, etc. These properties are not explicitly available in the retinal image and must be extracted by visual processing. In this paper we will deal with the reproduction of the image when the high dynamic range of the lightness causes distortions in the appearance and contrast of the image in certain regions e.g. because a part of the image is highly illuminated looking plain white or another is in darkness. High dynamic range (HDR) images enable to record a wider range of tonal detail than the cameras could capture in a single photo.

Dynamic range in photography describes the ratio between the maximum and minimum measurable light intensities. HDR imaging is a set of techniques that allow a far greater dynamic range of exposures than normal digital imaging techniques [4]. HDR capable sensors play an important role in the traffic safety as well, therefore they are important for use with cars because they must operate in dark and bright environments. An HDR sensor, in contrast to a linear sensor, can detect details that bright environments wash out, and it misses fewer details in dark environments [4]. Using HDR techniques in preprocessing phase of the images, the performance of different image processing algorithms can be improved, e.g. corner and edge detectors.

The paper is organized as follows: Section II gives a short overview of the existing principles, Section III describes the basic concept of the algorithm, Section IV introduces the so called detail factor and its estimation, while Section V describes the proposed method more detailed and finally Section VI and Section VII report conclusions and experimental results.

2 Background

There are some existing methods, which main aim is to obtain as detailed image as possible from multiple exposures. For example method in [5] is based on fusion in the Laplacian pyramid domain. The core of the algorithm is a simple maximization process in the Laplacian domain. Wide dynamic range CMOS image sensors play also very important role in HDR imaging [6]. The sensor introduced in [6] uses multiple time signals and such a way extends the dynamic range of the image sensor.

3 The Basic Concept

If the scene contains regions with high luminance values, then to see the details in that highly illuminated regions it is necessary to take a picture with lower exposure time, on the other hand if the scene contains very dark areas, then this exposure time should be higher. In such cases taking just only one image is not enough to capture every detail of the scene, more pictures are needed with various exposure time.

Given N images of the same scene, which were taken using different exposure time. The proposed method combines the given N images into a single HDR image in which each of detail involved in the input images can be found. The main idea of the method is the following. First of all, it is necessary to detect those regions of the input images in which the level of the involved detail is larger then the level of the same region in the other N-1 images. This procedure is performed by segmenting the images into small rectangular regions of the same size. One region can contain many but a limited number of connected local rectangular areas. The output HDR image is obtained by merging the estimated regions together. By the merging not just the contents of the regions have to be merged, but the sharp transitions, which occur between the borders of the regions should be also eliminated. For this purpose smoothing functions can be used. In this paper the Gaussian hump is used as the smoothing function. To each region one Gaussian function is assigned, with center coordinates identical to the center of gravity of the corresponding region. Finally, using the obtained regions and the corresponding Gaussian, blending procedure is applied to obtain the resulted HDR image. The quality of the output can be influenced by several parameters, like the size of the regions, the parameters of the Gaussian and the size of the rectangular areas, which were mentioned at the beginning of this section.

4 Measuring the Level of the Detail in an Image Region

For extracting all of the details involved in a set of images of the same scene made with different exposures, it is required to introduce a factor for characterizing the level of the detail in an image region. For this purpose the gradient of the intensity function was used, corresponding to the processed image and a linear mapping function, which was applied for setting up the sensitivity of the measurement of the detail level. In the followings the description of the estimation of the mentioned factor is introduced.

Let $I(x, y)$ be the pixel luminance at location $[x, y]$ in the image to be processed. Let us consider the group of neighboring pixels which belong to a 3×3 window

centered on [x, y]. For calculating the gradient of the intensity function in horizontal Ix and vertical Iy directions at position [x, y] the luminance differences between the neighboring pixels were used:

$$\Delta I_x = |I(x+1, y) - I(x, y)|, \tag{1}$$

$$\Delta I_y = |I(x, y-1) - I(x, y)|. \tag{2}$$

For the further processing the maximum of the estimated gradient values should be chosen, which solves as the input of the linear mapping function P defined as follows:

$$P(v) = v / I_{\max}, \tag{3}$$

where Imax is the maximal luminance value. For 8 bit grayscale images it equals 255. Let **R** be a rectangular image region of width rw and height rh, with upper left corner at position [x_r, y_r]. The level of the detail inside of the region **R** can be defined as follows:

$$M_D(\mathbf{R}) = \frac{N_e}{r_w r_h} \sum_{i=0}^{r_w} \sum_{j=0}^{r_h} P(r_{ij}), \tag{4}$$

where r_{ij} stands for the maximum of the gradients in horizontal and vertical direction [1], i.e.

$$r_{ij} = \max(\Delta I_x(x_r + i, y_r + j), \Delta I_y(x_r + i, y_r + j)), \tag{5}$$

and N_e represents the number of pixel positions inside the region **R** for which $r_{ij} > 0$. As higher is the calculated M_D value, as detailed is the analyzed region. In the followings we will use this parameter for characterizing the measure of the image detail.

5 Description of the Algorithm for Measuring the Level of the Detail in an Image Region

Let I_k denote the intensity function of the input image with index k, where k=1..N and N stands for the number of images to be processed, each of them taken with different exposure time. Each image contains regions, which are more detailed as the corresponding regions in the other N-1 images. Our goal is to produce an image, which is the combination of the input N images and contains all details involved in them without producing noise. Using such detailed image, the most of the feature detection methods can be improved and can effectively be used even if the lighting conditions are not ideal. The first step of the processing is to divide the pictures into small rectangular areas of same size. Let $w \times h$ be the size of these areas, where w is the width and h the height of the rectangular area (see Fig. 1).

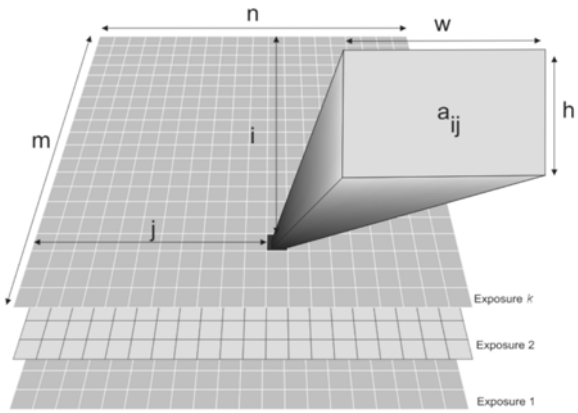

Figure 1
Illustration of the certain exposures and the small rectangles, inside of which the detail factor is estimated

After this division a grid of size nxm is obtained for each input image, where n and m represent the number of rectangular areas in horizontal and vertical direction respectively.

Let a_{ijk} be the area in the *i*th row and *j*th column of the grid corresponding to the image with index k (see Fig. 1). Let **D** be the matrix of highest detail factors, which element in ith row and *j*th column stands for the index of the image having the largest detail factor inside of the area a_{ij} among all input images (exposures), i.e.

$$\forall k \in \{1,..,N\} \wedge k \neq s, s = d_{ij} : M_{\mathbf{D}}(a_{ijs}) \geq M_{\mathbf{D}}(a_{ijk}), \tag{6}$$

where d_{ij} stands for the element in the *i*th row and *j*th column of matrix **D**. Now, we have the matrix **D**, which contains the indices of the input images with largest detail factor corresponding to the certain rectangular areas. Using this matrix we can easily find the areas with largest detail factor and merge them together. If we take into account the processing time necessary for such merging – which involves also the smoothing process for eliminating the sharp should be reduced to obtain the output in a desirable time interval. Increasing the size of the rectangular areas reduces the processing time, but the quality of the resulted HDR image will be lower. The reason is that large rectangular areas can fall with high probability onto such image positions, where a part of a concrete area can contain very detailed and non-detailed subareas, as well.

In the followings we will describe how to avoid such effects and how to increase the processing time by maintaining the quality of the output. As solution we can create a predefined number of groups using the small rectangular areas. By the creation the distance between the centers of the areas with the highest detail level corresponding to the same input image is used.

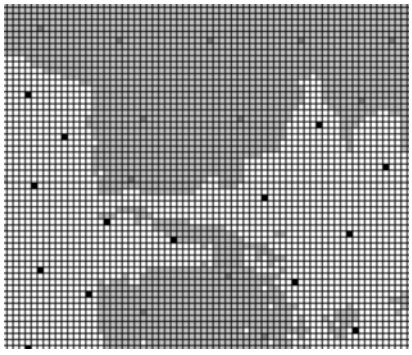

Figure 2

Illustration of the groups and their centers of gravities (black and dark gray rectangles). The figure illustrates a situation when two input images are given. The white region represents those areas of the first input image, which have the largest detail factor comparing to the same area in the second image. On the other hand the gray region illustrates those areas, which detail factor is the largest comparing to the same area in the first image

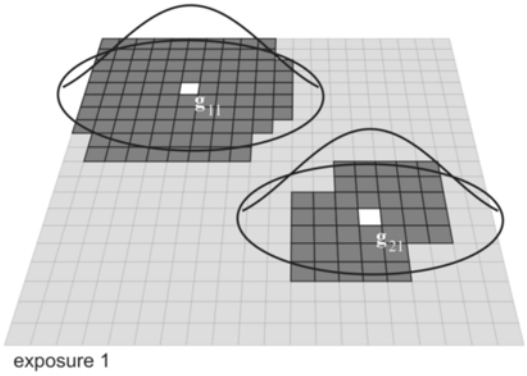

Figure 3

Illustration of an image segmented into groups. The dark gray areas are those which detail factor is the largest among all images. The white squares are illustrating the centers of gravities corresponding to the certain groups. Furthermore the Gaussian humps centered at g_{ij} positions can be seen. The small rectangles are the areas a_{ij}.

There is a lot of different series of such groups. We can form a group for example as follows: Suppose that we want to group those areas of the input image with index k, for which $d_{ij}=k$. First we take an arbitrary area a_{ij} satisfying the equation $d_{ij}=k$ as the first element of a group.

As next step we are searching for such areas and add them to this group, which distance to the center of the first element of the group is below a predefined threshold value. If there are no other areas then we can form another group from the remaining areas using the same procedure.

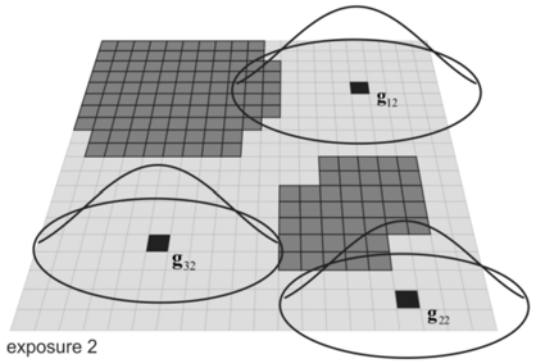

Figure 4
Illustrating a similar situation like in Fig. 3, but in this case the exposure 2 can be seen

Finally a set of groups is obtained, which centers of gravity are uniformly distributed among the areas of the kth image satisfying the equation $d_{ij}=k$. The whole process is repeated for each input image. An illustration of the result is shown in Fig. 2.

Let L_k be the number of the created groups of the input image with index $k=1..N$ and let $g_{pk} = [g_{pkx}, g_{pky}]$ denote the horizontal and vertical coordinate of the center of gravity of the pth group of the image with index k, where $p = 1..L_k$ (see Fig. 3-4). By the merging we will use the fuzzy theory in combination with the blending functions and theory [7][8][9][10]. First we have to choose a function, which is continuous and differentiable (C^1, C^2). Furthermore this function should have the maximal value at an estimated center of gravity g_{pk}, and is decreasing proportional to the distance to g_{pk}. These requirements are fulfilled for example by the Gaussian hump. If we place a Gaussian function over each gpk position, we can combine the certain groups of each image without producing sharp borders between the transitions. Before combining the groups, so called membership functions have to be constructed having the following properties.

$$\sum_{k=1}^{N}\sum_{p=1}^{L_k}\mu_{pk}(x,y)=1, \tag{7}$$

where μ_{pk} stands for the membership function corresponding to the group with center of gravity g_{pk}. In other words, using the membership functions, we can calculate the membership values of an arbitrary pixel position in the estimated groups. We can refer to these groups also as fuzzy sets [7]. Taking into account the above described conditions and requirements, the membership values pk can be defined as follows [10]:

$$\mu_{pk}(x,y)=\frac{e^{-\left(\frac{(x-g_{pkx})^2}{2\sigma_x^2}+\frac{(y-g_{pky})^2}{2\sigma_y^2}\right)}}{\sum_{u=1}^{N}\sum_{v=1}^{L_u}e^{-\left(\frac{(x-g_{vux})^2}{2\sigma_x^2}+\frac{(y-g_{vuy})^2}{2\sigma_y^2}\right)}}, \tag{8}$$

where p and k represent the group index and the input image index respectively, σ_x and σ_y stand for the standard deviation of the Gaussian function. Let G_{pk} describe the group or fuzzy set with center of gravity g_{pk}. Now we know the fuzzy sets and their membership functions. The next step is to construct the so called fuzzy rules of the form:

IF (x is from A) **THEN** $y = B$.

The meaning of this structure is the following: If the element x is the member of fuzzy set A with a non-zero membership value then the output value y is from fuzzy set B with a membership proportional to the membership value of the element x in fuzzy set A. In our case a fuzzy set G_{pk} will correspond to fuzzy set A, y to the output intensity and B to the intensity function of the images corresponding to G_{pk}. Our fuzzy rules have the following form:

IF (q is from G_{11}) THEN $I_{out} = I_1$

IF (q is from G_{21}) THEN $I_{out} = I_1$

...

IF (q is from $G_{L_1,1}$) THEN $I_{out} = I_1$

IF (q is from G_{12}) THEN $I_{out} = I_2$

IF (q is from G_{22}) THEN $I_{out} = I_2$

...

IF (q is from $G_{L_2 2}$) THEN $I_{out} = I_2$

. . .

IF (q is from G_{1N}) THEN $I_{out} = I_N$

IF (q is from G_{2N}) THEN $I_{out} = I_N$

...

IF (q is from $G_{L_N N}$) THEN $I_{out} = I_N$

where $\mathbf{q} = (x, y)$ is an arbitrary point in the image domain and I_{out} stands for the intensity of the output pixel at location $\mathbf{q}$. After evaluation of the fuzzy rules the output can be written as follows:

$$I_{out}(x,y) = \sum_{k=1}^{N}\sum_{p=1}^{L_k} \mu_{pk}(x,y) I_k(x,y). \tag{9}$$

The output luminance can be influenced by changing the threshold for the distance between the centers of the areas, i.e. the size of the groups. The standard deviation also enables to influence the output HDR image. As smaller is the standard deviation as higher influence the regions have with low detail level onto the result. Using such detailed image the edges can be also effectively extracted and advantageously used by further processing, e.g. object recognition, scene reconstruction, etc.

Conclusions

In this paper a new gradient based approach for extracting the image details was introduced, which is using multiple exposure images of the same scene as input data. The image parts involving the highest detail are choosen from each input image. Finally these parts are blended together using Gaussian blending functions. The proposed method can be applied for color images, as well. In this case the whole procedure for each color component has to be applied. As result a HDR image is obtained. The method can advantageously be applied, when the luminance properties are not appropriate, and each detail of the scene can not be captured using one exposure time only.

Examples

In this example the width and the height of the rectangular areas was chosen to be 5 pixels. Deviations σ_x=120 and σ_y=120 in this example. In Figs. 5 and 6 an overexposed and an underexposed image can be seen. Fig. 7 represents the resulted HDR images using the proposed method.

Acknowledgement

This work was supported by the Hungarian National Science Research Found (OTKA) under grants T048756 and T042896.

References

[1] F. Russo, “Fuzzy Filtering of Noisy Sensor Data,” in In Proc. of the IEEE Instrumentation and Measurement Technology Conference, Brussels, Belgium, 4-6 June 1996, pp. 1281-1285

[2] F. Russo, “Recent Advances in Fuzzy Techniques for Image Enhancement,” IEEE Transactions on Instrumentation and Measurement, 1998, Vol. 47, No. 6, pp. 1428-1434

[3] E. Adelson, A. Pentland, “The Perception of Shading and Reflectance,” in In D. Knill and W. Richards (eds.), Perception as Bayesian Inference, New York: Cambridge University Press, 1996, pp. 409-423

Figure 5
The input overexposed image

Figure 6
The input underexposed image

Figure 7
The resulted image after applying the proposed method

[4] L. S. Y. Li, E. Adelson, "Perceptually-based Range Compression for High Dynamic Range Images," Journal of Vision, 2005, Vol. 5, No. 8, p. 598

[5] A. B. R. Rubinstein, "Fusion of Differently Exposed Images," in Final Project Report, Israel Institute of Technology, 2004, p. 14

[6] S. K. Y. W. M. Sasaki, M. Mase, "A Wide Dynamic Range Cmos Image Sensor with Multiple Exposure Time Signals and Column-Parallel Cyclic a/d Converters," in IEEE Workshop on Charge-Coupled Devices and Advanced Image Sensors, 2005

[7] B. Y. George J. Klir, "Fuzzy Sets and Fuzzy Logic: Theory and Applications," in Prentice Hall PTR, Munich, Germany, 1995, p. 592

[8] H. Bidasaria, "Defining and Rendering of Textured Objects through the Use of Exponential Functions," Graphical Models and Image Processing, 1992, Vol. 54, No. 2, pp. 97-102

[9] L. Piegl, W. Tiller, "The Nurbs Book," in Springer-Verlag 1995-1997 (2nd ed.), 1995, p. 646

[10] D. Breen, W. Regli, M. Peysakhov, "B-splines and Nurbs," in Lecture, Geometric and Intelligent Computing Laboratory, Department of Computer Science, p. 42

Chaos and Natural Language Processing

Marius Crisan

Department of Computer and Software Engineering
Polytechnic University of Timisoara
V. Parvan 2, 300223 Timisoara, Romania
Tel.: +40256403254, E-mail: crisan@cs.upt.ro, http://www.cs.upt.ro/~crisan

Abstract: The article explores the possibility to construct a unified word feature out of the component features of letters. Each letter is modeled by a different attractor and finally embedded in a quadratic iterated map. The result is the word feature that can account for the meaning extraction process of language understanding. This is a new approach in natural language processing based on the deterministic chaotic behavior of dynamical systems.

1 Introduction

There is an increased interest in the modern era for developing techniques for both speech (or character/word sequences) recognition and synthesis. Natural language processing (NLP) provides those computational techniques that process spoken and written human language. An important class of methods for language recognition and generation is based on probabilistic models, such as N-grams model, Hidden Markov and Maximum Entropy model [1]. Given a sequence of units (words, letters, morphemes, sentences, etc.) these models try to compute a probability distribution over possible labels and choose the best label sequence. Another approach in NLP is to use neural networks, in particular self-organizing maps of symbol strings [2]-[4]. However, an important challenge for any NLP approach which may hinder its success is dealing with the nonlinear character of language phenomenon. Starting from the premise that natural language phenomena can be viewed as a dynamical system the purpose of this work is to investigate the possibility of modeling words/characters by a chaotic attractor.

2 Meaning as Wholeness in Dynamical Systems

We may consider consistently with other theories of language that the notion of *word* is the constituent element of a sentence (utterance). We might, at first, also

consider in a general formalization that a word is any sound-sequence that possesses the property of inflection. Normally, each word takes either a verbal, i.e., conjugational inflection, in which case it is called a verb, or a nominal, i.e., declensional inflection, in which case it is of a non-verbal category (substantives, adjectives, participles, etc.). All the other words which do not have declensional inflections, such as prepositions, may be considered to possess invariant inflection. However, to classify words only in terms of their inflection property is an incomplete task, and does not seem to help much in explaining how the meaning as structured information is conveyed by a sentence. Therefore, I suggest the employment of semantic criteria in defining the notion of word. According to such a view, a word is the meaning-bearing element of a sentence. The semantic criterion determines the minimum sequence length of the phonemes which convey a meaning. Thus, words may vary in complexity, from the shortest meaning-bearing ones to the more complex compound words. Based only on meaningful words, we may define, in general terms, a sentence as being a cluster of words capable to generate a cognitive meaning in an ideal receiver (hearer/reader). This cognition is a result of a reaction mechanism triggered by the series of words in the sentence.

An ideal receiver is qualified by the 'capacity' to extract meaning from a sentence. This capacity can be described by the cognition of four cognitive properties that have been assigned by the transmitter (speaker/writer) to a sentence: (1) semantic competency, (2) expectancy (syntactic/semantic), (3) contiguity in space and time, and (4) transmitter's intention [5]. These cognitive properties are the requirements for defining a grammatical and meaning-bearing sentence. A sentence is said to have semantic competency when the objects denoted by the respective words are compatible one to another. For instance, the sentence '*He sees the light*.' is grammatically acceptable, and has semantic competency, while the sentence '*He hears the color*.' even if it is grammatically acceptable, lacks semantic competency. Semantic expectancy refers to the capacity of an ideal receiver to infer the meaning of an incomplete sentence (utterance). Syntactic expectancy refers to the syntactic property *x* which has o be assigned to a sentence *s* when it is not grammatical, in order to make it suitable to transmit the meaning. This expectancy is measured by the predictor of the entropy of the entropic source. Contiguity is the property which imposes the absence of any unnecessary spatial (in written text) or temporal (in uttered) interval between the words of a sentence.

In a previous work [6], in defining meaning as something that must have a finite description, I introduced the concept of undivided meaning-whole (UMW). This is conceived as structured information which exists internally at the agent's information level. Even if UMW is a unitary information structure, it is describable rationally in terms of cognitive semantic units. These semantic units are the generating principle of producing the sequence of uttered words.

When an agent wants to communicate, it begins with the UMW existing internally in its knowledge base. When words are uttered producing different sounds in

sequence, it appears only to have differentiation. Ultimately, the sound sequence is perceived as a unity or UMW and only then the word meaning, which is also inherently present in the receiver's mind, is identified.

The above described capacity of the receiver to extract meaning from series of words led to another assumption, that the whole word/sentence meaning has to be inherently present in the mind of each agent. Thus, it can be explained how it is possible the UMW to be grasped by the hearer even before the whole sentence has been uttered. The sounds which differ from one another because of difference in pronouncement cause the cognition of the one changeless UMW without determining any change in it. Sometimes, reasoning may have to be applied to the components of the sentence so that the cognition is sufficiently clear to make possible the perception of the meaning-whole. It appears that the unitary word-meaning is an object of each person's own cognitive perception. When a word, such as 'tree' is pronounced or read there is the unitary perception or simultaneous cognition of trunk, branches, leaves, fruits, etc. in the receiver's mind. Communication (verbal or written) between people is only possible because of the existence of the UMW, which is potentially perceivable by everybody and revealed by words' sounds or symbols.

The concept of UMW is consistent with a more general view, suggested by Bohm in [7], regarding the possibilities for wholeness in the quantum theory to have an objective significance. This is in contrast with the classical view which must treat a whole as merely a convenient way of thinking about what is considered to be in reality nothing but a collection of independent parts in a mechanical kind of interaction. If wholeness and non-locality is an underlying reality then all the other natural phenomena must, one way or another, be consistent with such a model. Natural language generation and understanding is a phenomenon that might be modeled in such a way. UMW is like 'active information' in Bohm's language, and is the activity of form, rather than of substance. As Bohm puts it clearly [7], '…when we read a printed page, we do not assimilate the substance of the paper, but only the forms of the letters, and it is these forms which give rise to an information content in the reader which is manifested actively in his or her subsequent activities.' But, similar so called mind-like quality of matter reveals itself strongly at the quantum level. The form of the wave function manifests itself in the movements of the particles. From here, a new possibility of modeling the mind as a dynamical system is considered.

