
Production Systems and Information Engineering
Volume 5 (2009), pp. 3-18.

ADAPTIVE APPROACHES TO DISTRIBUTED RESOURCE ALLOCATION

Balázs Csanád Csáji
Computer and Automation Research Institute,
Hungarian Academy of Sciences; and
Department of Mathematical Engineering,
Catholic University of Louvain, Belgium
balazs.csaji@sztaki.hu

László Monostori
Computer and Automation Research Institute,
Hungarian Academy of Sciences; and
Faculty of Mechanical Engineering,
Budapest University of Technology and Economics
laszlo.monostori@sztaki.hu

[Received March 2009 and accepted March 2009]

Abstract. The problem of allocating scarce, reusable resources over time to interconnected tasks in uncertain and changing environments, in order to optimize a performance measure, arises in many real-world domains. The paper examines several recent distributed optimization approaches to this problem and compares their properties, such as the guarantees of finding (near-)optimal solutions, their robustness against disturbances or against imprecise, uncertain models, with a special emphasis on adaptive capabilities. The paper also presents a reinforcement learning based distributed resource control system and argues that this method represent one of the most promising approaches to handling resource allocation problems in the presence of uncertainties.

Keywords: resource allocation, adaptive algorithms, distributed optimization, stochastic processes, reinforcement learning

1. Introduction

Efficient allocation of reusable resources over time is an important problem in many real-world applications, such as manufacturing production control (e.g. production scheduling), fleet management (e.g. freight transportation), personnel management, scheduling of computer programs (e.g. in massively parallel GRID systems), managing a construction project or controlling a cellular mobile network. In general, they can be described as optimization problems

which include the assignment of a finite set of scarce reusable resources to interconnected tasks that have temporal extensions.

The resource allocation related combinatorial optimization problems, such as the job-shop scheduling problem or the traveling salesman problem, are known to be strongly NP-hard, moreover, they do not have any good polynomial time approximation algorithm, either [1]. These problems have a huge literature, e.g. [2], however, most classical approaches concentrate on static and deterministic variants and their scaling properties are often poor. In contrast, real-world problems are usually very large, the environment is uncertain and can even change dynamically. Therefore, complexity and uncertainty seriously limit the applicability of classical solution methods.

In the past decades a considerable amount of research has been done to enhance decision-making, such as resource allocation, and several new paradigms have appeared that handled the problem in large-scale, dynamic and uncertain environments. Distributed decision-making is often favorable [3], not only because it can speed up the computation, but also because it can result in more robust and flexible solutions. For example, if we take a multi-agent based point of view combined with a heterarchical architecture, it can present several advantages [4], such as self-configuration, scalability, fault tolerance, massive parallelism, reduced complexity, increased flexibility, reduced cost and potentially emergent behavior [5].

The structure of the paper is as follows. First, a general *Resource Allocation Problem* (RAP) is specified. Then, a few widespread distributed resource allocation approaches are considered, and their key properties are investigated, with a special emphasis on their adaptive capabilities. Finally, a *Reinforcement Learning* (RL) based distributed RA system is presented and its properties are demonstrated by experimental results. RL-based resource allocation is argued to be one of the most promising approaches among the systems presented.

2. Resource Allocation Framework

First, a deterministic resource allocation problem is considered: an instance of the problem can be characterized by an 8-tuple $\langle \mathcal{R}, \mathcal{S}, \mathcal{O}, \mathcal{T}, \mathcal{C}, d, e, i \rangle$. In detail the problem consists of a set of reusable *resources* $\mathcal{R}$ together with $\mathcal{S}$ that corresponds to the set of possible *resource states*. A set of allowed *operations* $\mathcal{O}$ is also given with a subset $\mathcal{T} \subseteq \mathcal{O}$, which denotes the *target operations* or *tasks*. $\mathcal{R}$, $\mathcal{S}$ and $\mathcal{O}$ are supposed to be finite and they are pairwise disjoint. There can be *precedence constraints* between the tasks, which are represented by a partial ordering $\mathcal{C} \subseteq \mathcal{T} \times \mathcal{T}$. The *durations* of the operations depending on the state of the executing resource are defined by a *partial* function $d : \mathcal{S} \times \mathcal{O} \to \mathbb{N}$,

where $\mathbb{N}$ is the set of natural numbers, thus, we have a discrete-time model. Every operation can *affect* the state of the executing resource as well, that is described by $e : \mathcal{S} \times \mathcal{O} \to \mathcal{S}$, which is also a partial function. It is assumed that $dom(d) = dom(e)$, where $dom(\cdot)$ denotes the domain set of a function. Finally, the *initial states* of the available resources are given by $i : \mathcal{R} \to \mathcal{S}$.