In line with Kantian thought, in [8] we find a similar insight, as above, regarding the linguistic apprehension. This is the interplay of two factors of different levels: (1) the empirical manifold of the separate letters or words and (2) the *a priori* synthesis of the manifold which imparts a unity to those elements which would otherwise have remained a mere manifold.

According to this kind of observations it appears motivated to use the concept of manifold for modeling the mind as the seat of language generation and

understanding. Manifolds are defined as topological spaces possessing families of local coordinate systems that are related to each other by coordinate transformations pertaining to a specific class. They may be seen also as the multidimensional analogue of a curved surface. This property seems suitable to represent both the natural language constraints and semantic content of linguistic objects.

Usually, a dynamical system is a smooth action of the reals or the integers on a manifold. The manifold is the state space or phase space of the system. Having a continuous function, F, the evolution of a variable x can then be given by the equation:

$$x_{t+1} = F(x_t). \tag{1}$$

The same system can behave either predictably or chaotically, depending on small changes in a single term of the equations that describe the system. Equation (1) can also be viewed as a difference equation ($x_{t+1} - x_t = F(x_t) - x_t$) and generates iterated maps. An important property of dynamical systems is that even very simple systems, described by simple equations, can have chaotic solutions. This doesn't mean that chaotic processes are random. They follow rules, but even the simple rules can produce amazing complexity. In this regard, another important concept is that of an attractor. An attractor is a region of state space invariant under the dynamics, towards which neighboring states in a given basin of attraction asymptotically approach in the course of dynamic evolution. The basin of attraction defines the set of points in the space of system variables such that initial conditions chosen in this set dynamically evolve to a particular attractor. It is important to note that a dynamical system may have multiple attractors that may coexist, each with its own basin of attraction [9]. This type of behavior is suitable for modeling self-organizing processes, and is thought to be a condition for a realistic representation of natural processes.

One example of such an approach is the topological feature map proposed by Kohonen [10], [11] for the projection of high dimensional pattern data into a low-dimensional feature space. The process of ordering an initial random map is called in this approach self-organization. The result is the topological ordering of pattern projections, or in other words the self-organizing map (SOM). Each input dimension is called a feature and is represented by an N-dimensional vector. Each node in the SOM is assigned an N-dimensional vector and is connected to every input dimension. The components or weights of this vector are adjusted following an unsupervised learning process. First, it is found the winning node, i.e, the node whose weight vector shows the best match with the input vector in the N-dimensional space. Next, all weight vectors in the neighborhood in the direction given by the input vector are adjusted. This process requires many iterations until it converges, i.e., all the adjustments approach zero. It begins with a large neighborhood and then gradually reduces it to a very small neighborhood. Consequently, the feature maps achieve both ordering and convergence properties,

and offer the advantages, of reducing dimensions and displaying similarities. However, SOM solutions (and neural networks in general) are yet in the need for improvement. For instance, in [12], an important problem for SOM is discussed. In order to obtain a realistic speech projection, the problem is to find a hypercubical SOM lattice where the sequences of projected speech feature vectors form continuous trajectories. In another work [13], both SOM and a supervised multilayer perceptron were used for bird sounds recognition. The conclusion was that although the tested algorithms proved to be quite robust recognition methods for a limited set of birds, the proposed method cannot beat a human expert listener.

On the other hand, the unexplored domain of dynamical systems and chaos theory may offer promising perspectives in modeling natural processes, and NLP might be one of them.

3 Attractor-based Word Modeling

In quantum experiments, when particles interact, it is as if they were all connected by indivisible links into a single whole. The same behavior is manifested by the chaotic solutions in an attractor, as we will see in this section. In spite of the apparent random behavior of these phenomena, there is an ordered pattern given by the form of the quantum wave (or potential) in the former case, and by the equations of the dynamic system in the latter.

Let's consider the simplest case of the quadratic iterated map described by the equation:

$$x_{t+1} = a_1 + a_2 x_t + a_3 x_t^2 \tag{2}$$

Even if it is so simple, it is nonlinearly stable and can manifest chaotic solutions. The initial conditions are drawn to a special type of attractor called a strange attractor. This may appear as a complicated geometrical object which gives the form of the dynamic behavior.

In nonlinear dynamics the problem is to predict if a given flow will pass through a given region of state space in finite time. One way to decide if the nonlinear system is stable is to actually simulate the dynamics of the equation. The primary method in the field of nonlinear dynamic systems is simply varying the coefficients of the nonlinear terms in a nonlinear equation and examining the behavior of the solutions. The initial values of the components of the model vector, $m_i(t)$, were selected at random in a process of finding a strange attractor. Strange attractors are bounded regions of phase space corresponding to positive Lyapunov exponents. We found more than 100 attractors. In Table I we presented a list of several coefficients along with the Lyapunov exponent for which the attractors were found by random search. The initial condition x_0 was selected in

the range 0.01 – 1 and lies within the basin in many cases. The Lyapunov exponent is computed in an iterated process according to the following equation [14], [15]:

$$LE = \Sigma \log_2 |a_2 + 2a_3x_t| / N \quad (3)$$

The sum is taken from a value of $t = 1$ to a value of $t = N$, where N is some large number.

Table 1

The coefficients values and the Lyapunov exponent for 25 attractors of (2)

Cur. No.	a_1	a_2	a_3	*LE*
1	1.2	-1	-1	0.4235
2	1.2	-0.2	-1.1	0.1198
3	1.1	-1.2	-0.8	0.3564
4	1.1	-1	-0.6	6.6073
5	1.1	-0.6	-1	0.1443
6	1	-0.7	-1.1	0.2512
7	0.9	-1.1	-1.1	0.3571
8	0.9	-1.1	-0.8	0.256
9	0.8	-1.2	-1.2	0.411
10	0.8	-0.9	-1	0.1383
11	0.7	-1.2	-0.8	0.2001
12	0.7	-1.1	-1.2	0.3029
13	-1.2	-1.2	0.7	0.2918
14	-1.2	-0.9	0.8	0.2793
15	-1.2	-0.6	1	0.2662
16	-1.1	-0.8	1	0.286
17	-1.1	-1	0.9	0.3054
18	-1	-1	0.7	0.1209
19	-0.8	-1.1	1.1	0.3047
20	-0.8	-1.1	0.7	6.9382
21	-0.7	-1	1	0.1248
22	-0.7	-1.2	1	0.285
23	-0.6	-1.2	1.2	0.2801
24	-0.5	-1.1	1.2	0.1375
25	-0.4	-1.2	1.2	0.1344

LE gives the rate of exponential divergence from perturbed initial conditions. If the value is positive (for instance, greater than 0.005) then there is sensitivity to initial conditions and a strange attractor can manifest. If the solution is chaotic, the successive iterates get farther apart, and the difference usually increases exponentially. The larger the *LE,* the greater is the rate of exponential divergence, and the wider the corresponding separatrix of the chaotic region. If *LE* is negative,

the solutions approach one another. If *LE* is 0 then the attractors are regular. They act as limit cycles, in which trajectories circle around a limiting trajectory which they asymptotically approach, but never reach.

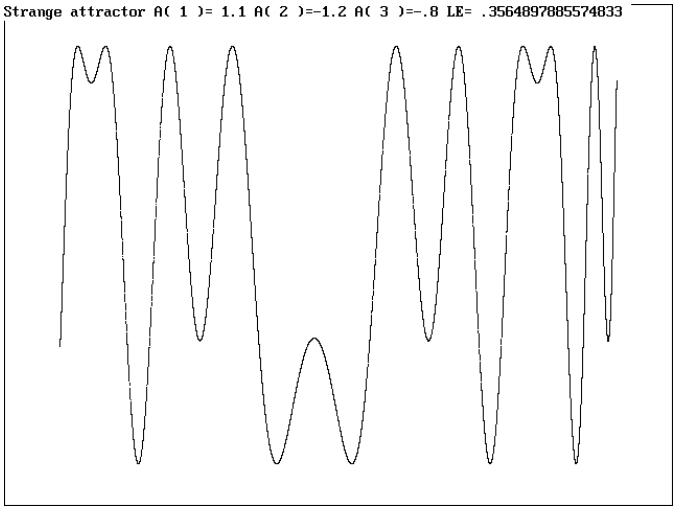

Figure 1
Quadratic iterated map of (2)

It's interesting to analyze in more details the behavior of an attractor. The idea of the self-organizing maps is to project the *N*-dimensional data into something that is better understood visually. A similar idea we follow in constructing iterated maps. It is convenient to plot the values in the iterated process versus their fifth previous iterate for a more suggestive aspect. In Fig. 1 it is presented the iterated map for the strange attractor No. 3. A remarkable property of the chaotic solutions, as noted above in connection with quantum, is the 'ballet-like' behavior as iterations progress. Each new dot on the map, representing the solution x_{t+1}, appears in a random position but orderly following the attractor's form.

In Fig. 2 it is shown the same attractor only after a few iterates (2000). It can be seen the sparse distribution of dots but along with the ordered path. This type of behavior is similar with the quantum phenomena, such as the distribution of photons along the interference pattern lines in the two slit interference experiment, when the photons are emitted in series one after the other. This is also akin to the quality of the perception act (word meaning). It is observed that a word meaning is at first perceived vaguely and then more and more clearly. Thus, through the process of repeated perception or iterations finally the meaning is revealed. We may suggest, therefore, that meaning can be mathematically modeled as a basin of attraction.

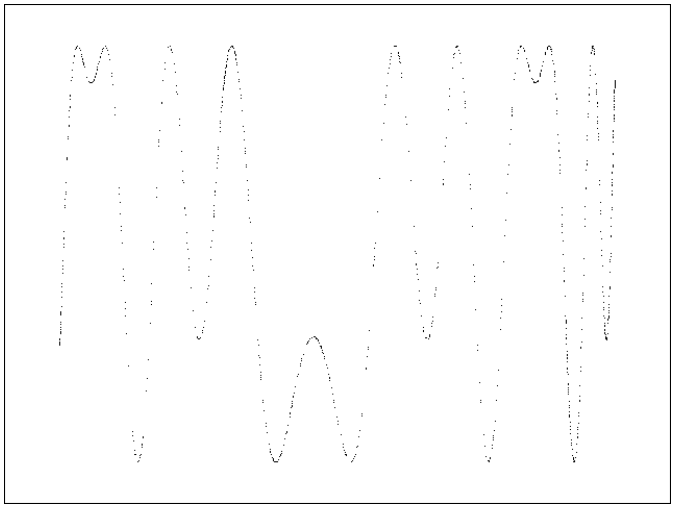

Figure 2

Quadratic iterated map of (2) after 2000 iterates. Note the sparse distribution of dots along the regular pattern of the strange attractor.

Another interesting property is the symmetry of a_1 and a_3 and the corresponding iterated map. Considering again the strange attractor $a_1 = 1.1$, $a_2 = -1.2$, $a_3 = -0.8$, a symmetric behavior can be obtain for the values $a_1 = -1.1$, $a_2 = -1.2$, $a_3 = 0.8$ as in Fig. 3.

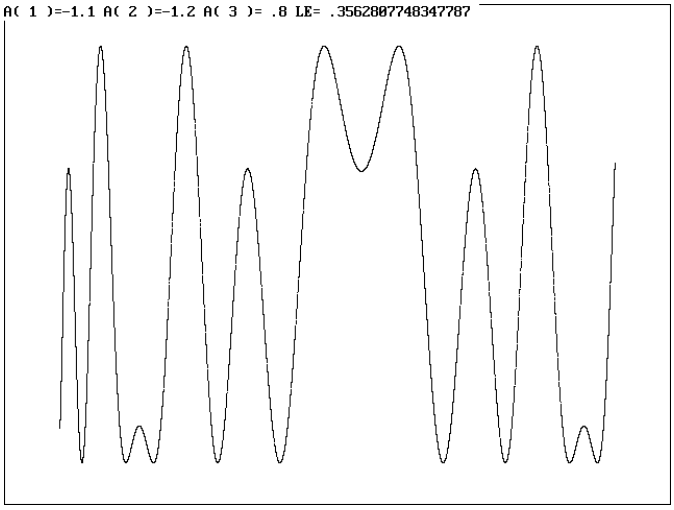

Figure 3

The symmetric quadratic iterated map of Fig 1, obtained by inversing the sign of a_1 and a_3

There is a huge possibility to obtain other attractors by tuning the values of the coefficients. The shape of the attractor changes smoothly with small variations of the coefficients. Even if the interval of variation is rather small, visible changes in the shape of the map can be obtained. For instance, if $a_1 = 1.02$ the value of *LE* is 0.09 and the limit cycles can be observed as the attractor becomes regular. If $a_1 = 1.3$ regular oscillations are manifested.

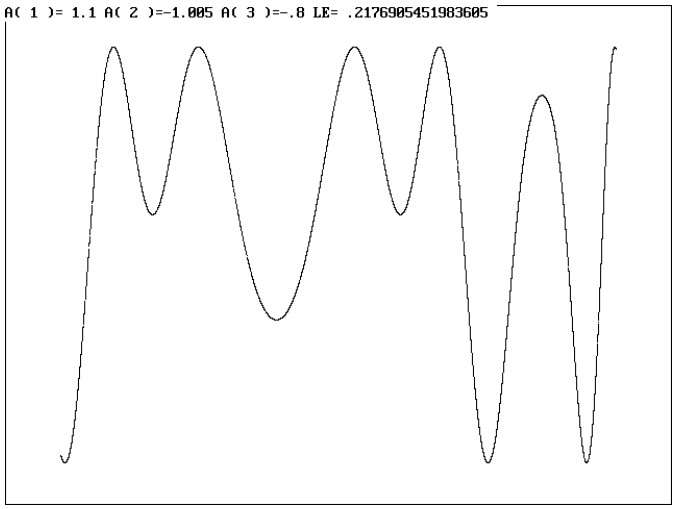

Figure 4
The completely changed quadratic iterated map of (2), obtained for $a_2 = -1.005$

An important change in shape can be obtained by modifying a_2. The value for a_2 always has to be negative for a bounded behavior. For $a_2 = -1.005$ the shape of the map is drastically changed as shown in Fig. 4.

4 Language Recognition

One widely used method to classify an object is to measure its features (characteristic properties). In general the features that are to be observed depend on the specific problem one has to solve. In language recognition, we deal with several kinds of features such as graphological, phonological, statistical, syntactic, lexical, semantic, and pragmatic. Graphological features are for instance letter positions and word shape. Phonological features are considered as the distinctive features from which phonemes can be constructed [16]. The syntactic features are present in the construction of words and sentences, and are part of speech tags and various components from a parse tree. Statistical features exploit the fact that more frequently occurring words are more familiar and hence more easily recognized. These features may be the frequency of occurrence of letters, letter

pairs and triplets, the average word length, the ratio of certain characters, word endings, consonant congestion [17], etc. Lexical features are used to represent the context. They consist of unigrams, bigrams (a pair of words that occur close to each other in text and in a particular order), and the surface form of the target word which may restrict its possible senses [18]. Semantic features indicate the meaning of words, and are usable for disambiguation of words in context. Pragmatic features are based on how the words are used. In general, the above mentioned features are tried to be described by morphology using the concept of morphemes or the constituent parts of words.

Irrespective of the feature's nature, the result of feature extraction or measurement is a set described as an **n**-dimensional feature vector associated with the observed object, which can thus be represented as a point in the **n**-dimensional feature space. Next, a classifier will assign the vector to one of several categories. While the use of features has a central place in pattern classification [19], the design and detection of features in natural language remains a difficult task because of language high complexity and the lack of a unitary theory.

The analysis in the previous section revealed the fact that attractors offer dynamic properties that can map in a continuous manner the feature vectors according to some input patterns. Considering the assumption of UMW, the goal is to construct a unified word feature that might account for the word meaning. I propose a possible non-linear many-to-one mapping from a conventional feature space to a new space constructed so that each word has a unique feature vector. Let's consider the simpler case of a 3-dimensional feature vector characterizing a letter. The vector for a generic letter 'A' is defined by the values $\boldsymbol{a} = [a_1, a_2, a_3]$, and similarly for the generic letters 'B' and 'C' the vectors are $\boldsymbol{b} = [b_1, b_2, b_3]$, and $\boldsymbol{c} = [c_1, c_2, c_3]$ respectively. The letter feature of 'A' results in an iterated process as

$$A_{t+1} = a_1 + a_2 A_t + a_3 A_t^2, \tag{3}$$

starting from an initial condition A_0.

Similar equations result for the letter features of 'B' and 'C', with the initial conditions B_0 and C_0 respectively, as the following:

$$B_{t+1} = b_1 + b_2 B_t + b_3 B_t^2, \tag{4}$$

$$C_{t+1} = c_1 + c_2 C_t + c_3 C_t^2. \tag{5}$$

Based on letters features, for each letter in a word (for instance with length 3) a unified feature vector $W = [A, B, C]$ can be constructed and mapped to the three coefficients of an equation of type (2). The result is of the following form:

$$W_{t+1} = A_t + B_t W_t + C_t W_t^2. \tag{6}$$

Eq. (6) is computed starting from an initial condition W_0 and manifests a chaotic deterministic behavior for a proper combination of the coefficients A, B, and C. In Fig. 5 it is presented the iterated map of (6) for the input vectors $\boldsymbol{a} = [0.8, -1.2, -$

0.9], $\boldsymbol{b}$ = [–1, –0.9, 1.1], and $\boldsymbol{c}$ = [1.1, –1.2, –0.8] after 5000 iterations, and the initial condition for all parameters of value 0.01. In order to have a suggestive view of the unified feature space and observe its internal structure, the values were plotted versus their third previous iteration. Also, the values of W were bounded to 10^8 for a convenient screening. The same sparse distribution of dots along the regular pattern of the feature space, typical for deterministic chaos, can be observed as the iterations progress.

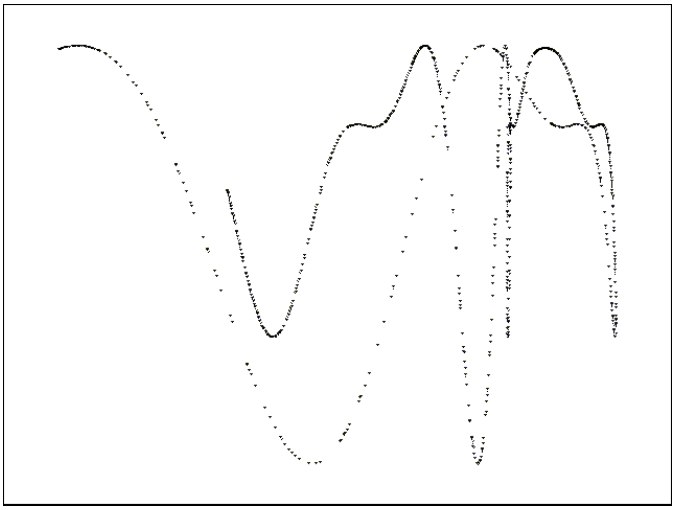

Figure 5
The chaotic deterministic behavior of (6) for the letter feature vectors **a** = [0.8, –1.2, –0.9], **b** = [–1, –0.9, 1.1], and **c** = [1.1, –1.2, –0.8]

Each valid word of length 3 will determine a corresponding iterated map. Small variations in the input will be tolerated and recognized with the same meaning but other illegal combinations will be rejected. For instance, in Fig. 6 we can see the feature space for a rather consistent deformation of the input vectors $\boldsymbol{b}$ = [–1.3, –0.6, 1.3] and $\boldsymbol{c}$ = [.9, –1.3, –1].

Comparing the feature spaces of Fig. 5 and Fig. 6 we can observe the vague resemblance between the two, and after a closer examination we can identify in fact a similar chaotic pattern. This means that the meaning was conserved even if some visible alterations affected two of the letter features. If the changes are more dramatic we expect a completely different pattern or even an unbound behavior. This indicates the lack of properties for a meaningful word. In Fig. 7 it is presented the case where the vectors **a** and **c** swaped their contents. This means another word where the first and last letters are interchanged.

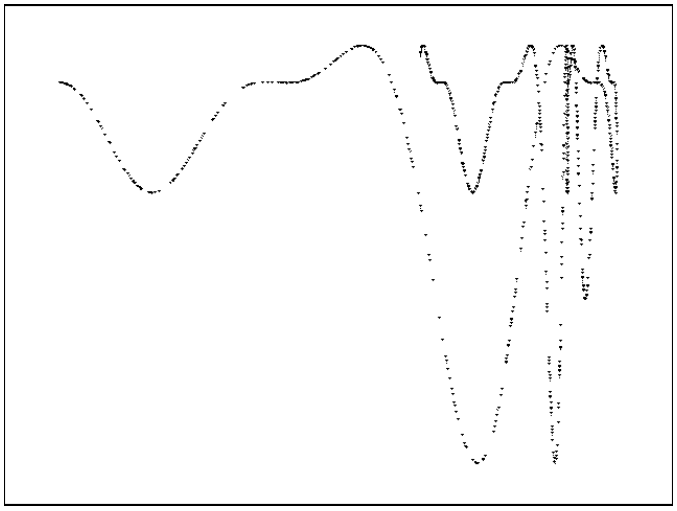

Figure 6
The feature space of (6) for **a** = [0.8, –1.2, –0.9], **b** = [–1.3, –0.6, 1.3], and **c** = [.9, –1.3, –1]. Note the vague resemblance with Fig. 5.

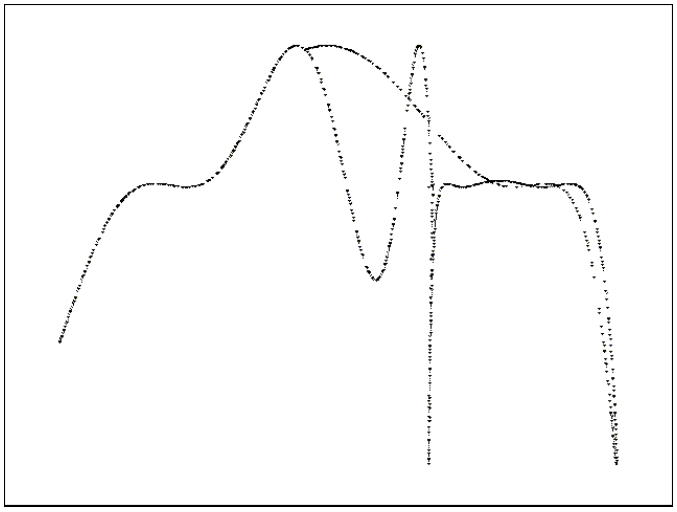

Figure 7
The chaotic deterministic behavior of (6) for the letter feature vectors **a** = [1.1, –1.2, –0.8], **b** = [–1, –0.9, 1.1], and **c** = [0.8, –1.2, –0.9]

A completely different pattern is obtained comparing to Fig. 5. Of course, depending on the classifier conventions, the pattern can be meaningful or not. In any case, it represents the unique feature vector for that word construction.

For words with higher length, higher-order iterated maps can be used. The proposed approach can be extended for a whole sentence. In this case, the unified feature vector of the sentence is constructed based on the features of the individual component words. This will be the UMW equivalent of the whole sentence.

Conclusions

Our purpose was to study the possibility of using dynamical systems in modeling natural language processing. We started from the premise of UMW and the observation facts of language apprehension and noted a similitude with the chaotic behavior of dynamical systems. The attractor behavior as was studied for the quadratic iterated maps seems to be robust enough to model the feature vectors formed for each word of length 3 in the dictionary. The unified word feature vector is obtained by a many-to-one mapping, starting from the component letters, and bears the unique information structure of the word meaning. Slight variations in the input feature vectors of the component letters are tolerated, without major changes of the pattern in feature space structure. This is an indication of meaning preservation in the case of noise. The chaotic deterministic behavior of the patterns in the feature space may account for meaning recognition process after a series of repeated perceptions. After enough iterations (or repeated perception) the attractor shape is recognized and consequently the corresponding meaning. The present work may be continued in the future by constructing the unified feature vector at the sentence level.

References

[1] D. Jurafsky, J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice-Hall, 2000

[2] T. Kohonen, "Self-organizing maps of symbol strings", Report A42, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1996

[3] T. Kohonen, P. Somervuo, "Self-organizing maps of symbol strings with application to speech recognition", in Proc. of Workshop on Self-Organizing Maps (WSOM'97), pp. 2-7, Espoo, Finland, 1997

[4] T. Honkela., "Self-Organizing Maps in Natural Language Processing", Espoo, Finland, 1997

[5] B. K. Matilal, *Logic, Language and Reality*, Motilal Banarsidass Publ., Delhi, 1997

[6] M. Crisan, "Meaning as Cognition," *Proceedings of the I International Conference on Multidisciplinary Information Sciences and Technologies-InSciT2006*, Merida, Spain, 2006, pp. 369-373

[7] D. Bohm, "A new theory of the relationship of mind and matter," *Philosophical Psychology*, Vol. 3, No. 2, 1990, pp. 271-286

[8] H. G. Coward, *The Sphota Theory of Language*, Motilal Banarsidass Delhi, 3rd ed. 1997

[9] G. Pulin and X. Jianxue, "On the multiple-attractor coexisting system with parameter uncertainties using generalized cell mapping method," *Journal Applied Mathematics and Mechanics*, Vol. 19, No. 12 December, 1998, pp. 1179-1187

[10] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, 43:59-69, 1982

[11] T. Kohonen, *Self-Organizing Maps*, Springer, Berlin (3rd extended ed. 2001), 1997

[12] P. Somervuo, "Speech Dimensionality Analysis on Hypercubical Self-Organizing Maps," *Neural Processing Letters*, Vol. 17-2, April 2003, pp. 125-136

[13] A. Selin, J. Turunen, and J. T. Tanttu, "Wavelets in Recognition of Bird Sounds," *EURASIP Journal on Advances in Signal Processing*, Vol. 2007, Article ID 51806, 9 pages, doi:10.1155/2007/51806

[14] J. C. Sprott, *Strange Attractors: Creating Patterns in Chaos*, M & T Books, 1993-09

[15] J. C. Sprott, *Chaos and Time-Series Analysis*, Oxford University Press, 2003

[16] S. King and P. Taylor, "Detection of phonological features in continuous speech using neural networks," *Computer Speech & Language*, Volume 14, Number 4, October 2000 , pp. 333-353(21)

[17] G. Windisch and L. Csink, "Language Identification Using Global Statistics of Natural Languages," *Proc. of 2nd Romanian-Hungarian Joint Symposium on Applied Computational Intelligence*, Timisoara, Romania, May 12-14, 2005, pp. 243-255

[18] S. Mohammad and T. Pedersen, "Combining Lexical and Syntactic Features for Supervised Word Sense Disambiguation", *Proceedings of the Conference on Computational Natural Language Learning* (CoNLL-2004), May, 2004, Boston, MA

[19] R. O. Duda, P. E. Hart and David G. Stork, *Pattern Classification*, John Wiley Interscience, 2001

Cascade Control Solution for Traction Motor for Hybrid Electric Vehicles

Zsuzsa Preitl, Péter Bauer

Department of Control and Transport Automation
Budapest University of Technology and Economics
H-1111 Budapest, Hungary
E-mail: preitl@sch.bme.hu, bauer.peter@mail.bme.hu

József Bokor

Computer and Automation Research Institute
Hungarian Academy of Sciences
H-1518 Budapest, Hungary
E-mail: bokor@sztaki.hu

Abstract: In this paper a hybrid electric vehicle is considered, which contains both an internal combustion engine and an electric motor (EM). Without focusing on the other components of the vehicle, the EM is treated in detail, both regarding modelling aspects and control solutions.
After a brief modelling of the plant, two cascade speed control solutions are presented: first a classical PI+PI cascade control solution is presented. The control systems related to traction electric motors (used in vehicle traction) must be able to cope with different requests, such as variation of the reference signal, load disturbances which depend on the transport conditions and parametric disturbances regarding changes in the total mass of the vehicle. For this purpose, in the design of the speed controller (external loop) a specific methodology based on extension of the symmetrical optimum method is presented. The controllers are developed using the Modulus–Optimum method for the inner loop, and the Extended Symmetrical Optimum Method, corrected based on the 2p-SO-method, for the outer loop (for a more efficient disturbance rejection).
In order to force the behaviour of the system regarding the reference input, a correction term is introduced as a non-homogenous structured PI controller solution.
Simulations were performed using numerical values taken from a real application consisting in a hybrid vehicle prototype, showing satisfactory behaviour.

Keywords: Electric Hybrid Vehicle, Driving system, Speed control, Extended Symmetrical Optimum method, 2p-SO-m

1 Introduction

Electric motors (EM) are used in a large variety of applications in industry, one of them is traction motors. Low power traction motors in electrical drive vehicles are frequently oriented on DC-machines (DC-m) or brushless DC motors (BLDC-m) [3], [4], [5] (but other solutions are also used). From the point of view of mathematical modelling, the two solutions differ only insignificantly, mainly on parameter calculus relations.

The control systems related to traction electric motors (used in vehicle traction) must be able to cope with different requests, determined by the multitude of conditions to which the process must fulfill:

- the vehicle's speed must be adapted to the actual traffic conditions, as a consequence, the *reference of the system* is permanently variable;
- depending of the vehicle velocity, route, weather etc., *load type disturbances* are permanently present and changing;
- with the modification of the total mass of the vehicle the *equivalent moment of inertia* is also modifying, and so the large time constant of the vehicle, resulting in a varying *parametric disturbance*;

These conditionings impose a tuning of the control parameters which fulfil simultaneously all requests. The paper is focused on control solution for an electrical driving application (electrical traction) as part of a hybrid electric vehicle. Details regarding the vehicle itself can be found in [1], [2].

The paper is structured as follows. Section II presents a detailed mathematical model of a separately excited DC-machine. Section III describes two control strategies applied for speed control, both consisting in cascade control. Section IV introduces the numerical values used, and based on these, simulations are performed and analysed. Finally, section V concludes the paper.

2 Plant Model

2.1 General Aspects

The functional block diagram of a series hybrid electric vehicle (HEV) is presented in Figure 1. The main components of the system are: the EM, which drives the wheels and whose control is dealt with in the paper (it can also work as a generator during regenerative braking), the electric generator which delivers electrical energy for the EM, a battery, the controllers and the power electronics. The electric generator is in rigid connection with the internal combustion engine.

2.2 Basic Equations

The operating range of a DC-m is divided into four quadrants: forward motoring, forward breaking, reverse motoring and reverse braking [5], [6], [7]. For the driving system (EM and the vehicle) a qualitative and quantitative modelling is used. The mathematical model of the driving system includes both the model of the motor and the dynamical model of the system.

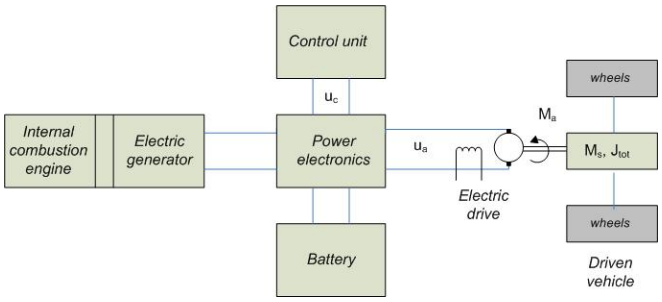

Figure 1
Functional block diagram of a series hybrid electric vehicle

• *Modelling of the motor*. The hypotheses accepted at modelling imply that in normal regimes the DC-m works in the linear domain where the flux (current) is constant in value (valid also for BLDC-m). A change in the excitation regime modifies the basic model, but a linearization in the new working point results in the basic situation.

The basic equations that characterize the functionality of the system are given in (1), where the following notations were used: T_A – time constant of the actuator (power electronics) [sec], u_a – armature voltage [V], k_a – actuator gain, u_C – command voltage from controller [V], L_a – inductance [H], T_a – electrical time constant, i_a – field current [A], e – counter electromotive voltage [V], k_e – coefficient [V/rad/sec], ω – rotor speed [rad/sec], J_{tot} – total moment of inertia of the plant [kgm^2], M_a – active torque [Nm], M_s – load torque [Nm], M_f – friction torque [Nm], J_m – moment of inertia of the DC-m [kgm^2], J_{veh} – moment of inertia of the vehicle reduced to the motor axis [Nm^2], J_w – moment of inertia of the two driven wheels reduced to motor axe (converted) [kgm^2].

$$
\begin{aligned}
&T_A \cdot \dot{u}_a + u_a + k_a = u_C \\
&L_a \cdot di_a + R_a \cdot i_a = u_a - e \\
&T_a = L_a / R_a \quad , \quad e = k_e \omega \\
&M_a = k_m \cdot i_a \\
&J_T \dot{\omega} = M_a - M_s - M_f \\
&J_{tot} = J_m + J_{veh} + J_w
\end{aligned}
\tag{1}
$$

As a result, the block diagram of the DC-m is depicted in Figure 2. Based on the bloc diagram from Figure 2, the four transfer functions (t.f.s) according to the DC-m can be defined; $\{H_{\omega,uc}(s), H_{\omega,ms}(s), H_{ia,uc}(s), H_{ia,ms}(s)\}$. The main t.f. regarding to which the controller will be designed is $H_{\omega,uc}(s)$; their expressions can be detailed for two more remarkable cases: $k_f \neq 0$ and the approximation case $k_f = 0$ frequently used in practice, rel. (2)-(5)

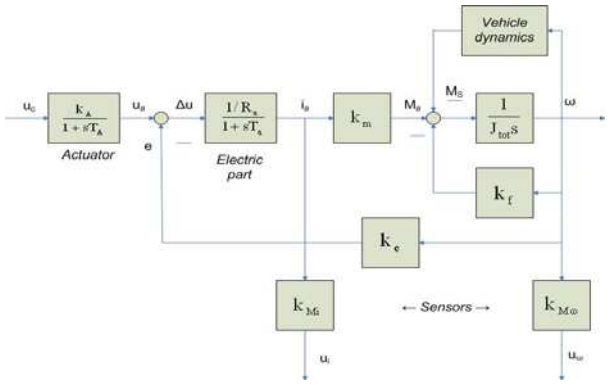

Figure 2
Block diagram of a DC-m

$$H_{\omega,uc}(s) = \frac{k_A}{1+sT_A}\frac{1/k_e}{1+sT_m+s^2T_aT_m} \approx \frac{k_A}{1+sT_A}\frac{1/k_e}{(1+sT_a)(1+sT_m)} \tag{2}$$

The factorised for is valid for $T_m >> T_a$, specific for electric traction.

$$H_{ia,uc}(s) = \frac{k_A}{1+sT_A}\frac{sT_m/R_a}{1+sT_m+s^2T_mT_a} \approx \frac{k_A}{1+sT_A}\frac{sT_m/R_a}{(1+sT_a)(1+sT_m)} \tag{3}$$

$$H_{\omega,ms}(s) = -\frac{R_a}{k_mk_e}\frac{1+sT_a}{1+sT_m+s^2T_mT_a} \approx -\frac{R_a}{k_mk_e}\frac{1+sT_a}{(1+sT_m)(1+sT_a)} \tag{4}$$

$$H_{ia,ms}(s) = -\frac{R_a}{k_m}\frac{1}{1+sT_m+s^2T_mT_a} \approx -\frac{R_a}{k_m(1+sT_m)(1+sT_a)} \tag{5}$$

Where the mechanical time constant is calculated based on relation: $T_m = \frac{J_{tot}R_a}{k_mk_e}$.

The total inertia is calculated as follows. It is supposed, as in eq. (1), that the total inertia contains the inertias of the vehicle, of the DC-m and of the two driven wheels with the drive shaft. This way, from the energy conservation principle, the following are derived:

$$\frac{1}{2} m_{tot} v^2 = \frac{1}{2} J_{veh} \omega^2 \quad \Rightarrow \quad J_{veh} = \frac{m_{tot} v^2}{\omega^2} \tag{6}$$

But $\omega = f_r \cdot \omega_v$, $v = r \cdot \omega_v = r \cdot \frac{\omega}{f_r}$

It results: $J_{veh} = \frac{m_{tot} \cdot r^2 \cdot \omega^2}{f_r{}^2 \cdot \omega^2} = m_{tot} \frac{r^2}{f_r{}^2}$ (7)

where: - ω_v - speed of the drive shaft and wheel; - r - radius of the wheel; - m_{tot} – total mass of the vehicle (including the driver); - v – linear velocity of vehicle; - f_r – drive ratio.

- *Equations of the vehicle dynamics.* The basic dynamical equations for the vehicle motion are presented in eq. (6) [1]:

$$\omega(t) = \frac{f_r}{w_r} v(t)$$

$$M_d(t) = \frac{w_r}{f_r} F_d(t) \tag{8}$$

$$F_d(t) = m \cdot \dot{v}(t) + \frac{1}{2} \rho v^2(t) \cdot A_d \cdot C_d + m_{veh} \cdot g \cdot C_r$$

2.3 About Drive Cycles

The testing of the behaviour of a vehicle through simulation requires a given reference that must be followed. Such reference signals, consisting in a pre-defined time-vehicle velocity scheme, are called drive-cycles [1]. In this paper a section of the New European Driving Cycle will be used for testing, consisting in an acceleration, then constant speed, and breaking until zero velocity is reached.

It must also be mentioned that when modelling the electric vehicle a driver behaviour model can also be taken into account, which has effect on the reference delivered to the electric drive. The modelling of the other functional blocks of the electric vehicle is not subject of this paper, but they are described in [1], [2].

3 Control Structures and Controller Design

3.1 Control Aim and Performances

The aims of the control structures applied to the DC-m are grouped as follows:

- To ensure good reference signal tracking (speed) with small settling time and small overshoot (good transients and zero-steady-state error at *v*=const. velocity).
- To ensure load disturbance rejection due to modifications in the driving conditions.
- To show reduced sensitivity [8] to changes in the total inertia of the system:

$$J_{tot} = J_{t0} + \Delta J_t \qquad \text{with} \qquad \Delta J_t \le 0.25 J_{t0} \tag{9}$$

The adopted two solutions are classical ones, both having two control loops in cascade structure:

- One interior control loop of the current, consisting in a PI controller and Anti-Windup-Reset (AWR) measure.
- One external control loop of rotor speed ω [rad/sec] with a PI controller.

The second control structure differs from the first through the outer loop, in which a forcing block was added to correct the current reference for the inner loop. It can decrease the response time of the system.

3.2 Presentation of the Control Loops. Controller Design

The block diagrams of the two control structures are depicted in Figures 3 and 4, in the form of Simulink diagrams. The two controllers, the current controller and the speed controller, are designed separately.

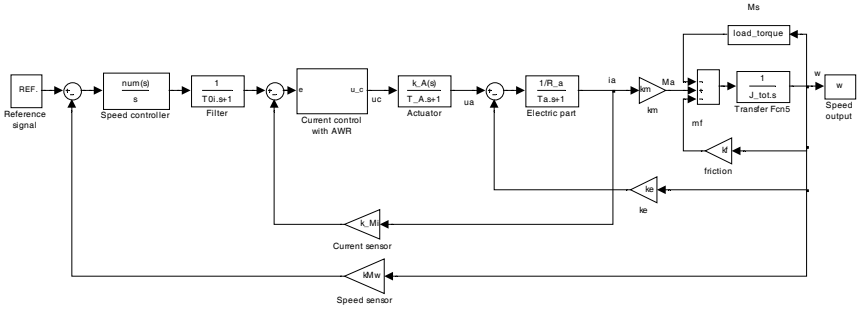

Figure 3
First cascade control structure for the DC-m

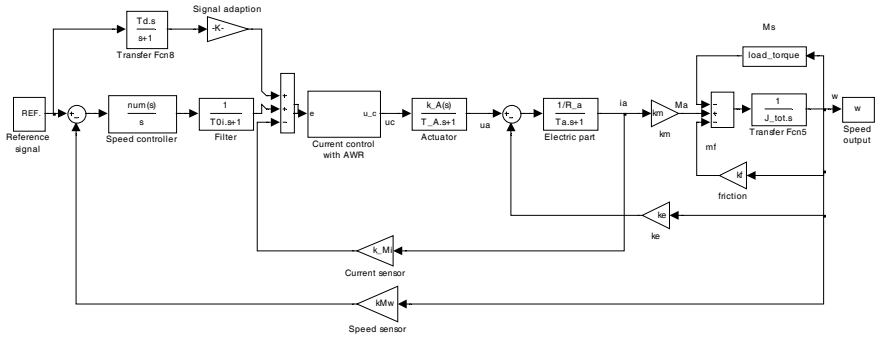

Figure 4
Second cascade control structure for the DC-m

- *The inner current loop* is identical in both cases, and it consists of a PI controller with AWR measure [9], [10]. The parameters of the current controller were determined in accordance with the properties of the inner loop, based on the Modulus Optimum criterion (MO) by Kessler [9], having the relations:

$$C(s) = \frac{k_{ri}}{s}(1 + sT_i),\ k_{ri} = \frac{1}{2k_{pi} \cdot T_{\Sigma i}},\quad T_{ii} = T_a \tag{10}$$

Where, k_{pi} – gain of the inner part of the plant, containing the actuator, electric circuit and current sensor), T_a – electric time constant, $T_{\Sigma i}$ – equivalent of small time constants, $T_a > T_{\Sigma i}$.

The AWR measure was introduced to attenuate the effects of going into limitation of the controller. Other methods for handling constraints of the control signal are also used, for example a solution where the controller itself is by a dynamic feedback of a static saturation element [11].

- *The speed control loop*, as the outer loop, consists of a PI controller in two variants for implementation: one homogenous variant and one case when a forcing filter for the reference value was introduced. This second variant ensures the possibility of accelerating the vehicle, depending on the power of the driving motor.

$$P(s) = \frac{k_p}{s(1 + sT_\Sigma)} \tag{11}$$

where T_Σ stands for the current loop and parasitic time constants, k_p characterizes the dynamics of the mechanical part of the driving system (J_{tot}) and the speed sensor ($k_{M\omega}$.). The transfer function of the PI controller has the transfer function:

$$C(s) = k_C\left(1 + \frac{1}{sT_c}\right) = \frac{k_c}{s}(1 + sT_c) \tag{12}$$

The open loop transfer function (t.f.) results:

$$L(s) = C(s)P(s) = \frac{k_c k_p (1 + sT_c)}{s^2 (1 + sT_\Sigma)} \tag{13}$$

And so the closed loop t.f. is:

$$H_r(s) = \frac{k_c k_p T_c s + k_c k_p}{s^3 T_\Sigma + s^2 + k_c k_p T_c s + k_c k_p} = \frac{b_1 s + b_0}{a_3 s^3 + a_2 s^2 + a_1 s + a_0} \tag{14}$$

with $b_1 = a_1$, $b_0 = a_0$ (due to the double integrator component in the open loop t.f.)

The design of the speed loop is based on an extension of the SO method from Kessler [7], the Extended Symmetrical Optimum method (ESO-m) [14]; the method is based on the following parameterization:

$$\beta^{1/2} a_0 a_2 = a_1^2 \quad , \qquad \beta^{1/2} a_1 a_3 = a_2^2 \tag{15}$$

Here β is a parameter that is chosen by the developer. A larger value of β ensures less oscillating transients and a bigger phase margin. Consequently, the controller parameters are calculated with the relations:

$$k_{c0} = \frac{1}{\beta^{3/2} k_p T_\Sigma^2} \, , \, T_{c0} = \beta T_\Sigma \tag{16}$$

In second stage, taken into account that $k_f>0$, the type of the load and that the system performance regarding reference tracking are less satisfactory, the results obtained in the first phase are corrected according to the double parameterization of the SO-m, introduced in [15] as 2p-SO-m and the particularity of the plant (inner loop). Designing the controller based on this approach, there can be ensured:

- Use of pre-calculated (crisp) tuning relations;
- The possibility of improving the system's phase margin, reducing its sensitivity and increasing its robustness;
- The possibility of using controllers with homogenous structure or with non-homogenous structure regarded to the inputs.
- The possibility of improving reference signal tracking by using reference filters with parameters that can be easily fixed.
- The possibility of improving reference tracking using adequate reference filters [15] and load disturbance rejection for some specific cases.

For a second order with lag benchmark type model of the plant, the controller tuning relations specific for 2p-SO-m are close to the ESO method, they allow an efficient correction of the controller parameters depending on the plant's time constants T_1, T_Σ (). For this, the correction relations can be used [15]:

$$k_c = \frac{(1+m)^3}{m} T_\Sigma \, k_{c0}, \text{ and } \quad \alpha_{kc}{}^* = \frac{k_c}{k_{c0}} = (1+m)^3 T_1 \tag{17}$$

$$T_c = T_{c0} \frac{\Delta_m(m)}{(1+m)^3} \text{ and } \quad \alpha_{Tc}{}^* = \frac{T_c}{T_{c0}} = \frac{\Delta_m(m)}{(1+m)^3} \tag{18}$$

For the chosen value of the β parameter, the controller parameters calculated in the first stage with the ESO-m are corrected regarding the relations (17) and (18), where the values of α_{kc} and α_{Tc} take into account the values of β and $m = T_{\Sigma a} / T_m$, with $T_{\Sigma a}$ - the time constant that characterizes the inner loop (current) and T_m - the mechanical time constant of the plant (see rel. (2)).

4 Case Study. Simulation Results

4.1 Numerical Values of the Plant

Details and numerical data for the considered application (a hybrid solar vehicle) are presented in [2]. Numerical values of the DC-m in the nominal functioning are synthesized in table 1. Further numerical values used (see also [1], [2]):

- Total mass of vehicle, including an 80kg heavy driver: $m_{tot}=1860\ kg$;
- Frontal area of vehicle: $A_d=2.4\ m^2$;
- Air drag coefficient: $C_d=0.4$;
- Air density: $\rho=1.225\ kg/m^3$;
- Rolling resistance coefficient: $C_r=0.015$;
- Wheel radius: $w_r=0.3\ m$;
- Final drive ratio: $f_r=4.875$.

The resulting time constants and other plant parameters are enumerated below:

- Mechanical time constant: $T_m=5.4$ sec;
- Electrical time constant: $T_a=0.1\ sec$;
- Total inertia: $J_{tot}=8.6^2$;
- Gain and time constant of actuator: $k_A=30$, $T_A=0.02\ sec;$
- Gains for current and speed sensors: $k_{Mi}=0.0238$, $k_{M\omega}=0.0178$.