The state of a resource can contain all relevant information about it, for example, its type and current setup (scheduling problems), its location and load (logistic problems) or its condition (maintenance and repair problems). Similarly, an operation can affect the state in many ways, e.g., it can change the setup of the resource, its location or its condition. The system must allocate each task (target operation) to a resource, however, there may be cases when first the state of a resource must be modified in order to be capable of executing a certain task (e.g. a transporter may first need to travel to its loading/source point, a machine may require repair or setup). In these cases non-task operations can be applied. They can modify the states of the resources without directly serving a demand (executing a task). It may be the case that during the resource allocation process a *non-task* operation is applied several times, but other *non-task* operations are completely avoided (for example because of their high cost). Nevertheless, finally all *tasks* must be completed.

A *solution* for a deterministic RAP is a partial function, the *resource allocator function*, $\varrho : \mathcal{R} \times \mathbb{N} \to \mathcal{O}$ that assigns the *starting times* of the operations to the resources. Note that the operations are supposed to be *non-preemptive* (they must not be interrupted).

A solution to a RAP is called *feasible* if the following properties are satisfied:

1. Each task is rendered to exactly one resource and start time:
$$\forall v \in \mathcal{T} : \exists\,! \langle r, t\rangle \in dom(\varrho) : v = \varrho(r, t)$$
2. All resources execute at most one operation at a time:
$$\neg\, \exists u, v \in \mathcal{O} : u = \varrho(r, t_1) \,\wedge\, v = \varrho(r, t_2) \,\wedge\, t_1 \le t_2 < t_1 + d(s(r, t_1), u)$$
3. The precedence constraints of the tasks are kept:
$$\forall \langle u, v\rangle \in \mathcal{C} : [u = \varrho(r_1, t_1) \,\wedge\, v = \varrho(r_2, t_2)] \Rightarrow [t_1 + d(s(r_1, t_1), u) \le t_2]$$
4. Every operation-to-resource assignment is valid:
$$\forall \langle r, t\rangle \in dom(\varrho) : \langle s(r, t), \varrho(r, t)\rangle \in dom(d)$$

where $s : \mathcal{R} \times \mathbb{N} \to \mathcal{S}$ describes the states of the resources at given time points,

$$s(r,t) = \begin{cases} i(r) & \text{if } t = 0 \\ s(r, t-1) & \text{if } \langle r, t\rangle \notin dom(\varrho) \\ e(s(r, t-1), \varrho(r, t)) & \text{otherwise.} \end{cases}$$

A RAP is called *correctly specified* if there exists at least one feasible solution. In what follows it is assumed that the problems are correctly specified. The set of all feasible solutions is denoted by $\mathbb{S}$. There is a performance (or cost) associated with each solution defined by a *performance measure* $\kappa : \mathbb{S} \to \mathbb{R}$ that often depends on the task completion times only. Typical performance measures that appear in practice include: maximum completion time or mean flow time. The aim of resource allocation is to compute a feasible solution with maximal performance (or minimal cost).

Note that the performance measure can assign penalties for violating *release* and *due* dates or even reflect the *priorities* of the tasks.

So far our model has been deterministic, now we turn to stochastic RAPs. The stochastic variant of the described general class of RAPs can be defined by randomizing functions d, e and i. Consequently, the operation durations become random, $d : \mathcal{S} \times \mathcal{O} \to \Delta(\mathbb{N})$, where $\Delta(\mathbb{N})$ is the space of probability distributions over $\mathbb{N}$. The effect of the operations is also uncertain, $e : \mathcal{S} \times \mathcal{O} \to \Delta(\mathcal{S})$ and the initial states of the resources can be stochastic as well, $i : \mathcal{R} \to \Delta(\mathcal{S})$. Note that the elements in the domain sets of functions d, e and i are probability distributions, we denote the corresponding random variables by D, E and I, respectively. We use the notation $X \sim f$ to indicate that random variable X has probability distribution f. Thus, $D(s, o) \sim d(s, o)$, $E(s, o) \sim e(s, o)$ and $I(r) \sim i(r)$ for all $s \in \mathcal{S}$, $o \in \mathcal{O}$ and $r \in \mathcal{R}$.

In stochastic RAPs the performance of a solution is also a random variable. Therefore, in order to compare the performance of different solutions we have to compare random variables. There are many ways in which this comparison can be made. For example, we can say that a random variable has stochastic dominance over another random variable 'almost surely', 'in likelihood ratio sense', 'stochastically', 'in the increasing convex sense' or 'in expectation'. In different applications various types of comparisons can be suitable, however, probably the most natural one is based upon the expected values of the random variables. The paper applies this kind of comparison.

Now, we classify the basic types of resource allocation techniques. In deterministic RAPs, there is no real difference between open- and closed-loop control. Thus, we can safely restrict ourself to open-loop methods. If the solution is aimed at generating the resource allocation off-line in advance, then it is called *predictive*. Thus, predictive solutions perform open-loop control and assume a deterministic environment. In stochastic resource allocation there are some data (e.g. the actual durations) that will only be available during the execution of the plan. According to the usage of this information, we identify two basic types of solution techniques. An open-loop solution that can deal with the uncertainties of the environment is called *proactive*. A proactive solution

allocates the operations to resources and defines the orders of the operations, but, because the durations are uncertain, it does not determine precise starting times. This kind of technique can be applied when only the durations of the operations are stochastic, but, the states of the resources are known in full (e.g. stochastic job-shop scheduling).