Table 1
Numerical values for nominal functioning

Torque	Rotation	Useful power	Voltage	Current	Absorbed Power	Efficiency
[Nm]	[rot/min]	[kw]	[V]	[A]	[kw]	[-]
50,16	1605	8,43	77,6	126	9,78	86,18

The controller parameters are:

- Current controller: $k_{ri}=7$, $T_{ri}=0.1$ (according to equation (8), plus an AWR time constant according to [9] having the value of $T_t=0.005$;
- Speed controller:
  - For the first case, see eq.(9): $k_c=35.28$, $T_c=1.75$;
  - For the second case the controller is the same, the feed forward correction term is of form: $C_{ff}(s)=\frac{560s}{s+1}$.

4.2 Simulation Results

The simulation scenarios are the following: the first control structure is simulated, followed by the second cascade structure simulations (comparison of the currents' and dynamics), ended by simulations for the first case regarding sensitivity aspects for a change in the mass of the plant. The reference signal is the same for all three cases, consisting in an acceleration part, a part with constant velocity and a part of deceleration until a stop is reached. The load of the system is taken into account as in [16].

(a) Simple cascade structure: Figures 6, 7, 8 and 9.

(b) Cascade structure with correction of the current reference: in this case the differences in the current behaviour are depicted, together with the active power (dashed line – simple cascade structure, solid line – structure with current correction). The differences in the speed dynamics are not significant, the active power differences are proportional with the current, Figures 10 and 11.

(c) Simple cascade structure with modified load (Figures 12, 13 and 14) (for the first cascade structure): the mass of the vehicle is changed with +25% of it (solid line – original load, dashed line – increased load):

$m_{veh}=m_{veh0}+\Delta m=1860+0.25*1860=2332kg.$

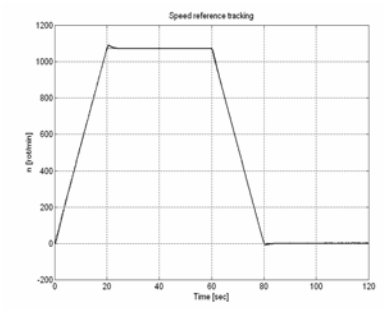

Figure 6
Speed reference tracking

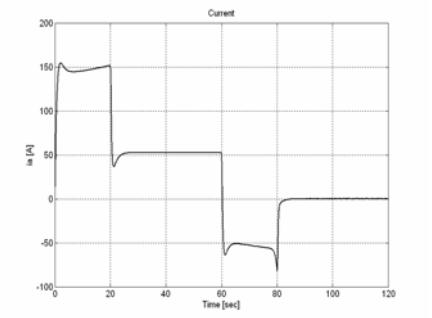

Figure 7
Behaviour of the current

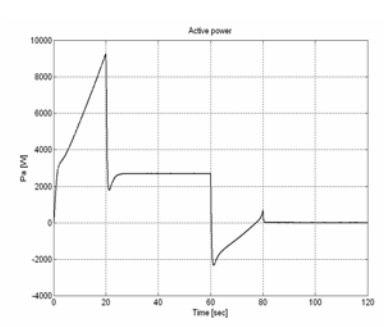

Figure 8
Active power consumption (negative values mean generation)

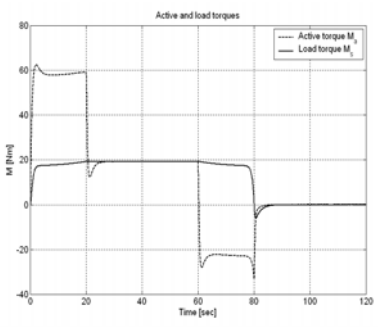

Figure 9
Active torque M_a vs. disturbance torque M_s

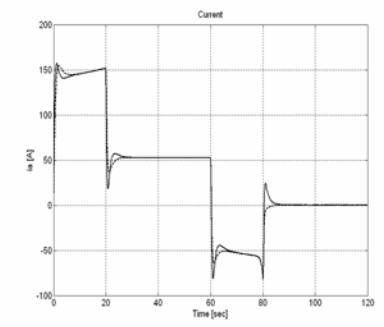

Figure 10
Comparison of the currents

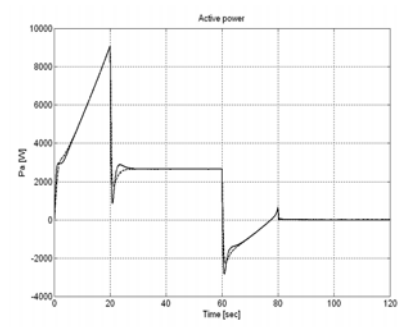

Figure 11
Comparison of the active powers

The speed was not presented since almost the same behaviour resulted. But in order to achieve this performance, the current is higher (since it needs more power to carry the increased weight). Still the current does not reach its maximal

admissible value (4 times the nominal value of *126 A*). The active power is higher (*12 kW* compared to *9kW* at starting), but without exceeding the maximal power of *15 kW* of the machine. Both the active torque and the load torque are higher, as expected. Regenerative braking appears when the current (and implicitly the active power) is negative.

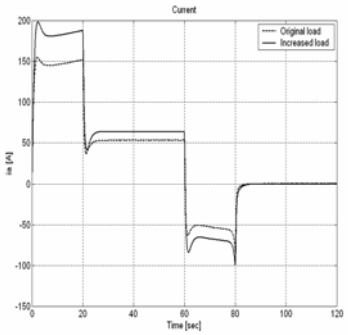

Figure 12
Behaviour of the current

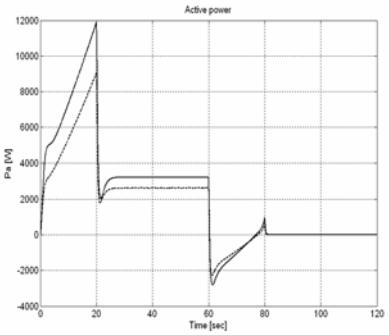

Figure 13
Active power consumption (negative values mean generation)

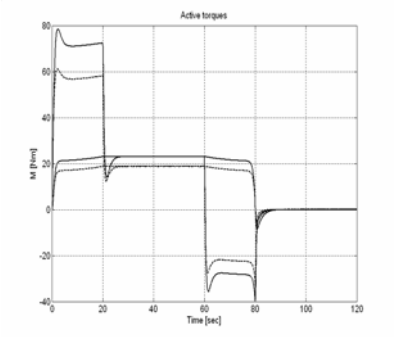

Figure 14
Active torque M_a vs. disturbance torque M_s

Conclusions

The paper presents a cascade control solution for electrical drives used for traction in two variants – without and with a forcing feed forward term for the current reference –, both consisting of cascade control structures. In order to ensure high performances, for controller design different variants of the Modulus Optimum tuning method (MO-m) were used, namely the Extended Symmetrical Optimum method (ESO-m) and a correction of it based on the tuning method named a Double Parameterization of the ESO method (2p-ESO-m), introduced in [15].

Simulations were performed using the Matlab/Simulink environment, for a reference drive cycle. The simulated cases reflect a very good behaviour of the system both regarding reference tracking and also sensitivity to parameter changes. The application considered in the paper is based on a real application of a hybrid solar vehicle.

Acknowledgements

The authors address their thanks to the support from the Hungarian National Office for Research and Technology through the project 'Advanced Vehicles and Vehicle Control Knowledge Center' (ref. number NKTH RET04/2004), which is gratefully acknowledged. The authors also gratefully acknowledge the contribution of Hungarian National Science Foundation (OTKA #K60767).

References

[1] P. Bauer, Zs. Preitl, T. Péter, P. Gáspár, Z. Szabó, J. Bokor (2006). Control-oriented Modelling of a Series Hybrid Solar Vehicle, *Workshop on Hybrid Solar Vehicles*, November 6, 2006, University of Salerno, Italy

[1] I. Arsie, R. Di Martino, G.Rizzo, M. Sorrentino (2006). A Model for a Hybrid Solar Vehicle Prototype, *Workshop on Hybrid Solar Vehicles*, November 6, 2006, University of Salerno, Italy

[2] R. M. Crowder (1998). *Electric Drives and their Controls*, Oxford University Press Inc., New York

[3] M. Ehsani, K. M. Rahman, M. D. Bellar, A. J. Severinsky (2001). Evaluation of Soft Switching for EV and HEV Motor Drives, *IEEE Transactions on Industrial Electronics*, Vol. 48, No. 1, February 2001, pp. 82-90

[4] R. Schonfeld (1988). *Digitale Regelung Electrischer Antriebe,* Dr. Alfred Huthig Verlag, Heidelberg, 1988

[5] G. Rizzoni (1993). *Principles and Applications of Electrical Engineering*, Richard D. Irwin Inc.

[6] C.-M. Ong (1998). *Dynamic Simulation of Electric Machinery Using Matlab/Simulink,* Prentice Hall PTR, Upper Saddle River, New Jearsey 07458

[7] K. J. Åstrom. *Model Uncertainity and Robust Control. Chapter on Control Theory*, (Internet presentation), pp. 63-100

[8] K. J. Åstrom, T. Hägglund (1995). *PID Controllers. Theory, Design and Tuning* Research Triangle Park, North Carolina

[9] P. Hippe, C. Wurmthaler (1999). Systematic Closed Loop Design in the Presence of Input Saturation, *Automatica*, Vol. 40, pp. 1221-1228

[10] A. Barta, R. Bars, I. Vajk, Zs. Preitl (2005), Practical Controller Design Considering Input Saturation, *6th International Carpathian Control Conference ICCC 2005,* Lillafüred, Hungary, May 24-27, 2005, Proceedings, Vol. I, pp. 51-56

[11] J. Quevedo, T. Escobet (Editors) (2000). *IFAC workshop on Digital Control. Past present and Future of PID Control, PID'00*, Preprints, Terrassa, Spain, April 5-7, 2000

[12] K. J. Åstrom, T. Hägglund (2000). Benchmark Systems for PID Control*, IFAC workshop on Digital Control*, Terrassa, Spain, April 5-7, 2000, pp. 181-182

[13] S. Preitl, R.-E. Precup (1999). An Extension of Tuning Relations after Symmetrical Optimum Method for PI and PID Controllers*, Automatica,* Vol. 35, No. 10, pp. 1731-1736

[14] Zs. Preitl (2005). Improving Disturbance Rejection by Means of a Double Parameterization of the Symmetrical Optimum Method, *Scientific Bulletin of the "Politehnica" University of Timişoara, Series Automation and Computers*, Politehnica Publishing House, Timişoara, ISSN 1224-600X, Vol. 50(64), pp. 25-34

[15] M. Imecs (2000). How to Correlate the Mechanical Load Characteristics, PWM and Field-Orientation Methods in Vector Control Systems of AC Drives, *Buletinul Institutului Politehnic Iasi,* Tomul XLVI (L), Fasc. 5*,* Electrotehnica, Energetica, Electronica, A X-a Conferinta Nationala de Actionari Electrice

Medical Predictions System

Doina Drăgulescu

Politehnica University of Timisoara
RO-300222 Mihai Viteazu Blvd.
ddrag@cmpicsu.upt.ro

Adriana Albu

Politehnica University of Timisoara
RO-300223 Vasile Parvan Blvd.
adriana.albu@aut.upt.ro

Abstract: Health has a strong impact upon all activities and human experts must have the ability to decide, in any circumstances, what is the illness level of a patient, which is the adequate treatment and which will be the evolution of the patient during the treatment. But medical decision making may be a very difficult activity. There are a lot of applications in artificial intelligence domain that try to help human experts offering solutions for a problem. This paper describes an expert system developed in order to make some predictions regarding the hepatitis infection.

Keywords: hepatitis infection, expert system, logical inference, statistical inference, artificial neural networks

1 Introduction

Medical domain is characterized, like many other domains, by an exponential evolution of the knowledge. There are a lot of tools which try to reduce the risk of error apparition in medical life. Diagnosis has a very important role here. It is the first step from a set of therapeutic actions; an error at this level can have dramatic consequences.

The presence of technology in diagnosis phase is welcome because of its advantages: pragmatism, repeatability, efficiency, immunity toward perturbation factors that are specific to human beings (fatigue, stress, diminished attention). The technology doesn't replace human experts in this point of medical assistance; it only tries to help them, implementing systems that are able to select or to generate data which are relevant for the physicians.

The system presented here belongs to this context. It is made using the main two branches of artificial intelligence:

— the traditional one, represented by expert systems (based on logical and statistical inference);

— the connexionist one, where the most common forms used are artificial neural networks.

The goal of the system is to offer predictions about patients infected with hepatitis virus. Hepatitis is one of the principal causes for liver cancer. A correct diagnosis and an adequate treatment could reduce the risks of liver cancer apparition.

The first step is to decide, using logical inference, what type of hepatitis virus is present. There are three possibilities:

— hepatitis B

— hepatitis B+D

— hepatitis C.

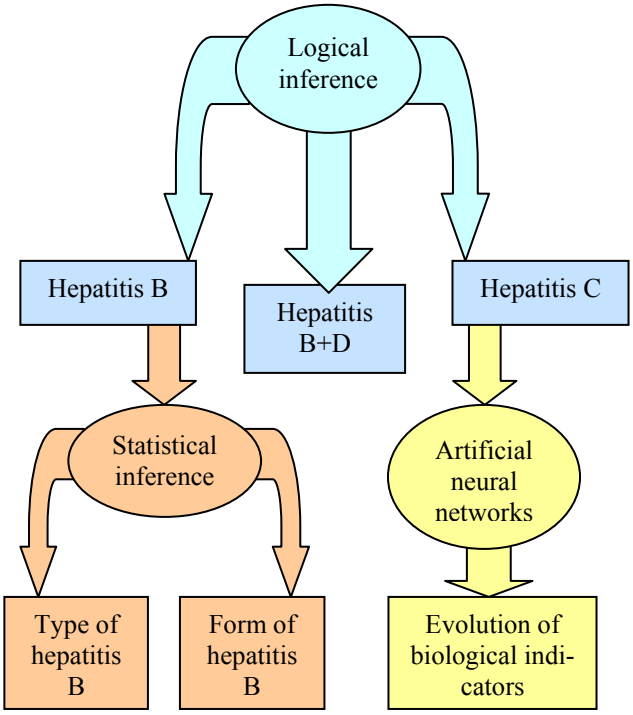

Figure 1
The structure of the system

After the type of hepatitis was set, it is necessary to find some more information about this. There are many forms of hepatitis B. The system described in this paper will decide, based on statistical inference, which one is possible to appear for a patient. If the disease is hepatitis C, it will be important to predict the treatment response and the evolution of laboratory analysis during the treatment, because hepatitis C has a very expensive treatment and severe side effects can often appear. Artificial neural networks will be used in order to do the predictions regarding hepatitis C. Fig. 1 is a schematic description of the system presented in this paper. The application is implemented in MATLAB 7.0, which is a high-performance language and integrates computation, visualization, and programming in a very attractive environment.

Of course such a system will never replace human experts. A tool made to suggest a decision is able to extract information from other solved cases so it can obtain experience and can also take into consideration the results of the last researches, but won't be able to replace the most important factor in decision making: human judgment [1]. Therefore, the final decision has to be made by a human expert. These systems are created only to suggest a solution.

2 Methods

A Expert Systems

There are two main possibilities of implementing expert systems: by logical inference and by statistical inference. Both of them were used in this system, in order to make some predictions regarding the hepatitis diagnosis and the evolution of an infected patient.

Logical Inference

The logical inference could be used in medicine to build expert systems that will produce a diagnosis starting from a set of premises. An expert system implements human reasoning and it needs some rules to make it possible. This type of system is also called rules based expert system and it is the most used system for implementing medical diagnosis [2]. It has a graph structure and a chain logical evaluation is applied on this structure. Such an expert system could be easy to implement and also very easy to use for a non-engineer because its rules are similarly with the natural medical language.

For hepatitis diagnosis it is necessary to specify which are the factors that define different types of hepatitis. After that, the rules for the expert system can be drawn.

There is a set of markers that have to be analyzed in order to decide what type of hepatitis is present in a patient organism. These markers are described in Table I.

Table I
The markers for the hepatitis diagnosis

Marker	Value	Name
AgHBs	Positive	M1
AgHBs	Negative	M2
anti – VHD	Negative	M3
anti – VHD	Positive	M4
anti – VHC	Positive	M5

There are considered three possibilities: hepatitis B virus, hepatitis B+D virus and hepatitis C virus. The logical model consists of the following rules, which are created using the markers that appear in Table I:

R1: If M1 and M3 then B

R2: If M1 and M4 then B+D

R3: If M2 and M5 then C

Fig. 2 presents the expert system built using these rules.

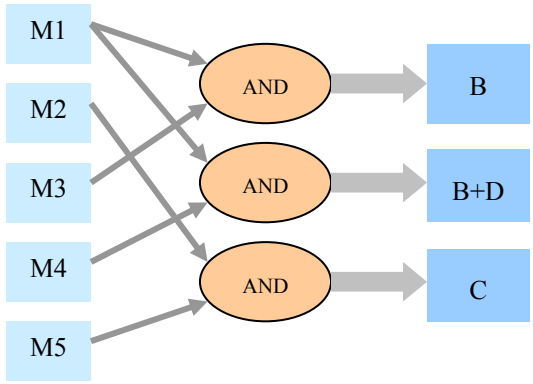

Figure 2
Rules based expert system for hepatitis diagnosis

This type of expert system is easy to be implemented for simple rules like 'logical premises → conclusion', but it is not suitable to use logical inference for huge amounts of connected knowledge because the graph becomes too complex. Frequently, it is hard to express the rules for the system and also the translation of implicit knowledge into explicit rules would lead to loss and distortion of information content [3]. On the other hand, the tree structure of rule-based relationships becomes too complex if new levels of knowledge are added. For example, there are many types of hepatitis B and if the system described before has to decide between these types, it will be difficult to implement it.

Statistical Inference

The statistical inference is an alternative to logical inference and offers a lot of methods that use information of a sample, to learn about population characteristics and to provide some conclusions or decisions. A problem that must be taken into consideration is linked to the fact that inferences are done based on the information contained in a sample, which is only a part of the whole population. From this point of view it is necessary to indicate the precision of the results. The probability plays an important role, being used to define the quality of an affirmation, to measure the uncertainty or to describe the chance for an event to happen.

In this area, the most frequently used method is the *Bayes's theorem*, which sets a probabilistic value for each considered output (disease, if the system is applied in medical diagnosis). Bayesian networks have an important area of applicability in the entire field of artificial intelligence, setting a posterior probability when prior probability is known [4].

Bayes's theorem suggests that probabilities can be improved with new information (Fig. 3). The analysis starts with the prior probabilities (preceding the experience) for the interesting events. Then it is used a supplementary information from a sample, a test, a report or from other sources, information that affects the probability of the events. The prior probability will be revised using this new information and the result will be the posterior probability (after the experience and based on the experience). Bayes's theorem is an easy way to find the posterior probability.

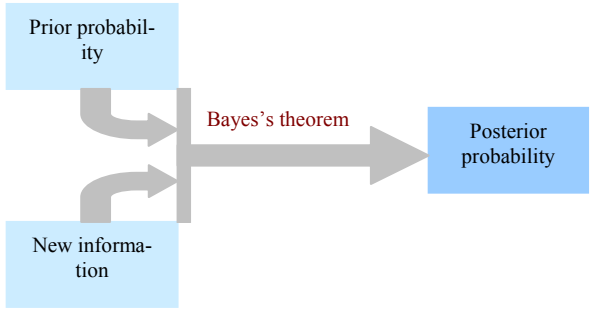

Figure 3
Bayes's theorem

There are three evolutional types of hepatitis B (usual, with relapses and with decompensations) and six grades of disease (easy, medium, grave, prolonged, cholestatic and comatose). It is very useful to have an expert system that can predict, using symptoms and laboratory test results, what type and what form of hepatitis B is present for a new patient. Bayes's theorem [5] will be used to build such a system. It needs a database with symptoms for a number of patients (Ω - statistical population) that have associated a final diagnosis set. In this application was used a database with over 150 patients with hepatitis B virus infection.

Bayes's theorem is a formula with conditioned probabilities. If it is applied in medical diagnosis, its form is:

$$p(D_k \mid S) = \frac{p(S \mid D_k) \cdot p(D_k)}{p(S)} \tag{1}$$

where D_k is a disease and S a set of symptoms. Using the theorem it can be calculated, for a patient, the probability of appearance for each disease D_k when the set of symptoms S is present.

$p(D_k)$ is easy to find because the frequency of apparition of disease $\delta_k=1$ in statistical population Ω is known:

$$p(D_k) = \frac{card\{x \in \Omega \mid \delta_k(x) = 1\}}{card\Omega} = \frac{cardD_k}{card\Omega} \tag{2}$$

$p(S| D_k)$ can be calculated if the considered symptoms are conditioned independents for a disease δ_k:

$$p(S \mid D_k) = \prod_{i=1}^{n} p(\sigma_i \mid D_k) \tag{3}$$

where σ_i is a symptom,

$$p(\sigma_i \mid D_k) = \frac{p(\sigma_i, D_k)}{p(D_k)} \tag{4}$$

and:

$$\begin{aligned} p(\sigma_i, D_k) &= \frac{card\{x \in \Omega \delta_k(x) = 1, \sigma_i(x) = 1\}}{card\Omega} \\ &= \frac{card\{D_k \cap S_i\}}{card\Omega} \end{aligned} \tag{5}$$

$p(S)$ is hard to be determined. If it is supposed that a patient suffers of only one disease at a moment, then the following formula could be used:

$$p(S) = \sum_{j=1}^{m} p(S \mid D_j) \cdot p(D_j) \tag{6}$$

where j is an index of all investigated diseases $\delta_1, \delta_2, \ldots, \delta_m$.

The Bayes's theorem becomes:

$$p(D_k \mid S) = \frac{p(D_k) \cdot \prod_{i=1}^{n} p(\sigma_i \mid D_k)}{\sum_{j=1}^{m} \left[p(D_j) \cdot \prod_{i=1}^{n} p(\sigma_i \mid D_j) \right]} \tag{7}$$

with $k = 1,..., m$.

This formula will be applied for each evolutional type and each form of hepatitis B disease, offering for each one a plausibility score.

Such an expert system could be successfully used if it is developed for mutual exclusive diseases and independent symptoms. But sometimes these restrictions cannot be accomplished because there are situations when some symptoms have the same cause (being connected) and a patient can suffer of more than one disease. It was also observed that Bayes's theorem needs an excessive calculation time if statistical population Ω is very large. In order to avoid these problems, two other statistical algorithms were implemented: Aitken's formula and Logistic model.

Aitken's formula [5] is an alternative for equation (3) (which is the most time consumer in Bayes's theorem). The probability $p(S \mid D_k)$ can be quickly found if this formula is used:

$$p(S \mid D_k) = \frac{1}{T} \sum_{t=1}^{T} \lambda_\delta^{\,n-st} \cdot (1 - \lambda_\delta)^{st}, \quad k = 1,..., m \tag{8}$$

where: m – the number of considered diseases;

T – total number of patients;

λ_δ – smoothing factor for the disease δ ($0.5 \le \lambda_\delta \le 1$);

st – Hamming distance between the vector of new patient's symptoms $S = (S_1, S_2, \ldots, S_n)$ and the vector of symptoms of the patient t from the database $S^t = (S_1^t, S_2^t, \ldots, S_n^t)$.

The Hamming distance derives from the Minkovski formula. If all the elements S_i and S_i^t (i=1, …, n) are binary codified, than the Hamming distance is the number of elements that are different in S and S^t:

$$d_{Hamming} = \sum_{i=1}^{n} XOR(S_i, S_i^t) \tag{9}$$

Logistic model [5] is a solution to the problem of mutual exclusive diseases which appears in Bayes's theorem. It starts with the notion of anti-probability:

$$o(E) = \frac{p(E)}{p(\overline{E})} = \frac{p(E)}{1 - p(E)} \tag{10}$$

and conditioned anti-probability:

$$o(E \mid F) = \frac{p(E \mid F)}{p(\overline{E} \mid F)} \tag{11}$$

From (10) and (11), where E and F are two events, can be written equations (12) and (13):

$$p(E) = \frac{o(E)}{1 + o(E)} \tag{12}$$

$$p(E \mid F) = \frac{o(E \mid F)}{1 + o(E \mid F)} \tag{13}$$

It is easier to calculate o(E|F) than p(E|F). Logistic discrimination will be used in order to find the logarithm of the anti-probability of disease D_k conditioned by the vector S:

$$ln\, o(D_k \mid S = s) = w_{0k} + \sum_{i=1}^{n} w_{ik} \cdot sign(\sigma_i) \tag{14}$$

where: n – the number of symptoms;

m – the number of diseases;

$k = 1, \ldots, m$;

w_i – are called 'weights' and they are calculated with the equations (15) and (16):

$$w_{0k} = ln\, o(D_k) \tag{15}$$

$$w_{ik} = ln \frac{p(\sigma_i \mid D_k)}{p(\sigma_i \mid \overline{D_k})} \tag{16}$$

For the patient that is diagnosed it is analyzed the list of symptoms and it is calculated for each symptom σ_i the value of the function *signum*, using the expression (17):

$$sign(\sigma_i) = \begin{cases} -1, if\ \sigma_i = 0 \\ 1, if\ \sigma_i = 1 \end{cases}, \quad i = 1, \ldots, n \tag{17}$$

At the end, the probability of apparition for each disease D_k when a set of symptoms S is present can be found:

$$p(D_k \mid S) = \frac{e^{ln\,o(D_k|S)}}{1 + e^{ln\,o(D_k|S)}} \tag{18}$$

This is exactly the desired result.

B Artificial Neural Networks

There are a lot of cases when is not possible to implement human intelligence with expert systems. This is the reason why artificial neural networks have been developed. The initial idea was that in order to reproduce human intelligence, it would be necessary to build systems with a similar architecture [6].

Artificial neural networks are developed based on brain structure, representing a simplified mathematical model of central nervous system. Like the brain, artificial neural networks can recognize patterns, manage data, and, most important, learn [7]. They are made by artificial neurons, which implement the essence of biological neuron.

In this system, artificial neural networks are used in order to make some predictions regarding the treatment response for a patient infected with hepatitis C virus. Hepatitis C is a serious and frequent disease and its evolution has to be carefully overseen during the treatment. Even the efficiency of the hepatitis C treatment improves continuously, the burden of this infection will remain a major issue for the next several decades.

The patients from this study (almost 200) have been kept under observation for 12 months to establish the treatment's influence on the evolution of four biological indicators (TGP, TGO, GGT, and ARN VHC). Three different treatment schemes have been instituted:

— Simple Interferon (IFN);

— Peg interferon α-2a;

— Peg interferon α-2b.

The system offers for each evaluated biological indicator predictions regarding the next 12 months evolution, indicating its growing tendency, its stabilizing or decreasing tendency. It was developed using feed-forward neural networks with back-propagation learning algorithm. Its architecture is in fact a network of neural networks. Each neural network has a layer of 10 hidden neurons, a single output unit and a variable number of inputs.

For each of the four biological indicators that have been studied, there are four layers of neural networks. The networks on the first layer receive as inputs: patient's age, sex, location (rural/urban), treatment scheme, Knodell score, hepatic fibrosis score and value of the parameter for which the prediction is made, at the initial moment (before the treatment starts). These networks have as output the value of the biological parameter at 3 months. On the following layers the net-

works have the same structure as the first layer ones, but they have in addition, as inputs, the outputs of the networks on the former layers; therefore, the networks on the last layer will have not 7 inputs (as the networks on the first layer) but 10 (the initial inputs and the values of biological indicators at 3, 6, and 9 months).

The advantage of this architecture is that the input data are processed separate for each biological indicator. The disadvantage is that the errors are propagated through the system because the results of the networks from the first level (together with their errors) are used in the following levels. But this disadvantage can be minimized by learning process.

3 The Expert System

The application has a complex structure, analyzing information connected to the apparition of the hepatitis infection, its evolution, the antecedents, the symptoms, the results of the laboratory tests, and the evolution of some specific biological indicators during the treatment. It develops a multifunctional database and implements an expert system used in order to diagnose different types of hepatitis and to realize some predictions regarding the evolution of the patient and the response to the treatment. The system uses two major components (an inference machine and an architecture of neural networks) that operate on the multifunctional database (Fig. 4). It has an interdisciplinary character and fulfils the requirements of a system used in medical diagnosis and prediction.

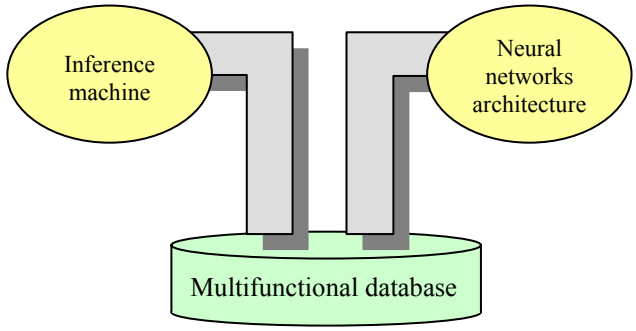

Figure 4
The configuration of the expert system

First of all, the system offers the possibility to diagnose the most frequent hepatitis types: B, B+D and C. Logical inference is used in order to do this. The result can be seen in one of the applications interfaces (Fig. 5). The user has to set the values of the markers that determine which is the hepatitis type. After that, on the bases of the rules described in section II, the result is displayed.

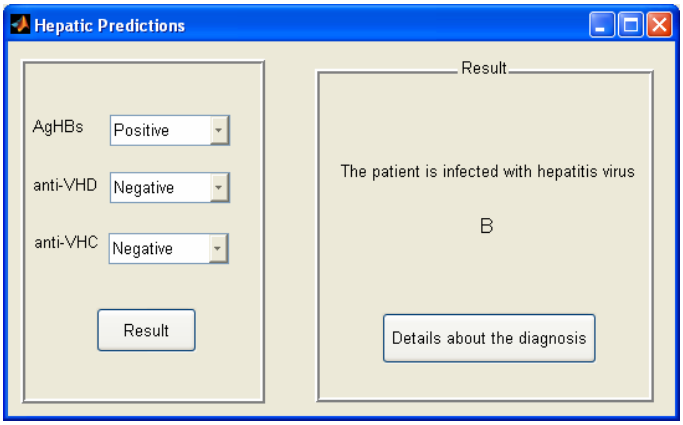

Figure 5
Hepatitis diagnosis

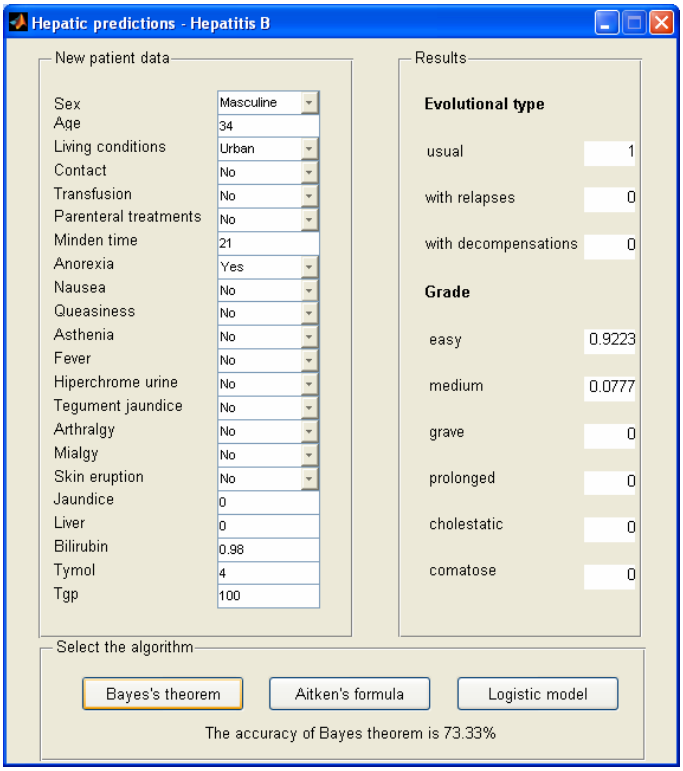

Figure 6
Hepatitis B – Predictions regarding evolutional type and form

For example, if AgHBs is positive, anti-VHD is negative and anti-VHC is also negative, then the patient is infected with hepatitis B virus. If the human expert needs more predictions regarding the diagnosis, than he can use the other two branches of the application. The button 'Details about the diagnosis' from Fig. 5 will lead to apparition of Fig. 6 if the diagnosis is hepatitis B or Fig. 7 if patient is infected with hepatitis C virus.

For hepatitis B is developed an expert system based on statistical inference. The user must set the characteristics of the patient: sex, age, living conditions, symptoms, and the results of laboratory tests. After that, he will choose one of the three implemented algorithms (Bayes's theorem, Aitken's formula, or Logistic model) and the plausibility scores for each evolutional type and grade of hepatitis B are calculated (as can be seen in the right part of Fig. 6).

These statistical algorithms are using a part of the multifunctional database: 165 patients infected with hepatitis B virus. The data which describe medical status of these patients were collected from Clinical Hospital of Infectious Diseases No. 4 'Victor BABES', Timisoara.

Fig. 7 is the user interface for hepatitis C predictions. The user has to choose a range regarding the age of the patient, the sex, the location where the patient lives (rural/urban), the treatment (IFN, Peg interferon α-2a or Peg interferon α-2b) and has to introduce the values of the Knodell score and of the fibrosis score. It is also necessary to introduce the values of the biological indicators before the treatment. The system will predict the evolution of the biological indicators depending on the treatment. Looking at the predicted tendency of the biological indicators during the treatment, a physician can estimate if the patient will respond to a treatment or not.

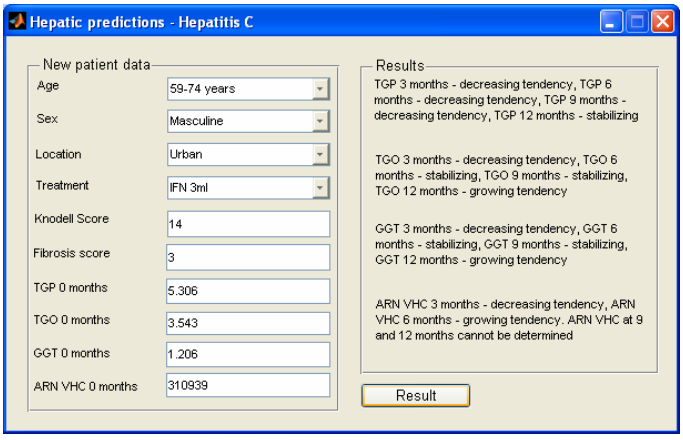

Figure 7
Hepatitis C - The prediction of biological indicators evolution

The artificial neural networks, which were used in order to do the predictions regarding the patient's treatment response, have the ability of learning and they need some data about a lot of patients (symptoms, laboratory tests, characteristics, etc.). All these are stored in another part of multifunctional database, which contain almost 200 patients infected with hepatitis C virus. These real data were collected from Country Clinical Emergency Hospital, Timisoara.

Conclusions

This paper tried to evidence some important aspects connected to medical decision making. Therefore, the system presented here is made from three important parts. First of all, logical inference is used to decide what type of hepatitis virus is present for a new patient. The possibilities are B, B+D and C. After that, the second part of the system will be used to see what will be the type and the grade of hepatitis B (if the patient is infected with hepatitis B virus). This branch of the system is developed using methods from statistical inference. The third part of the system is made for the patients infected with hepatitis C virus and it predicts the biological parameters evolution during the treatment using artificial neural networks.

The hepatitis is a serious disease, its treatment is expensive and severe side effects can appear very often. Therefore, it is important to set a correct diagnosis and to identify those patients who most probably can react to the treatment, so that the others can be protected from a treatment with no benefits. That's for what the use of such a system can support the physicians' decisions.

References

[1] V. Podgorelec, P. Kokol: Self-Adapting Evolutionary Decision Support Model, in *Proceedings of the IEEE International Symposium on Industrial Electronics*, 1999, Vol. 3, pp. 1484-1489

[2] D. Petrică: Structure of Models for Medical Knowledge Processing, in *Scientific Bulletin of "Politehnica" University Timişoara, Romania, Transactions on Automatic Control and Computer Science*, 2004, Vol. 49 (63), No. 2

[3] J. M. Zurada: Introduction to Artificial Neural Systems, *West Publishing Company*, United States of America, 1992

[4] D. Niedermayer: An Introduction to Bayesian Networks and their Contemporary Applications, www.niedermayer.ca/papers/bayesian/bayes.html, 1998

[5] N. Szirbik, D. Pescaru: Expert Systems, *Politehnica University Timisoara*, 2000

[6] The Statistics Homepage: Neural Networks, www.statoftinc.com/textbook/stathome.html

[7] P. A. Maiellaro, R. Cozzolongo, P. Marino: Artificial Neural Networks for the Prediction of Response to Interferon plus Ribavirin Treatment in Patients with Chronic Hepatitis C, in *Current Pharmaceutical Design*, Vol. 10, pp. 2001-2009, 2004

Map Building and Localization of a Robot Using Omnidirectional Image Sequences

Zoltán Vámossy

Institute of Software Technology
John von Neumann Faculty of Informatics
Budapest Tech
Bécsi út 96/B, H-1034 Budapest, Hungary
vamossy.zoltan@nik.bmf.hu

Abstract: The paper describes a map building module, where the image sequences of the omnidirectional camera are transformed into virtual top-view ones and melted into the global dynamic map. After learning the environment from training images, a current image is compared to the training set by appearance-based matching. Appropriate classification strategies yield an estimate of the robot's current position.

Keywords: PAL (Panoramic Annular Lens), omnidirectional vision, appearance-based navigation

1 Introduction

The goal of this project is to create a wheeled robot equipped with a panoramic annual lens (PAL)-optics, which is capable of autonomous navigation and collision avoidance within a weakly textured environment. The long-term goal of this project is the ability of autonomous mapping of the environment; finding, and navigating through user-specified path; and searching for a predefined object within an unknown environment. The digitalized picture of a camera serves as base information, which is filtered with different image processing algorithms. One of the elements of the investigation is the analysis of these basic algorithms and their collective effect. Another utilization of the camera image is to scan the position of objects in the robot's environment, and the mechanism of the map-building. In the following, the article summarizes the theoretical background, the main components, the applied techniques and the results of the system.

2 Omnidirectional Imaging [1]

Imaging in general means the effort to portray the three-dimensionality of space conveyed by signal bearing waves on an Euclidean flat surface. Omnidirectional imaging shows 'panoramic vision', which means that there is pictorial representation encircling the spectator having real objects as a foreground.

The problem, however, is how to achieve that appearance of things at a given place and time conceived will be recorded and displayed as something actually existing at the place and time, even when no perceiver is present and a constituent of the object whose appearance it is, i.e. a scenic picture giving an effect of extension of the vista i.e. three dimensionality results. With other words, the problem of imaging and displaying is connected to the means how to obtain the optical imprint of the three-dimensional world in such a way that the cortex space in our mind matches the physical space, which is mostly experienced by touch, muscle tension and movement.

In general, our imaging strategy assumes that the person (or imaging system) stands facing the 3D environment on a level plane and looks through the picture plane – represented as an upright transparent surface – at a space chunk, which contains the natural horizon (the distant line where the earth or sea apparently meets the sky) where parallel lines are going to merge in a point, the vanishing point.

The emphasis is on 'space chunk', since the metric relation of a sphere cannot be mapped onto the metric relations of a plane, and, as a result, the sphere of vision can be perceived only in discrete chunks, called visual field.

It follows from this so-called See-Through-Window (STW) imaging strategy that our visual apparatus, and any imaging system based on its analogy, is not capable of perceiving and/or recording 360° panoramic view of space at once, i.e., it cannot render a 'pictorial representation of space and time data encircling the spectator and often having real objects as a foreground'.

2.1 Omnidirectional Imaging is the Result of Centric Minded Thinking

If one assumes that the geometric structure of space encircling us is cylindrical, rather than spherical, centric minded imaging (CMI) can be established. This way of thinking may be backed up by the observation that the visual signal processing of Mother Nature seems to operate according to a similar philosophy. The presence of vertical parallax is less important for us than the horizontal one, further, stereopsis exists only horizontally, and, therefore, it is more appropriate to speak of cylinder of vision, instead of sphere of vision.

If now it is assumed that the radius of the circumscribed cylinder is equal to the vision distance, a panoramic view of the image volume shows up on the wall of this imaginary cylinder. However, the result of this course of thoughts is only a 360° panoramic view, but not an omnidirectional panoramic image in the sense of image definition, since it is not an intensity pattern displayed on an Euclidean flat surface yet.

It can be shown that by using special stretching maneuvers one can transform this panoramic view image projection onto a plane surface perpendicular to the axis of the imaginary cylinder. As a result, a panoramic annular image of the three-dimensional environment is formed, where points in the cylindrical space seen at constant field angles perpendicular to the axis of the cylinder of vision are located on concentric rings in the image plane. The geometric relation of the three-dimensional environment remains represented in polar coordinates and provides an image in which the points retain the same 1:1 relation to each others as in reality. This allows a distortion-free omnidirectional display of the imaged scene.

Analyzing further the annular image one can find that this optical imprint displays the 2D skeleton of the encircling 3D environment in such a way that one may get data on the place and time position of object points, since the width of the ring shaped image corresponds to the viewing angle in the direction of the axis of the cylinder of vision.

From the described stretching and transformation maneuvers it follows that the technical realization of such an optical system must be of catadioptric type.

Several patents have been filed all around the world, claiming better performing optical systems for CMI. All these endeavors can be classified into two main groups: either they are based on multiplexed element design using several optical elements such as lenses and/or cones and/or prisms and/or mirrors with coinciding optical axes, or the others use a single glass block with sophisticatedly shaped refracting and reflecting surfaces.

2.2 PAL, the Omnidirectional Imaging Block

The omnidirectional imaging block PAL (Panoramic Annular Lens) consists of a single glass block with reflective and/or refractive plane, concave and convex surfaces. This means that already in the simplest case the number of possible shapes of the imaging block amounts to the number of the iterative variations of the fourth class which can be formed of three elements, i.e., to 81. However, it has to be emphasized that only a few of them produce good quality images. A well designed PAL-optic

1) is almost afocal, and both a virtual and a real image are formed inside the optic;

2) it renders, via a relay lens, a sharp image from right up against the lens surface out to infinity;

3) its center region around the optical axis does not take part in the forming of the panoramic annular image; it serves only to ensure undisturbed passing through of the image forming rays;

4) objects to the front of the optic are imaged to the interior of the annular image, and objects to the rear of the optic appear on the outer rim of the annular image;

5) a collimated light beam entering the PAL-optic through its plane surface, after passing through the lens, leaves it in form of a light cylinder that evenly illuminates the surrounding. The height of the light cylinder at a given distance from the optical axis depends upon the refractive index of the material the CMI block has been made of.

The interpretation of the resulting panoramic images may cause some confusion, since we are not accustomed to see our three-dimensional world in front and behind us simultaneously, only in discrete chunks successively. Since relations can be established between the polar coordinates of this centric minded imaging (CMI) and the rectangular coordinates of the STW image plane, the annular images can be displayed in Cartesian coordinates. Using appropriate software, the ring shaped image can be 'straightened out', i.e., the 360° panoramic image displayed in polar coordinates can be converted into Cartesian coordinates, and the mentioned discomfort immediately disappears.

3 Experimental Mobile Robot System

A remote controlled Model RC is used as the base of the mobile robot [2]. The Model RC is capable of precision controlling: both the direction and the speed can be set. The range of the remote control is about 30-40 meters and the maximal speed is about 20 km/hours.

The PC is connected to the remote controller with an extra electronics, and then controls both controller transistors with three impulses. The impulses are generated by an 18F1320 PIC micro controller. An interrupt is generated in the PIC program with a timer every 15ms. On every interrupt, the PC is sending three signals: the first signal sets the beginning of the periodical time, the second impulse controls the direction of the car; the third control-data sets the speed of the car.

The wireless camera with PAL optics is mounted on the top of the mobile car (Figure 1), looking at the floor; therefore, it is capable of observing the immediate environment only. The result of the camera is relayed using a TV tuner to the main PC program, which controls the robot automatically using the images gained from the camera as its only input.

Figure 1
The mounted PAL on the Model RC

4 Map Building from the Image Sequences

The image flow arrives from the input module, which is responsible for either capturing images from a camera, or play back a test video file. It forwards the images to both the decision maker, and the map builder module. The decision maker analyses the image, and sends a direction/speed signal to the navigation module, which, in turn, forwards it to the controller PIC.

4.1 Image Preprocessing

In order to make a valid control decision, the image is preprocessed by three filters:

1) Using a HSL filter, the program segments the image to Hue, Saturation, and Luminance components. The Hue component is between 0, and 360, the Saturation, and the Luminance will fall between 0, and 100. The HSL filter is used in two experiments: in line following mode, when the predefined track is homogeneous, or in object-following mode when the object to be followed is significantly differing in color from the rest of the environment.

2) The RGB filter is almost as efficient as the HSL, but the algorithm is significantly faster. Using this filter, the three color channels is analyzed using a minimum and a maximum values; if the color of the pixel is within these values, then it remains intact on the resulting image; otherwise the filter will make it black.

3) Using the threshold filter, image binarization can achieved; the result image will contain only the pixels needed for navigation.

4.2 Feature Tracking and Selection

Let *I* and *J* be two 2D gray scaled images [2]. The image *I* will be referenced as the first image, and the image *J* as the second image. Consider an image point *u* on the first image. The goal of feature tracking is to find the location *v* on the second image *J* such as the intensities are 'similar'. The vector *d* describes the relative motion is the image velocity at the point, also known as the optical flow. It is essential to define the notion of similarity in a 2D neighborhood sense. During the optical flow calculation the problem is to define the velocity *d* as being the vector that minimizes the summed squared differences of the intensities in *I* and *J* for a given integration window. To provide solution to that problem, the pyramidal implementation of the Lucas–Kanade algorithm is used. An iterative implementation of the Lucas-Kanade optical flow computation provides sufficient local tracking accuracy.

So far the tracking procedure is described that takes care of following a point *u* on an image *I* to another location *v* on another image *J*. However it is not described what means to select the point *u* on *I* in the first place. This step is called feature selection. It is very intuitive to approach the problem of feature selection once the mathematical ground for tracking is led out. At that step, the spatial gradient matrix is required to be invertible, or in other words, the minimum eigenvalue must be large enough (larger than a threshold). This characterizes pixels that are easy to track. Therefore, the process of selection goes as follows:

1) Compute the matrix and its minimum eigenvalue λ at every pixel in the image *I*.

2) Call λmax the maximum value of λ over the whole image.

3) Retain the image pixels that have a λ value larger than a percentage of λmax. This percentage can be 10% or 5%.

4) From those pixels, retain the local maximum pixels (a pixel is kept if its λ value is larger than that of any other pixel in its neighborhood).

5) Keep the subset of those pixels so that the minimum distance between any pair of pixels is larger than a given threshold distance (e.g. 10 or 5 pixels).

After that process, the remaining pixels are typically 'good to track'. They are the selected feature points that are fed to the tracker.

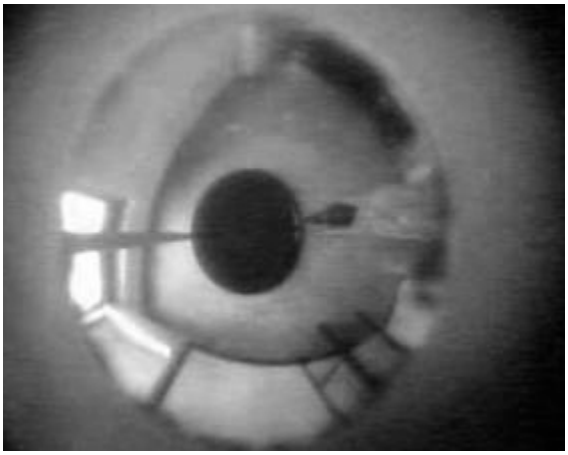

Figure 2
The PAL image

4.3 Map Building

The map builder module prepares the user-defined navigation: after the robot builds up the map of the environment, the user sets some checkpoints, and the robot tries to find the shortest path, and navigate through them, while avoiding any obstacles. The specific purposes of this module are localization of the robot, and maintaining a virtual top-view map of the environment.

To achieve this goal, the image (Figure 2) sequences received from the PAL-optics is mapped to a 'virtual top-view' (Figure 3). The result of this transformation is the following: the straight edges lies on the floor near to the robot become straight on the transformed image also, and this step later allows the insertion of the transformed part in the global map. To apply this transformation the algorithm assumes that the PAL-image is a regular annulus. Thus, transforming the distance of the pixels from the PAL-center will result in a top-view, map-like image. To determine the parameter of the transformation function, during a calibration process the relation is measured between the real distance and the pixel based PAL-distance on several points from the center of the image; and a cubic spline interpolation is applied on the measured data.

To increase the performance of the algorithm a transformation matrix is generated, which determines the source for every pixel on the resulting image. Once this matrix is generated, it can be used for every image, with real-time speed. After the top-view transformation, the module uses a static mask to cut out segments from the image, which has no information-content (for example, the central blind-spot), or the segment shows object lies far away from the robot.

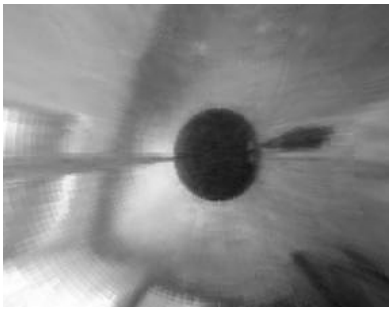

Figure 3
Virtual top-view image

The localization process uses the combination of two techniques: the program searches and tracks characteristic features on the resulting image, and parallel it calculates a summed histogram value in radial directions in every 3 degrees. The feature points and the summed histogram values are used to determine the relative location, and angle of the robot. After localizing the robot, the module will rotate the image to reflect the initial direction of the robot; the resulting image is melted into the global map also.

In order to dynamically extend the map as the robot gathers more and more information, a static bitmap would not be sufficient; instead, the map is divided it into several, small images, and the module stores the two dimensional ordering between these segments. The 'melting' of one image into the global map is used in the navigation process.

5 Results

Although the tests started recently, some early results are already available. The developed system was tested indoor and outdoor environment also. Both cases the free work space were weakly-textured and significantly lighter then the obstacles. In the first experiments some simple navigation tasks were tested: collision avoidance, line-, and object following. One complex algorithm is used for all three of the navigation types.

First, the properties of the PAL-optics needs to determined. For line following, the algorithm determines a direction line, which converges to the center of the image, and begins inner radius distance away from the center point. The optimal length of the line of sight can be set using the distance value.

The algorithm determines the pixel intensity under the line of sight starting from 90 degrees (forward), scanning at each iteration first left, then right, until it

reaches ±90° scanning degree value. For every line of sight, it determines the sum of the pixel intensities, whether it exceeds the value set by threshold minimum; the maximum of these values will be used as a navigation direction. This value is later verified, whether the robot fits on the given path, or not. If there is no appropriate path on the top part of the image, the bottom part will be analyzed by the same method, and the robot will either reverse, or stop.

For free fall navigation, the Canny edge-detection algorithm is used: the contour on the image will be interpreted as obstacle to avoid. The algorithm will potentially avoid these obstacles by selecting the longest clear direction.

For object following, the algorithm uses the RGB, and HSL filter described above, to distinct the object from the rest of the environment. After applying binarization, the controlling line will head towards the object.

The developed map building system was tested in outdoor environment located near the building of dormitory. After building the map from the virtual top-view image sequences (Figure 4), binarization was used to determine the free working space of the robot. After this step, path was planned with a wave propagation based method to achieve the goal.

Conclusions

As conclusion the system has met its current specified criteria: it is capable of collision avoided line-, and object following, within a weakly textured environment and the robot capable to build up appearance based map. To improve the skills of the robot the method can be found in [4] will be implemented.

Acknowledgement

The research has been supported by the OM TDK grant under the terms of grant OM FPO 245540/2005.

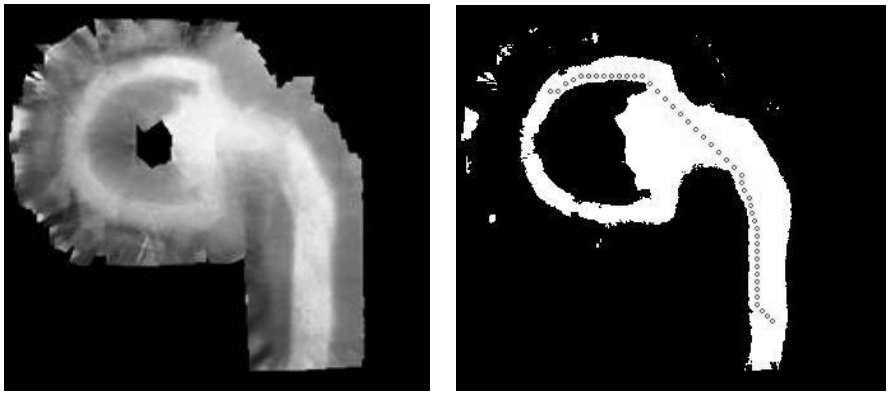

Figure 4
Constructed map from the melted image sequences (left) and the path (gray dots) generated with the wave propagation technique (right)

References

[1] P. Greguss, F. Alpek, M. Patko, "Changing from Guided Robocars to Autonomous Robocars by Using Humanoid Machine Vision System", in. *Proc. of 7th International Workshop on RAAD'98*, Bratislava, Slovakia, 1998, pp. 127-131

[2] J. Y. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker", Intel Corporation, Microprocessor Research Labs, 2000, http://www.intel.com/research/mrl/research/opencv/

[3] L. Mornailla, T. Pekár, Cs. Solymosi, Z. Vámossy, "Mobile Robot Navigation Using Omnidirectional Vision", in *Proc. 15th Int. Workshop on Robotics in Alpe-Adria-Danube Region*, June 15-17, 2006, Balatonfüred, Hungary, CD, ISBN 963 7154 48 5

[4] J. Tick: "Workflow Model Representation Concepts" in *Proc. of 7th International Symposium of Hungarian Researchers on Computational Intelligence*, HUCI 2006, Budapest, Hungary, November 24-25, 2006, pp. 329-337

Using Petri Nets to Enhance Web Usage Mining[1]

Shih-Yang Yang

Department of Information Management
Kang-Ning Junior College of Medical Care and Management
Nei-Hu, 114, Taiwan
Shihyang@knjc.edu.tw

Po-Zung Chen, Chu-Hao Sun

Department of Computer Science and Information Engineering
Tamkang University
Tamsui, 251, Taiwan
pozung@mail.tku.edu.tw, steven.sun@fubon.com

Abstract: Precise analysis of the web structure can facilitate data processing and enhance the accuracy of the mining results in the procedure of web usage mining. Many researchers have identified that pageview identification and path completion are of great importance in the result of web usage mining. Currently, there is still a lack of an effective and systematic method to analyze and deal with the two steps.
In the present study, we propose the application of Petri Nets (PN), a model used to analyze the framework of webpages in a website. We adopt Place in the PN model to represent webpage on the websites and use Transition to represent hyperlink. The study explores how to undergo the pageview identification after we use the PN model to conduct the analysis of the framework and then obtain incident matrix. Likewise, we use reachability property in the model to undergo path completion.

Keywords: Petri nets, Web usage mining, Data Preprocessing

1 Introduction

This study introduces a method to enhance a web usage mining using Petri Nets (PN) in modeling a web structure. We also observe that PN can help resolve pageview identification and path completion, particularly in a complex webpage comprising many frames in one single pageview.

[1] This work is supported by the National Science Council, Taiwan, ROC, under Grant # NSC 93-2213-E-032-016.

As the internet becomes globally popular, more and more business transactions have been done through websites nowadays. To achieve a better website management and design capability, many website management personnel started reviewing their site users' webpage browsing frequency, sequence and even duration on each webpage browsed, adopting the user's web usage profiles. Hence, web usage mining has become a hot research topic.

A website is comprised of a series of web pages and hyperlinks. Although most web-usage-mining related studies only focus on, and analyze, the users' web usage profiles, some related studies [1] [2] point out that a good-quality analysis on web structure can provide gains and benefits for web usage mining, too. Most previous web usage mining utilizes a webpage as an analysis component, but after the HTML standard language, such as 'frameset', started being applied to webpage design, a website user's browsed display might exhibit more than one web page concurrently; hence, an analysis based on a web page component can no longer truly reveal the user's usage states and behaviors. A pageview is defined as the visual picture of a web page in a specific client environment at a specific point in time [3]. The use of pageview as the analysis component for web usage mining could more accurately reveal a website user's browsing behaviors, but such method will increase the complexity of data preprocessing during web usage mining.

In general, the process of web usage mining can be divided into three parts, namely preprocessing, pattern discovery, and pattern analysis [4]. Preprocessing will process untreated site files and user profile data into page classification, site topology and server session files. Pattern discovery will process a server session file into rules, patterns, and statistics information. Pattern analysis looks into the rules, patterns, and statistics information obtained from pattern discovery for results that will be of interest to the management personnel.

The preprocessing step can be generally divided into content preprocessing, structure preprocessing, and usage preprocessing. Content preprocessing classifies site files into page classification to help pattern analysis. Structure preprocessing converts site files into a site topology to help pageview identification and path complete. Usage preprocessing coverts raw usage data into click stream of episodic user behaviors via some steps such as data cleaning, user identification, session identification, pageview identification, path complete (if necessary).

Concerning data cleaning, user identification, and session identification faced in a usage preprocessing process, many previous studies have investigated a great deal and rendered some processing steps and methods [5] [6]; although some researchers point out that an understanding on website structure can be useful in carrying out pageview identification and path complete [1][7][8], it is still hard to establish fine solutions for pageview identification and path complete.

Petri Nets (PN) is a high-level graphical model widely used in modeling system activities with concurrency. PN can store the analyzed results in a matrix for

future follow-up analyses, and some already-verified properties held by PN, such as reachability, can also be used to resolve some unsettled problems in the model. According to the definition in [9], it is formally defined as a 5-tuple PN of (P, T, I, O, M_0), where

(1) $P = \{p_1, p_2 \ldots p_m\}$, a finite set of places;

(2) $T = \{t_1, t_2 \ldots t_n\}$, a finite set of transitions; $P \cup T \neq \emptyset$, and $P \cap T = \emptyset$;

(3) I: P x T→N, an input function that defines directed arcs from places to transitions, where N is a set of nonnegative integers;

(4) O: T x P →N, an output function that defines directed arcs from transitions to places;

(5) M_0: P →N, the initial marking. A marking is an assignment of tokens to a place;

PN is carried out by firing transitions. A transition, t, is said to be enabled if each input place, p, of t contains at least an amount of token equal to the weight of the directed arc connected to t from p. In a PN model, we can utilize the different token amounts in the places to represent the different system states. Since a fire of transition in the system often can be associated with a change of the token amount in a place, PN hence can represent, or model, the system dynamic behaviors via the fire of transitions. An incidence matrix records all token-amount changes in all places after all fired transitions. For PN with n transitions and m places, the incidence matrix A, where $A=[a_{ij}]$, is an n×m matrix of integers; its typical entry is given by

$$a_{ij}=a_{ij}^{+} - a_{ij}^{-}, \quad (1)$$

where

$a_{ij}^{+}=O(t_i, p_j)$, the weight of the arc from Transition i to its Output Place j, and

$a_{ij}^{-}=I(ti, p_j)$, the weight of the arc to Transition i from its Input Place j;

a_{ij}^{+}, a_{ij}^{-} and a_{ij} represent the number of tokens removed, added, and changed in Place p_j, respectively, when Transition t_i fires once.

Concerning the PN properties, reachability is one of them that is often discussed and utilized; this property is originally expected to explore if the modeled system can be transited from one state to another. During the processing or operations, we can also simultaneously trace out what are the possible intermediate states during the transitions from the initial state to the destination one. In a PN model, a marking M_i is said to be reachable from a marking, M_0, if there exist a sequence of transition firings which can transform a marking, M_0, to M_i. According to the definition in [10], we can obtain the state equation as shown in (2) as follows,

where A is an incidence matrix, $\{ U_0,U_1,\ldots,U_d \}$, representing the firing sequence from M_0 to M_d if M_d is reachable from M_0:

$$M_d=M_0+A^T \sum_{k=1}^{d} Uk \tag{2}$$

The purpose of this study is to enhance web usage mining by PN. We use a parsing algorithm to retrieve the website's webpage contnets, analyze the webpage's contents and find the incidence matrix which represents web structure. We can apply the web structure information in the incidence matrix and the reachability properties obtained from the PN model to help proceed with pageview identification and path complete. Also we apply Markov analysis to provide usage statistics in pattern discovery. Figure 1 represents our proposed PN based web usage mining structure.

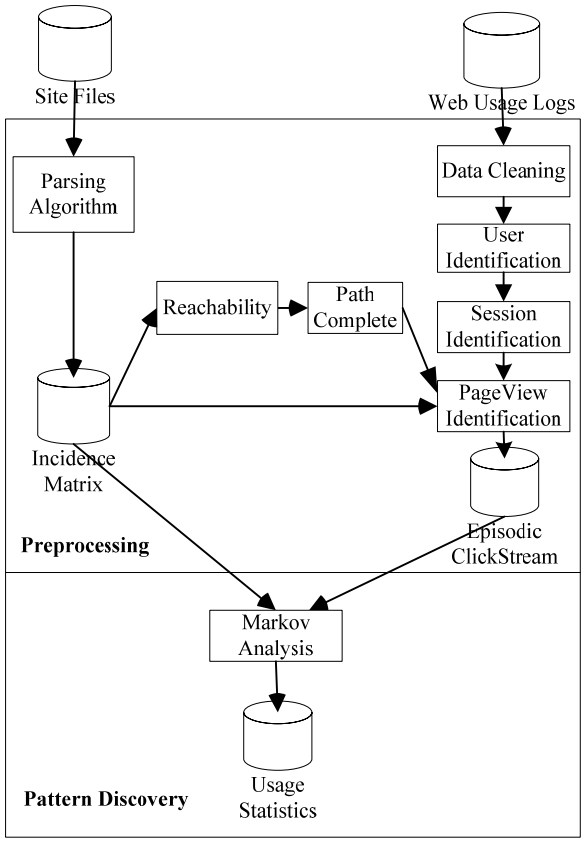

Figure 1
PN based web usage mining structure

In this paper, we focus on how to utilize a parsing algorithm to retrieve the website's webpage contents, analyze the webpage's contents and find the incidence matrix which represents web structure. Also we show how to apply the reachability properties to help pageview identification and path complete processes. For the part of how to apply Markov analysis to provide usage statistics in pattern discovery, please refer to [11].

2 Modeling a Website Structure with Petri Nets

A website is comprised of many web pages, where the web pages can be linked to from one another via hyperlinks. We will use places in PN to represent webpage, and transitions to hyperlinks.

Using PN as the website structure model, we can attain:

PN= (P, T, I, O, M_0),

Where

P stands for the web pages in the website;

T represents the hyperlinks in the web pages;

I denote the removed webpage volumes in the browser's pageviews after the hyperlinks are fired;

O denotes the added webpage volumes in the browser's pageviews after the hyperlinks are fired;

M_0 signifies the retrieved webpage in the browser after the first time the user enters the website.

In the parsing algorithm, we get the contents of the default root page first, and then traverse the web topology by visiting hyperlinks in all the web pages one by one to find the corresponding pageviews and construct the web structure accordingly.

There are three data structures, namely Pageview, PageviewQueue and VisitedPageviewTree. Pageview is a set of places, which represents the relevant information of web pages in the pageview. PageviewQueue keeps the pageview information about which pageview should be parsing next. By this structure, we implement a breadth-first constructing algorithm. VisetedPageviewTree maintains the tree of visited pageview, which is used to avoid repeatedly visiting the same pageview. There are five functions in this algorithm, namely GetWebPageContent, Parse, Generate_pageview, Generate_Transition and Generate_Aij. GetWebPageContent will get the content of the web page from a given web URL. Parse will find the HTML hyperlinks and its corresponding pageview from a given

web page content. By visiting hyperlinks in all the web pages one by one, Generate_pageview will generate the corresponding destination pageview and add new place if necessary. Generate_Transition and Generate_Aij will construct the web structure represented by incidence matrix Aij.

The parsing algorithm is shown in Figure 2.

```
Input: Pageview, a set of places.
Output: Aij, the incidence matrix of PN
Data Structure:
VisitedPageviewTree, maintain the tree of visited pageviews.
PageviewQueue, keep the pageview information about which pageview should be
parsing in the next step.
1.  Parsing_Algorithm (Pageview, Aij)
2.  home=Pageview;
3.  PageviewQueue.Enqueue(home);
4.  Do while  PageviewQueue.count > 0
5.      current_pageview = PageviewQueue.Dequeue();
6.      If VistedPageviewTree.Exist(current_pageview)=false Then
7.          VisitedPageviewTree.Add(current_pageview);
8.        For all places in current_pageview;      //current_place.
9.        web_page_URL=current_place.URL;
10.       web_page_content= GetWebPageContent(web_page_URL);
11.       Set_of_link=Parse(web_page_content);
12.          For each link in Set_of_link
13.              destination_pageview=Generate_Pageview(link,     current_pageview);
                 //add new place if necessary.
14.              transition=Generate_Transition(current_pageview,
                 destination_pageview);
15.              Generate_Aij(transition, Aij);
16.              PageviewQueue.Enqueue(destination_pageview);
17.           End For
18.     End For
19.    End If
20. Loop
```

Figure 2

The parsing algorithm

Taking the website represented in Table 1 as an example, Default.htm is the homepage of this website.

Table 1

A website example

Webpage names	The links in the webpage
1	A
	B
A	C
B	D

C	D<p>
	Default<p>
D	C<p>
Default	Index
Index	<frameset cols="20%,80%"> <frame src="1.htm" name="left" > <frame src="A.htm" name="right">

Table 2 illustrates the execution of the main loop in the parsing algorithm, ‘for all places in current_pageview(code# 8~18)’, where P# represents the place number of current_place in the current_pageview and T# is the transition number returned by Generate_Transition(), code# 19. Note that, the visited pageview (1,2,3)*, (6)*, (0)* and (5)* founded by the function VistedPageviewTree.Exist() are indicated by * in Table 2.

Table 2
The execution of main loop in the parsing algorithm

Current Pageview	P#	T#	Pageviw_Queue	Generate_Aij	
				Aij=-1	Aij=1
(0)	0	0	{ (1, 2, 3) }	A[0,0]	A[1,0] A[2,0] A[3,0]
(1, 2,3)	1		{ }		
	2	1	{ (1, 2, 4) }	A[1,1] A[2,1] A[3,1]	A[1,1] A[2,1] A[4,1]
	3	2	{ (1, 2,4), (5) }	A[1,2] A[2,2] A[3,2]	A[5,2]
(1, 2,4)	1		{ (5) }		
	2	3	{ (5),(1, 2,3) }	A[1,3] A[2,3] A[4,3]	A[1,3] A[2,3] A[3,3]
	4	4	{ (5),(1, 2,3), (6) }	A[1,4] A[2,4] A[4,4]	A[6,4]
(5)	5	5	{(1,2,3), (6), (6) }	A[5,5]	A[6,5]
		6	{(1,2,3),(6), (6),(0) }	A[5,6]	A[0,6]
(1,2,3)*			{(6), (6),(0) }		
(6)	6	7	{(6),(0),(5) }	A[6,7]	A[5,7]
(6)*			{(0) (5)}		
(0)*			{(5)}		
(5)*			{ }		

Table 3 shows the place added by the function Generate_Pageview() and its corresponding webpage. Table 4 shows the transition number generated by the function Generate_Transition() and its corresponding hyperlink.

Table 3
The place number and its corresponding webpage

Place #	Webpage names
0	Default
1	Index
2	1
3	A
4	B
5	C
6	D

Table 4
The transition number and its corresponding hyperlink

Transition #	Hyperink
0	Index
1	B
2	C
3	A
4	D
5	D<p>
6	Default<p>
7	C<p>

The web structure can be expressed by the incidence matrix, A_{ij}, in Figure 3, and its corresponding PN in Figure 4.

$$\lfloor A_{ij} \rfloor_{7\times 8} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0^* & -1 & 0^* & -1 & 0 & 0 & 0 \\ 1 & 0^* & -1 & 0^* & -1 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 \end{bmatrix}$$

Figure 3
The incidence matrix representing the webpage structure shown in Table 1.
(0* denotes that the values of the input and output function values of the place are equal when the transition is fired.)

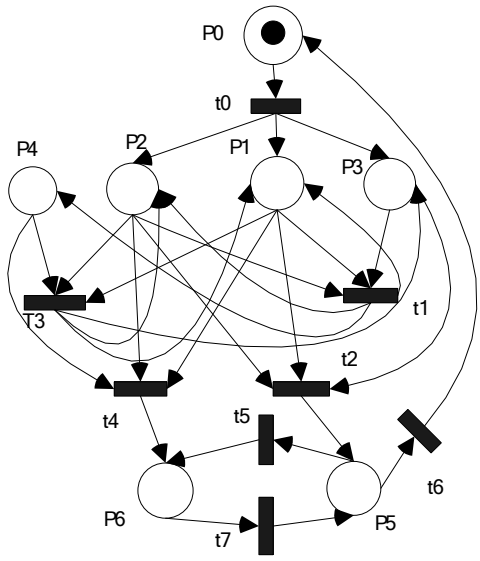

Figure 4
The Petri Nets corresponding to the website of Table 1

3 Using PN Model to Enhance Data Preprocessing

In the process of browsing a website, the user will only retrieve the first webpage through a direct key-in of the webpage's website address to request the webpage contents; very often the user's afterward browsed web pages are requested via service requests for new webpage contents toward the websites pointed to by the browser, according to the attributes of the hyperlinks, after the user clicks said hyperlinks within the browser. The browser will then display the related contents to the user based on the web pages' protocols after the requested web pages are transmitted to the user from the websites. Hence, the user's browsing process can be taken as an inter-webpage transition process.

The main task of web usage mining is to retrieve the information meaningful to the system management personnel from the web server's accumulated usage profiles left by all the browser users. As the profiles are only the sequentially recorded contents of the services provided by the web server, the profiles not only could contain multiple browsing profiles of different browser users but also could take in some extra or erroneous profiles. The website management personnel must proceed with preprocessing to these usage profiles if they are to correctly analyze said users' webpage-contents usage sequence. Hence, a data preprocessing is needed to enhance information processing before we can analyze the usage

profiles. The first step of data preprocessing often is to delete the erroneous or useless data or columns in the usage profiles via data cleaning; after finishing data cleaning, we next need to extract different users' usage profiles with user identification, using the user's IP column. Each user's website usage profile might include his multiple website usage records within a period of time; hence, we need to divide said user's usage profile into his multiple browsing session log files.

After completing said session identification, we still need to face problems related to path complete and pageview identification during data preprocessing. As we propose to use PN to model a website structure, we can further apply the incidence matrix and related properties obtained from the Petri Nets model to help proceed with path complete and pageview identification.

In the part of structure preprocessing, we first utilize a site spider to retrieve the website's webpage contents; we then use parsing program to analyze the webpage's contents to locate all places, i.e. the web pages and transitions, which are the hyperlinks causing the transitions; we also analyze the incidence matrix between the places and the transitions.

In the part of usage preprocessing, we will sequentially finish data cleaning, user identification, and session identification. In pageview identification, we will proceed with it using the pageview information provided by the incidence matrix. If missing paths are found during the identification process, we will activate the path complete process to locate the possible missing paths to carry out the path complete process and, then, continue working on pageview identification. The related component diagram are referred to Figure 5.

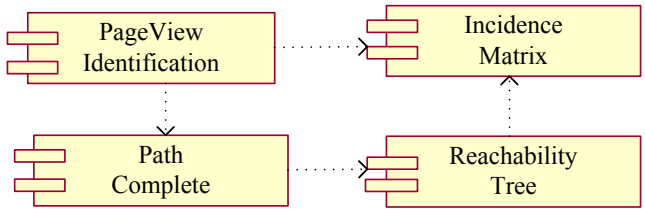

Figure 5
The component diagram of pageview identification and path complete

3.1 Pageview Identification

A pageview represents a display comprising all the webpages appearing concurrently in the browser during the process the user reads a webpage. The main function of pageview identification is to identify and mend the user session log file, with the help of the web structure information, to find out the real displayed contents and sequence in the browser during the user's browsing process. We adopt Petri Nets to model a website structure and proceed with pageview identification with the help of an incidence matrix.

Table 5
A user session before pageview identification

TID	Request file	Referring file
1	Default.htm	_
2	Index.htm	Default.htm
3	1.htm	Index.htm
4	A.htm	Index.htm
5	B.htm	1.htm
6	D.htm	B.htm
7	C.htm	D.htm

Taking a user profile as an example as shown in Table 5, we observe from the incidence matrix, as shown in Fig. 3, that when the system retrieves the first usage profile having a transition identification (TID) as 1, the transition related to P_0, hence, consists of T_0 only due to that the place corresponding to Default.htm is P_0 and that $[A_{0j}]$=(-1, 0, 0, 0, 0, 0, 1, 0); we can recognize that T_0 indicates the transformation of the pageview comprising Index.htm, 1.htm, and A.htm from the pageview containing Default.htm only, for $[A_{i0}]$=(-1, 1, 1 ,1, 0, 0, 0). The first, second and third usage profiles in Table 5 are Index.htm, 1.htm and A.htm, with a good match; hence, we can confirm that the three profiles represent the pageview comprising Index.htm, 1.htm and A.htm. As we proceed with the identification of the fifth data, we observe that T_1 stands for the transition to Index.htm, 1.htm, and B.htm, and T_3 the transition to C.htm as there are only two transitions, T_1 and T_3,related to Index.htm, 1.htm and A.htm; the request webpage of the fifth profile is B.htm. Hence, we can arrive at that the fifth profile represents the user's entering the pageview comprising Index.htm, 1.htm and B.htm. Based on such identification method, we can find out the sixth and seventh profiles representing, respectively, the user's entering the pageview comprising D.htm and C.htm; the user's profiles after completing the pageview identification are shown in Table 6.

Table 6
A user session after pageview identification

Pageview ID	Request file
1	Default.htm
2	Index.htm
2	1.htm
2	A.htm
3	Index.htm
3	1.htm
3	B.htm
4	D.htm
5	C.htm

3.2 Path Complete

Web Usage Logs are the records of the web content requests that all web users have made to that website. It's possible that web user encounters problem of Browser cache or Proxy server when requesting service from website. So it's not unusual some user's records are missing from the web usage logs. If it's not managed well, error will happen in Page identification process and, in turn, affect the correctness of web usage mining.

So Path Complete needs should be activated to patch it once the web usage logs are found incomplete during Pageview identification process

Taking the user session in Table 5 for example, if we lost the web logs from TID 2 to 5. After the pageview identification process identified the pageview of first transaction, it will find that the current pageview cannot transfer to next pageview directly. The process will launched the path complete process to find out the lost transition. Since we can have the initial state marking $M_0=[1, 0, 0, 0, 0, 0, 0]^T$ and the destination state marking $M_d=[0, 0, 0, 0, 0, 0, 1]^T$. According to (2), the equation can be illustrated as Figure 6.

$$\begin{bmatrix}0\\0\\0\\0\\0\\1\\0\end{bmatrix} = \begin{bmatrix}0\\1\\1\\0\\1\\0\\0\end{bmatrix} + \begin{bmatrix}-1 & 0 & 0 & 0 & 0 & 0 & 1 & 0\\ 1 & 0^* & -1 & 0^* & -1 & 0 & 0 & 0\\ 1 & 0^* & -1 & 0^* & -1 & 0 & 0 & 0\\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & -1 & -1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & -1 & -1 & 1\\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1\end{bmatrix} \times [T_k]$$

Figure 6
The state equation of path complete

We can also obtain the $[T_k]=[1, 1, 0, 0, 1, 0, 0, 0]^T$, which means that the initial state can be transferred to destination sate through continuous fired in T_0, T_1 and T_4. To take one step ahead, we can find the firing sequence of M_0 to M_d is T_0->T_1->T_4 in the help of incidence matrix.

Conclusions

A good-quality data preprocessing is one of the keys in determining if the web usage mining will be a success; however, relatively few researchers have expressed how to proceed with pageview identification and path complete in data preprocessing. In this paper, we propose the use of Petri Nets as a webpage structure model for website simulation and demonstrate that with this model, we

can not only adopt Petri Nests' incidence matrix to help carry out pageview identification but also utilize Petri Nests' reachability property to fulfill path complete.

References

[1] Robert Cooley: The Use of Web Structure and Content to Identify Subjectively Interesting Web Usage Patterns, ACM Transactions on Internet Technology, Vol. 3, No. 2, May 2003, pp. 93-116

[2] Berendt, B. Mobasher, M. Nakagawa, M. Spiliopoulou: The Impact of Site Structure and User Environment on Session Reconstruction in Web Usage Analysis, Proc. WEBKDD 2002: Mining Web Data for Discovery Usage Patterns and Profiles, LNCS 2703 Springer-Verlag, 2002

[3] W3C Web Characterization Terminology & Definitions Sheet, http://www.w3.org/1999/05/WCA-terms/

[4] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, Pan-Ning Tan: Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data, SIGKDD Explorations, Vol. 1, Issue 2, Jan. 2000, pp. 12-23

[5] Robert Cooley, Bamshad Mobasher, Jaideep Srivastava: Data Preparation for Mining World Wide Web Browsing Patterns, Journal of Knowledge and Information System, 1(1), 1999, pp. 5-32

[6] Murat Ali Bayir, Ismail H. Toroslu, Ahmet Cosar: A New Approach for Reactive Web Usage Mining Data Processing, Proceeding of the 22nd International Conference on Data Engineering Workshops (ICDEW'06)

[7] M. Spiliopoulou, B. Mobasher, B. Berendt, M. Nakagawa: A Framework for the Evaluation of Session Reconstruction Heuristics in Web Usage Analysis, INFORMS Journal on Computing, 2003

[8] Magdalini Eirinaki, Michalis Vazirgiannis: Web Mining for Web personalization, ACM Transactions on Internet Technology, Vol. 3, No. 1, Feb. 2003, pp. 1-27

[9] Jiacun Wang: Timed Petri Nets, Theory and Application, Boston: Kluwer Academic Publishers, 1998

[10] Tadao Murata: Petri Nets: Properties, Analysis and Applications, in Proceedings of the IEEE, Vol. 77, No. 4, 1989

[11] BinHong Wang: Markov Analysis for STPN Web Structure Model, Master thesis, Department of Computer Science and Information Engineering, Tamkang University, 2007, in Chinese

Stability Analysis Method for Fuzzy Control Systems Dedicated Controlling Nonlinear Processes

Marius-Lucian Tomescu

Computer Science Faculty, "Aurel Vlaicu" University of Arad
Complex Universitar M, Str. Elena Dragoi 2, RO-310330 Arad, Romania
E-mail: tom_uav@yahoo.com

Stefan Preitl, Radu-Emil Precup

Dept. of Automation and Applied Inf., "Politehnica" University of Timisoara
Bd. V. Parvan 2, RO-300223 Timisoara, Romania
E-mail: stefan.preitl@aut.upt.ro, radu.precup@aut.upt.ro

József K. Tar

Institute of Intelligent Engineering Systems, Budapest Tech Polytechnical Institution
Bécsi út 96/B, H-1034 Budapest, Hungary
E-mail: tar.jozsef@nik.bmf.hu

Abstract: This paper presents a new stability analysis method for nonlinear processes with Takagi-Sugeno (T-S) fuzzy logic controllers (FLCs). The design of the FLCs is based on heuristic fuzzy rules. The stability analysis of these fuzzy control systems is performed using LaSalle's invariant set principle with non-quadratic Lyapunov candidate function. This paper proves that if the derivative of Lyapunov function is negative semi-definite in the active region of each fuzzy rule, then the overall system will be asymptotically stable in the sense of Lyapunov (ISL). The stability theorem suggested in the paper ensures sufficient stability conditions for fuzzy control systems controlling a class of nonlinear processes. The end of the paper contains an illustrative example that describes an application of the stability analysis method.

Keywords: Fuzzy logic controller, Lyapunov stability, nonlinear system, LaSalle's invariant set principle

1 Introduction

The investigations of the stability of Takagi-Sugeno (T-S) fuzzy control systems begin before 1990 with increased frequency afterwards [1-5]. In principle, for the stability analysis of a fuzzy controller any method can be used witch is suitable for the analysis of nonlinear dynamic systems. Today, there exist preoccupations reported in the literature [6, 7] on the stability analysis and design of T-S fuzzy control systems. The majority of these papers is based on linear matrix inequality (LMI) framework [8] and the stability conditions of fuzzy control systems employs quadratic Lyapunov functions. In this case, there exist two shortcomings:

- first, the linearization can result in uncertainties and inaccuracies of the fuzzy models involved,
- second, using the quadratic Lyapunov functions the stability conditions become usually very restrictive.

This paper presents a new stability analysis method for nonlinear processes with T-S fuzzy logic controllers (FLCs) without process linearization and without using the quadratic Lyapunov functions in the derivation and proof of the stability conditions. The rest of the paper is organized as follows. Section 2 recalls the Takagi-Sugeno fuzzy control systems controlling nonlinear processes. Section 3 gives a stability theorem for nonlinear systems with T-S FLCs and an algorithm for the design of a stable fuzzy control system. An illustrative example presented in Section 4 shows that good control system performance can be obtained by applying the suggested algorithm. Section 5 concludes the paper.

2 One Class of Fuzzy Logic Control Systems

A fuzzy logic control system consists of a process and a fuzzy logic controller as shown in Figure 1. Let $X \subset R^n$ be a universe of discourse. The controlled process is accepted to be characterized by the class of single-input *n*-th order nonlinear system modelled by the state-space equations in (1):

$$\begin{aligned} &\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{b}(\mathbf{x})u, \\ &\mathbf{x}(t_0) = \mathbf{x}_0, \end{aligned} \tag{1}$$

where: $\mathbf{x} \in X$, $\mathbf{x} = [x_1 \quad x_2 \quad ... \quad x_n]^T$ is the state vector, $n \in IN^*$, $\dot{\mathbf{x}} = [\dot{x}_1 \quad \dot{x}_2 \quad ... \quad \dot{x}_n]^T$ is the derivative of $\mathbf{x}$ with respect to the time variable *t*, $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \quad f_2(\mathbf{x}) \quad ... \quad f_n(\mathbf{x})]^T$ and $\mathbf{b}(\mathbf{x}) = [b_1(\mathbf{x}) \quad b_2(\mathbf{x}) \quad ... \quad b_n(\mathbf{x})]^T$ are functions describing the dynamics of the process, *u* is the control signal fed to the process, obtained by the weighted-sum defuzzification method for T-S FLCs.

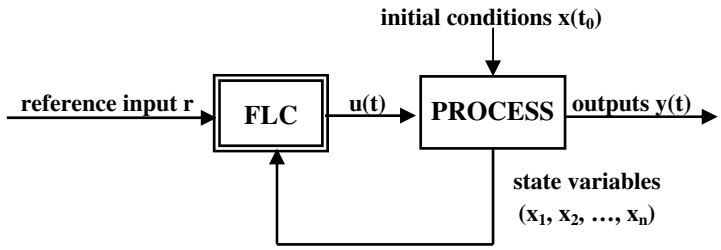

Figure 1
Fuzzy logic control system structure

The FLC consists of r fuzzy rules. The i-th IF–THEN rule in the fuzzy rule base of the FLC, referred to as Takagi-Sugeno fuzzy rule, has the following expression:

Rule i: IF x_i is $\tilde{X}_{i,1}$ AND … AND x_n is $\tilde{X}_{i,n}$ THEN $u = u_i(x)$, $i = \overline{1,r}$,

$r \in N, r \geq 2$, (2)

where X_{i1}, X_{i2} … X_{in} are fuzzy sets that describe the linguistics terms (LTs) of input variables, $u = u_i(x)$ is the control output of rule i, and the function AND is a t-norm. u_i can be a single value or a function of states vector, $x(t)$. Each fuzzy rule generates a firing degree $\alpha_i \in [0,1], i = 1,2,..,r$, according to (3):

$$\alpha_i(\mathbf{x}) = \min(\mu_{\tilde{X}_{i,1}}(x_1), \mu_{\tilde{X}_{i,2}}(x_2), ..., \mu_{\tilde{X}_{i,n}}(x_n)). \quad (3)$$

It is assumed that for any $\mathbf{x}$ belonging to the input universe of discourse, X, there exists at least one α_i among all rules that is not equal to zero.

The control signal u, which must be applied to the process, is a function of α_i and u_i. By applying the weighted-sum defuzzification method, the output of the FLC is given by:

$$u = \frac{\sum_{i=1}^{r} \alpha_i u_i}{\sum_{i=1}^{r} \alpha_i}. \quad (4)$$

Definition 1: For any input $x_0 \in X$, if the firing degree $\alpha_i(x_0)$ corresponding to fuzzy rule i is zero, this fuzzy rule i is called an inactive fuzzy rule for the input x_0; otherwise, it is called an active fuzzy rule.

It should be noted that with $x = x_0$, an inactive fuzzy rule will not affect the controller output $u(x_0)$. Hence (4) can be rewritten so as to consider all active fuzzy rules only,

$$u(\mathbf{x}_0) = \frac{\sum_{i=1,\alpha_i \neq 0}^{r} \alpha_i(\mathbf{x}_0) u_i(\mathbf{x}_0)}{\sum_{i=1,\alpha_i \neq 0}^{r} \alpha_i(\mathbf{x}_0)}. \tag{5}$$

Definition 2: An active region of a fuzzy rule *i* is defined as a set $X_i^A = \{\mathbf{x} \in X \,|\, \alpha_i(\mathbf{x}) \neq 0\}$.

3 Stability Analysis of Fuzzy Control Systems Controlling Nonlinear Processes

The stability analysis theorem presented here is based on LaSalle's invariant set principle called also global invariant set theorem and referred in [9]. The premise of the stability criterion in this paper is that, if the control output of each rule to fulfil the same conditions (presented in the next Theorem), the overall system will be stable ISL. The theorem ensures sufficient stability conditions for the fuzzy control systems with the structure described in Section 2. This Section is focused on Theorem 1 that can be expressed also as a useful stability analysis algorithm.

Let $V: R^n \to R$, $V(\mathbf{x}) > 0, \forall \mathbf{x} \neq 0$ be a scalar function with continuous first-order partial derivatives. The time derivative of $V(\mathbf{x})$ along the open-loop trajectory (1) is given by:

$$\begin{aligned} \dot{V}(\mathbf{x}) &= \frac{dV}{dt} = \sum_{i=1}^{n} \frac{\partial V}{\partial x_i} \frac{dx_i}{dt} = \sum_{i=1}^{n} \frac{\partial V}{\partial x_i} \left(f_i(\mathbf{x}) + b_i(\mathbf{x}) u \right) = \\ &= \sum_{i=1}^{n} \frac{\partial V}{\partial x_i} f_i(\mathbf{x}) + u \sum_{i=1}^{n} \frac{\partial V}{\partial x_i} b_i(\mathbf{x}) = F(\mathbf{x}) + B(\mathbf{x}) u, \end{aligned} \tag{6}$$

where:

$$F(\mathbf{x}) = \sum_{i=1}^{n} \frac{\partial V}{\partial x_i} f_i(\mathbf{x}) \tag{7}$$

and:

$$B(\mathbf{x}) = \sum_{i=1}^{n} \frac{\partial V}{\partial x_i} b_i(\mathbf{x}). \tag{8}$$

Now, we define:

$$B^0 = \left\{ \mathbf{x} \in X \,\middle|\, B(\mathbf{x}) = 0 \right\}, \tag{9}$$

$$B^+ = \left\{ \mathbf{x} \in X \,\middle|\, B(\mathbf{x}) > 0 \right\}, \tag{10}$$

$$B^- = \left\{ \mathbf{x} \in X \,\middle|\, B(\mathbf{x}) < 0 \right\}. \tag{11}$$

The main result of this paper is given by the following Theorem:

Theorem 1: Let the fuzzy control system consisting of the T-S FLC described in Section 2 and the nonlinear process with the state-space equations (1) with $\mathbf{x} = 0$ an equilibrium point. If there exists a function $V : X \to R$, $V(\mathbf{x}) > 0, \forall \mathbf{x} \neq 0$ with continuous first-order partial derivatives and:

1 $F(\mathbf{x}) \leq 0, \ \forall \mathbf{x} \in B^0$,

2 $u_i(\mathbf{x}) \leq -\dfrac{F(\mathbf{x})}{B(\mathbf{x})}$ for $\mathbf{x} \in X_i^A \bigcap B^+$ and $u_i(\mathbf{x}) \geq -\dfrac{F(\mathbf{x})}{B(\mathbf{x})}$ for $\mathbf{x} \in X_i^A \bigcap B^-$, $i = \overline{1, r}$,

3 the set $S = \left\{ \mathbf{x} \in X \,\middle|\, \dot{V}(\mathbf{x}) = 0 \right\}$ does no contain any state trajectory of the system except the trivial trajectory $\mathbf{x}(t) = 0$ for $t \geq 0$,

then the fuzzy control system is globally asymptotically stable ISL at the origin.

Proof

Further on, we will prove that the derivative of V with respect to time, $\dot{V}$, is negative semi-definite in terms of employing (1) in order to obtain the closed-loop system structure.

Consider an arbitrary input $\mathbf{x_0} \in X$. Then three cases will be considered as follows.

Case 1: If $\mathbf{x_0} \in X_i^A \bigcap B^+ \neq 0$ then $B(\mathbf{x}_0)$ is strictly positive. From the condition two of Theorem 1 it results that:

$$u_i(\mathbf{x_0}) \leq -\frac{F(\mathbf{x}_0)}{B(\mathbf{x_0})} \Rightarrow$$

$$\Rightarrow u(\mathbf{x_0}) = \frac{\sum_{i=1,\alpha_i \neq 0}^{r} \alpha_i(\mathbf{x_0}) u_i(\mathbf{x_0})}{\sum_{i=1,\alpha_i \neq 0}^{r} \alpha_i(\mathbf{x_0})} \leq \frac{-\dfrac{F(\mathbf{x_0})}{B(\mathbf{x_0})} \sum_{i=1,\alpha_i \neq 0}^{r} \alpha_i(\mathbf{x_0})}{\sum_{i=1,\alpha_i \neq 0}^{r} \alpha_i(\mathbf{x_0})} = -\frac{F(\mathbf{x_0})}{B(\mathbf{x_0})} \Rightarrow$$

$$\Rightarrow \dot{V}(\mathbf{x_0}) = F(\mathbf{x_0}) + B(\mathbf{x_0})u(\mathbf{x_0}) \le F(\mathbf{x_0}) + B(\mathbf{x_0})\left(-\frac{F(\mathbf{x_0})}{B(\mathbf{x_0})}\right) = 0. \quad (12)$$

Therefore, $u_i(\mathbf{x_0}) \le -\frac{F(\mathbf{x_0})}{B(\mathbf{x_0})} \Rightarrow \dot{V}(\mathbf{x_0}) \le 0, \forall \mathbf{x_0} \in X_i^A \bigcap B^+ \ne 0.$ (13)

Case 2: If $\mathbf{x_0} \in X_i^A \bigcap B^- \ne 0$ then $B(\mathbf{x_0})$ is strictly negative. From the condition two of Theorem 1 it results that:

$$u_i(\mathbf{x_0}) \ge -\frac{F(\mathbf{x_0})}{B(\mathbf{x_0})} \Rightarrow$$

$$\Rightarrow u(\mathbf{x_0}) = \frac{\sum_{i=1,\alpha_i \ne 0}^{r} \alpha_i(\mathbf{x_0})u_i(\mathbf{x_0})}{\sum_{i=1,\alpha_i \ne 0}^{r} \alpha_i(\mathbf{x_0})} \ge \frac{-\frac{F(\mathbf{x_0})}{B(\mathbf{x_0})}\sum_{i=1,\alpha_i \ne 0}^{r} \alpha_i(\mathbf{x_0})}{\sum_{i=1,\alpha_i \ne 0}^{r} \alpha_i(\mathbf{x_0})} = -\frac{F(\mathbf{x_0})}{B(\mathbf{x_0})} \Rightarrow$$

$$\Rightarrow \dot{V}(\mathbf{x_0}) = F(\mathbf{x_0}) + B(\mathbf{x_0})u(\mathbf{x_0}) \le F(\mathbf{x_0}) + B(\mathbf{x_0})\left(-\frac{F(\mathbf{x_0})}{B(\mathbf{x_0})}\right) = 0. \quad (14)$$

Therefore, $u_i(\mathbf{x_0}) \ge -\frac{F(\mathbf{x_0})}{B(\mathbf{x_0})} \Rightarrow \dot{V}(\mathbf{x_0}) \le 0, \forall \mathbf{x}_0 \in X_i^A \bigcap B^- \ne 0$. (15)

Case 3: If $\mathbf{x}_0 \in B^0$. In this case we have, from condition 3 of Theorem 1, that $F(\mathbf{x_0}) \le 0$. Therefore:

$$\dot{V}(\mathbf{x_0}) = F(\mathbf{x_0}) + B(\mathbf{x_0})u(\mathbf{x_0}) = F(\mathbf{x_0}) \le 0, \forall \mathbf{x_0} \in B^0 . \quad (16)$$

From above three cases one may conclude that $\dot{V}(\mathbf{x}) \le 0, \forall \mathbf{x} \in X$.

Summarizing, $\dot{V}$ is negative semi-definite.

Condition 3 of theorem ensures the fulfilment of LaSalle's invariant set principle. Both the condition of regarding the sign of $\dot{V}$ and the condition 3 satisfy the conditions from LaSalle's global invariant set theorem. Therefore, the equilibrium point at the origin is globally asymptotically stable. The proof is now complete ■

The above stability theorem ensures sufficient stability conditions regarding the accepted class of fuzzy control systems described briefly in Section 2.

Theorem 1 proves that if the Lyapunov function is negative semi-definite in the active region of each fuzzy rule then the overall system will be asymptotically ISL.

3.1 The Stability Analysis Algorithm

The stability analysis algorithm ensuring the stability of the class of fuzzy logic control systems considered in Section 2 is based on Theorem 1. It consists of the following steps:

1 *Determine the state-space equations of the nonlinear process,*

2 *Determine the membership function of the LTs in the T-S FLC structure,*

3 *Determine the premise of each fuzzy rule,*

4 *Set the* V *function, calculate its derivative and the expression of the functions* $F(x)$ *and* $B(x)$ *and the sets* B^0, B^+ *and* B^0 *as well,*

5 *If* $F(x) \le 0, \forall x \in B^0$ *then go to step 6. Else go to step 4.*

6 *For each fuzzy control rule i determine* u_i *such that* $u_i(\mathbf{x}) \le -\dfrac{F(\mathbf{x})}{B(\mathbf{x})}$ *for* $\mathbf{x} \in X_i^A \cap B^+$ *and* $u_i(\mathbf{x}) \ge -\dfrac{F(\mathbf{x})}{B(\mathbf{x})}$ *for* $\mathbf{x} \in X_i^A \cap B^-$, $i = \overline{1,r}$,

7 *Check that the set* $S = \{\mathbf{x} \in X \,|\, \dot{V}(\mathbf{x}) = 0\}$ *does not contain any state trajectory of the system except the trivial one,* $\mathbf{x}(t) = 0$ *for* $t \ge 0$.

4 Design Example

This Section presents an example that deals with one chaotic Lorenz system to be controlled by a Takagi-Sugeno FLC. Modern discussions of chaos are mainly based on the works about the Lorenz attractor. The Lorenz equation is commonly defined as three coupled ordinary differential equations expressed in (17) to model the convective motion of fluid cell, which is warmed from below and cooled from above:

$$\frac{dx}{dt} = \sigma(y - x), \quad \frac{dy}{dt} = x(\rho - z) - y, \quad \frac{dz}{dt} = xy - \beta z, \tag{17}$$

where the three parameters $\sigma, \rho, \beta > 0$ are called the Prandtl number, the Rayleigh number, and a physical proportion, respectively. These constants determine the behaviour of the system and these three equations exhibit chaotic behaviour i.e. they are extremely sensitive to initial conditions. A small change in initial conditions leads quickly to large differences in corresponding solutions.

The classic values used to demonstrate chaos are $\sigma = 10$ and $\beta = \frac{8}{3}$. Let $X = [-40, 40] \times [-40, 40] \times [-40, 40]$.

4.1 Design of Stable Fuzzy Logic Control System

The algorithm presented in Section 3 will be applied in the sequel in order to find the values of u_i for which the system (17) can be stabilized by the above described T-S FLC. A similar fuzzy logic control system has been designed in [10] but involving Barbashin-Krasovskii's theorem.

Step 1: The design of the fuzzy logic control system with TS FLC starts with rewriting the ordinary differential equation (17) in the following form representing the state-space equations of the controlled process with $x_1 = x$, $x_2 = y$ and $x_3 = z$:

$$\dot{\mathbf{x}} = \begin{pmatrix} \sigma(x_2 - x_1) \\ x_1(\rho - x_3) - x_2 \\ x_1 x_2 - \beta x_3 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} u, \quad \mathbf{x_0}(t) = \mathbf{x_0}. \tag{18}$$

Step 2: The first two equations are considered in the T-S FLC design. The fuzzification module of T-S FLC is set according to Figure 2 showing the membership functions that describe the LTs of the linguistic variables of x_1 and x_2. The LTs representing Positive, Zero and Negative values are noted by P, Z and N, respectively. The inference engine employs the fuzzy logic operator AND modelled by the *min* t-norm.

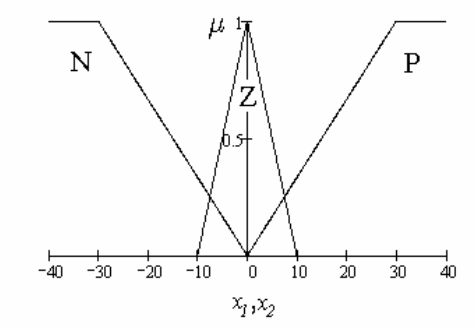

Figure 2
Membership functions of x_1 and x_2

Step 3: The inference engine is assisted by the complete set of fuzzy control rules illustrated in Table 1.

Table 1
Fuzzy Control Rule Base

Rule	Antecedent		Consequent
	x_1	x_2	u
1	P	P	u_1
2	N	N	u_2
3	P	N	u_3
4	N	P	u_4
5	P	Z	u_5
6	N	Z	u_6
7	Z	P	u_7
8	Z	N	u_8
9	Z	Z	u_9

Step 4: Let $V(\mathbf{x}) = \frac{1}{2}\left(x_1^2 + x_2^2 + x_3^2\right)$ be a Lyapunov function candidate, which is a continuously differentiable positive function on the domain *X*. The time derivative of *V* along the trajectories of the system (18) is given by:

$$\dot{V}(\mathbf{x}) = -\sigma x_1^2 - x_2^2 - \beta x_3^2 + x_1 x_2 (\sigma + \rho) + x_1 u . \tag{19}$$

Then (19) results in $F(\mathbf{x}) = -\sigma x_1^2 - x_2^2 - \beta x_3^2 + x_1 x_2 (\sigma + \rho)$, $B(\mathbf{x}) = x_1$ and

$B^0 = \left\{(0 \quad x_2 \quad x_3) \in R^3\right\}$, $B^+ = \left\{(x_1 \quad x_2 \quad x_3) \in R^3 \middle| x_1 > 0\right\}$,

$B^- = \left\{(x_1 \quad x_2 \quad x_3) \in R^3 \middle| x_1 < 0\right\}$.

Step 5: Since $F(\mathbf{x}) \le 0, \forall \mathbf{x} \in B^0$, the step 6 continues is applied.

Step 6: Further on, we will analyze each fuzzy control rule.

For rule 1: x_1 is P, x_2 is P and $X_1^A \times [-40,40] \cap B^+ = (0,40] \times (0,40] \times [-40,40]$, $X_1^A \cap B^- = \varnothing$. In this case we must have that $u_1(\mathbf{x}) \le -\frac{F(\mathbf{x})}{B(\mathbf{x})} = \sigma x_1 + \frac{x_2^2 + \beta x_3^2}{x_1} - x_2(\sigma + \rho)$. From this inequality we set $u_1(\mathbf{x}) = -x_2(\sigma + \rho)$.

For rule 2: x_1 is N, x_2 is N and $X_2^A \times [-40,40] \cap B^- = [-40,0) \times [-40,0) \times [-40,40]$, $X_2^A \cap B^+ = \varnothing$. In this case we must have that $u_2(\mathbf{x}) \ge -\frac{F(\mathbf{x})}{B(\mathbf{x})} = \sigma x_1 + \frac{x_2^2 + \beta x_3^2}{x_1} - x_2(\sigma + \rho)$. From this inequality we set $u_2(\mathbf{x}) = -x_2(\sigma + \rho)$.

For rule 3: x_1 is P, x_2 is N and $X_3^A \times [-40,40] \cap B^+ = (0,40] \times [-40,0) \times [-40,40]$, $X_3^A \cap B^- = \varnothing$. In this case we must have that $u_3(\mathbf{x}) \le -\frac{F(\mathbf{x})}{B(\mathbf{x})} = \sigma x_1 + \frac{x_2^2 + \beta x_3^2}{x_1} - x_2(\sigma + \rho)$. From this inequality we set $u_3(\mathbf{x}) = -1$.

For rule 4: x_1 is N, x_2 is P and $X_4^A \times [-40,40] \cap B^- = [-40,0) \times (0,40] \times [-40,40]$, $X_4^A \cap B^+ = \varnothing$. In this case we must have that $u_4(\mathbf{x}) \ge -\frac{F(\mathbf{x})}{B(\mathbf{x})} = \sigma x_1 + \frac{x_2^2 + \beta x_3^2}{x_1} - x_2(\sigma + \rho)$. From this inequality we set $u_4(\mathbf{x}) = 1$.

For rule 5: x_1 is P, x_2 is Z and $X_5^A \times [-40,40] \cap B^+ = (0,40] \times (-10,10) \times [-40,40]$, $X_5^A \cap B^- = \varnothing$. In this case we must have that $u_5(\mathbf{x}) \le -\frac{F(\mathbf{x})}{B(\mathbf{x})} = \sigma x_1 + \frac{x_2^2 + \beta x_3^2}{x_1} - x_2(\sigma + \rho)$. From this inequality we set $u_5(\mathbf{x}) = \sigma x_1 + \frac{x_2^2 + \beta x_3^2}{x_1} - 10(\sigma + \rho)$.

For rule 6: x_1 is N, x_2 is Z and $X_6^A \times [-40,40] \cap B^- = [-40,0) \times (-10,10) \times [-40,40]$, $X_6^A \cap B^+ = \varnothing$. In this case we must have that $u_6(\mathbf{x}) \ge -\frac{F(\mathbf{x})}{B(\mathbf{x})} = \sigma x_1 + \frac{x_2^2 + \beta x_3^2}{x_1} - x_2(\sigma + \rho)$. From this inequality we set $u_6(\mathbf{x}) = \sigma x_1 + \frac{x_2^2 + \beta x_3^2}{x_1} + 10(\sigma + \rho)$.

For rule 7: x_1 is Z, x_2 is P. Two cases should be considered:

a. $X_7^A \times [-40,40] \cap B^- = [-10,0) \times (0,40] \times [-40,40]$. In this case we must have that $u_7(\mathbf{x}) \ge -\frac{F(\mathbf{x})}{B(\mathbf{x})} = \sigma x_1 + \frac{x_2^2 + \beta x_3^2}{x_1} - x_2(\sigma + \rho)$.

b. $X_7^A \times [-40,40] \cap B^+ = (0,10] \times (0,40] \times [-40,40]$. In this case we must have that $u_7(\mathbf{x}) \le -\frac{F(\mathbf{x})}{B(\mathbf{x})} = \sigma x_1 + \frac{x_2^2 + \beta x_3^2}{x_1} - x_2(\sigma + \rho)$.

From both cases, set $u_7(\mathbf{x}) = -x_2(\sigma + \rho)$.

For **rules 8 and 9** a similar reasoning to that of rule 7 swill be applied with the result $u_8(\mathbf{x}) = u_9(\mathbf{x}) = u_7(\mathbf{x})$.

Step 8: We note that $\dot{V}_i(\mathbf{x}) = F(\mathbf{x}) + B(\mathbf{x})u_i$ and $S_i = \{\mathbf{x} \in X \,|\, \dot{V}_i(\mathbf{x}) = 0\}$. Use (4) result that $\dot{V}(\mathbf{x}) = \dfrac{\sum_{i=1}^{n} \alpha_i(\mathbf{x})\dot{V}_i(\mathbf{x})}{\sum_{i=1}^{n} \alpha_i(\mathbf{x})}$. We prove now that $S \subseteq \bigcup_{i=1}^{n} S_i$. We suppose that there exists $\mathbf{x}_0 \in S$. Then:

$$\dot{V}(\mathbf{x_0}) = 0 \Rightarrow \frac{\sum_{i=1}^{n} \alpha_i(\mathbf{x})\dot{V}_i(\mathbf{x})}{\sum_{i=1}^{n} \alpha_i(\mathbf{x})} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i(\mathbf{x})\dot{V}_i(\mathbf{x}) = 0. \tag{20}$$

It is important to highlight that the interpretation of (20) is that there exists at least one rule index i such that $\dot{V}_i(\mathbf{x}) = 0$. Therefore $S \subseteq \bigcup_{i=1}^{n} S_i$. Since $S_i = \varnothing$ for $i = \overline{1,8}$ and $S_9 = \{0\}$, the result is $S = \{0\}$. Thus, $S = \{\mathbf{x} \in X \,|\, \dot{V}(\mathbf{x}) = 0\}$ does not contain any state trajectory of the system except the trivial one, $\mathbf{x}(t) = 0$ for $t \geq 0$. Concluding, due to Theorem 1 it results that the system composed by this T-S FLC and the Lorenz process described by (18) is globally asymptotically stable ISL at the origin.

4.2 Simulation Results

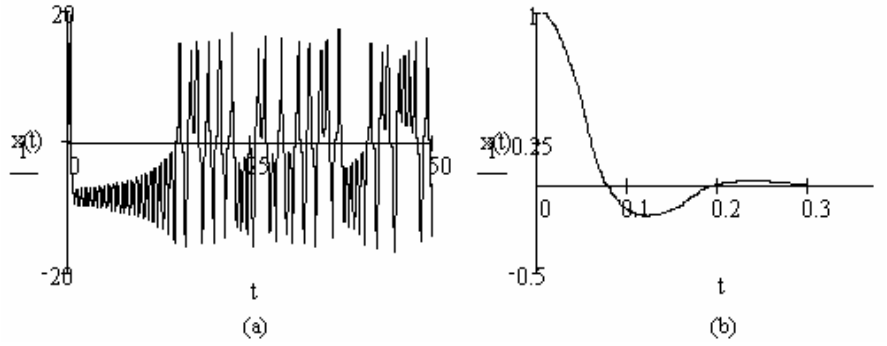

Figure 3
State variable x_1 versus time of Lorenz chaotic system without FLC (a) and with FLC (b)

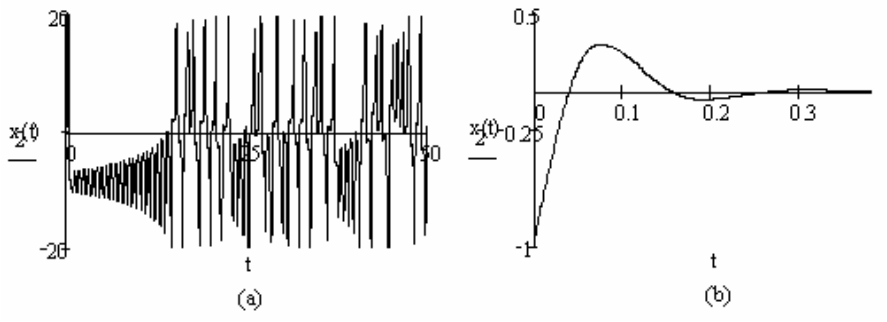

Figure 4
State variable x_2 versus time of Lorenz chaotic system without FLC (a) and with FLC (b)

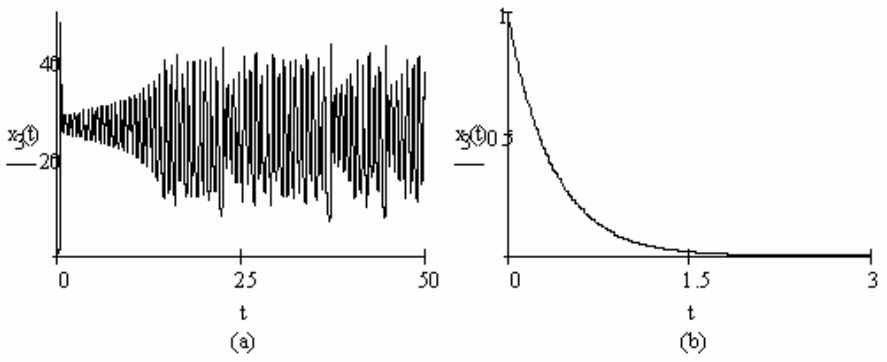

Figure 5
State variable x_3 versus time of Lorenz chaotic system without FLC (a) and with FLC (b)

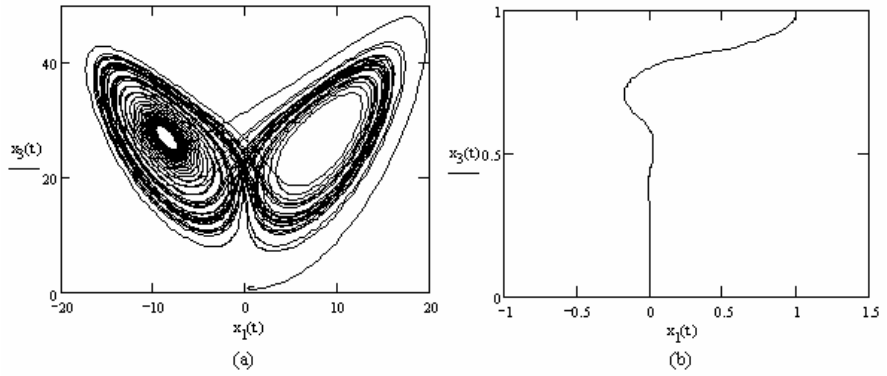

Figure 6
2D phase portraits of Lorenz system without control (a) and with FLC (b)

Considering the values of process parameters $\sigma = 10$, $\rho = 28$, $\beta = \frac{8}{3}$, the initial state $x_1(0) = 1$, $x_2(0) = -1$ and $x_3(0) = 1$, the responses of x_1, x_2 and x_3 versus time in the closed-loop system are illustrated in Figures 3 to 7.

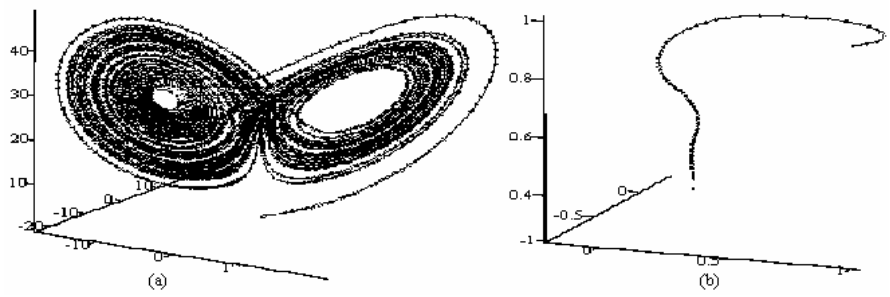

Figure 7
3D phase portrait Lorenz system without control (a) and with FLC (b)

Conclusions

A new approach to the globally asymptotically stability analysis of fuzzy control systems employing T-S FLCs dedicated to a class of nonlinear processes has been introduced. The new stability analysis approach is different to Lyapunov's theorem in several important aspects and allows more applications. In particular, it is well-suited to controlling processes where the derivative of the Lyapunov function candidate is not negative definite, therefore applying the LaSalle's invariant set principle to nonlinear processes controlled by T-S FLCs can be applied to a wide area of nonlinear dynamic systems. Using the proposed stability analysis approach makes the inserting of a new fuzzy rule (with the index r+1) become very easy because this needs only the fulfilment of the condition $\dot{V}_{r+1}(x) \le 0$.

The stability analysis algorithm proposed in this paper, based on Theorem 1, guarantees sufficient stability conditions for the fuzzy control systems. This algorithm can result in a design method, which is advantageous because the stability analysis decomposed to the analysis of each fuzzy rule. Therefore, the complexity of system analysis is reduced drastically.

This paper has shown, by the Lorenz system, how the stability analysis algorithm can be applied to the design of a stable fuzzy control system for a nonlinear process. Our stability approach can be applied also in situations when the system has an equilibrium point different to the origin and / or the reference input of the fuzzy control system is non-zero by appropriately defined state transforms.

Future research will be focused on increasing the area of applications [11-16]. But this must be accompanied by the derivation of transparent design methods for low-cost fuzzy logic controllers.

References

[1] K. Tanaka, M. Sugeno: Stability Analysis of Fuzzy Systems Using Lyapunov's Direct Method, Proceedings of NAPFIPS'90 Conference, Toronto, Canada, 1990, pp. 133-136

[2] R. Langari, M. Tomizuka: Analysis and Synthesis of Fuzzy Linguistic Control Systems, Proceedings of 1990 ASME Winter Annual Meeting, Dallas, TX, 1990, pp. 35-42

[3] S. Kitamura, T. Kurozumi T: Extended Circle Criterion, and Stability Analysis of Fuzzy Control Systems, Proceedings of International Fuzzy Engineering Symposium, Yokohama, Japan, 1991, pp. 634-643

[4] K. Tanaka, M. Sugeno: Stability Analysis and Design of Fuzzy Control Systems, in Fuzzy Sets and Systems, Vol. 45, No. 2, 1992, pp. 135-156

[5] S. S. Farinwata, G. Vachtsevanos: Stability Analysis of the Fuzzy Logic Controller Designed by the Phase Portrait Assignment Algorithm, Proceedings of Second IEEE International Conference on Fuzzy Systems, San Francisco, CA, 1993, pp. 1377-1382

[6] H. Ohtake, K. Tanaka, H. O. Wang: Piecewise Fuzzy Model Construction and Controller Design Based on Piecewise Lyapunov Function, Proceedings of American Control Conference, New York, NY, 2007, pp. 259-262

[7] H. K. Lam, F. H. F. Leung: Fuzzy Controller with Stability and Performance Rules for Nonlinear Systems, Fuzzy Sets and Systems, Vol. 158, No. 2, 2007, pp. 147-163

[8] K. Tanaka, H. O. Wang: Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach, John Wiley & Sons, New York, 2001

[9] J. J. E. Slotine, W. Li: Applied Nonlinear Control, Prentice-Hall, Englewood Cliffs, NJ, 1991

[10] M. L. Tomescu: Fuzzy Logic Controller for the Liénard System, Proceedings of 4th International Symposium on Applied Computational Intelligence and Informatics, SACI 2007, Timisoara, Romania, 2007, pp. 129-133

[11] L. Horváth, I. J. Rudas: Modeling and Problem Solving Methods for Engineers, Elsevier, Academic Press, Amsterdam, New York, 2004

[12] P. Baranyi, L. Szeidl, P. Várlaki, Y. Yam: Numerical Reconstruction of the HOSVD-based Canonical Form of Polytopic Dynamic Models,

Proceedings of 10th International Conference on Intelligent Engineering Systems, INES 2006, London, UK, 2006, pp. 196-201

[13] R.-E. Precup, S. Preitl, P. Korondi: Fuzzy Controllers with Maximum Sensitivity for Servosystems, IEEE Transactions on Industrial Electronics, Vol. 54, No. 3, 2007, pp. 1298-1310

[14] Zs. Cs. Johanyák, S. Kovács: Sparese Fuzzy System Generation by Rule Base Extension, Proceedings of 11th International Conference on Intelligent Engineering Systems, INES 2007, Budapest, Hungary, 2007, pp. 99-104

[15] G. Klančar, Škrjanc: Tracking-Error Model-based Predictive Control for Mobile Robots in Real Time, Robotics and Autonomous Systems, Vol. 55, No. 6, 2007, pp. 460-469

[16] J. Vaščák: Navigation of Mobile Robots by Computational Intelligence Means, Proceedings of 5th Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence and Informatics, SAMI 2007, Poprad, Slovakia, 2007, pp. 71-82