Finally, in the stochastic case a closed-loop solution to a RAP is called *reactive*. A reactive solution is allowed to make the decisions on-line, as the resource allocation process actually evolves and more information becomes available. Naturally, a reactive solution is not a simple ϱ function, but instead a resource allocation *policy* (a mapping from states to actions) which controls the process. Predictive and proactive RA has been investigated extensively over the past decades. The paper focuses on reactive resource allocation solutions only.

3. Distributed Resource Allocation

In this section a few widespread distributed resource allocation approaches will be considered and their key properties, such as the guarantees of finding an optimal (or a near optimal) solution, their robustness against different disturbances, such as breakdowns, or against imprecise, uncertain models, will be investigated, with a special emphasis on their adaptive capabilities.

A multi-agent system is a special distributed system with localized decision-making and, usually, localized storage. An agent is basically a self-directed (mostly software) entity with its own value system and a means to communicate with other such objects [4]. For a general survey on the application of multi-agent systems in manufacturing, see [6].

3.1. The PROSA Architecture

A basic agent-based architecture for manufacturing systems is PROSA [7]. The general idea underlying this approach is to consider both the resources (for example, machines and transporters) and the jobs (sets of interconnected tasks) as active entities. The standard architecture of the PROSA approach (see Figure 1) consists of three types of basic agents: order agents (internal logistics), product agents (process plans), and resource agents (resource handling). However, the PROSA architecture in itself is only a general framework and it does not offer any direct resource allocation solutions.

PROSA is a starting point for the design and development of multi-agent manufacturing control. Resource agents correspond to physical parts (production resources in the system, such as factories, shops, machines, furnaces, conveyors, pipelines, material storages, personnel, etc.), and contain an information processing part that controls the resource. Product agents hold the process

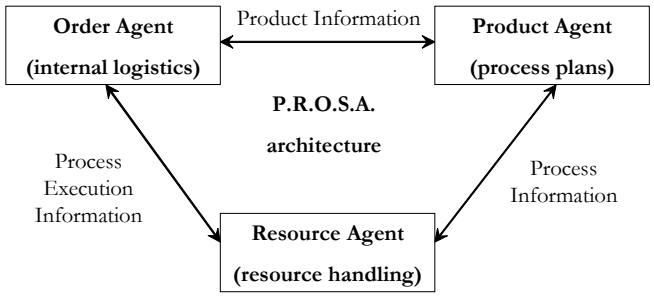


Figure 1. The PROSA reference architecture

and product knowledge to ensure the correct making of the product. They act as information server to other agents. Order agents represent a task or a job (an ordered set of tasks) in the manufacturing system. They are responsible for performing the assigned work correctly, effectively and on time.

3.2. Swarm Optimization

A great number of distributed optimization techniques were inspired by various biological systems [8], such as bird flocks, wolf packs, fish schools, termite hills or ant colonies. These approaches show up strongly robust and parallel behavior.

The ant colony optimization algorithm [9] is, in general, a randomized algorithm to solve *Shortest Path (SP)* problems in graphs. It can be shown that RAs can be formalized as special SP problems.

The PROSA architecture can also be extended by ant-colony-type optimization methods [10], in that case a new type of agent is introduced, called an ant. Agents of this type are mobile and they gather and distribute information in the manufacturing system. Their main assumption is that the agents are much faster than the ironware that they control, and that makes the system capable of prediction. Agents are faster and therefore can emulate the system's behaviour several times before the actual decision is taken.

The resource allocation in this system is made by local decisions. Each order agent sends out ants (mobile agents), which are moving downstream in a virtual manner. They gather information about the possible schedules from the resource agents and then return to the order agent with the information. The order agent chooses a schedule and then it sends ants to book the necessary resources. After that the order agent regularly sends booking ants to re-book the previously found best schedule, because if the booking is not refreshed, it

evaporates (like the pheromone in the analogy of food-foraging ants) after a while. From time to time the order agent sends ants to survey the possible new (and better) schedules. If they find a better solution, the order agent sends ants to book the resources that are needed for the new schedule and the old booking information will simply evaporate.

Swarm optimization methods are very robust, they can naturally adapt to environmental changes, since the agents continuously explore the current situation and the obsolete data simply evaporates if not refreshed regularly. However, these techniques often have the disadvantage that finding an optimal or even a relatively good solution cannot be easily guaranteed theoretically. For example, the ant-colony-based extension of PROSA faces almost exclusively the routing problem in resource allocation (how the tasks that belong to the same job should be processes through the machines) and it mostly ignores sequencing problems (the efficient ordering of the tasks that belong to different jobs). Therefore, the ant-colony-based extension of PROSA is very likely to be strongly sub-optimal, despite its very nice adaptive capabilities.

3.3. Negotiation-Based Approaches

There are multi-agent systems which use some kinds of negotiation or market-based mechanism in order to achieve efficient resource allocation [11]. In this case, the tasks or the jobs are associated with order agents, while the resources are controlled by resource agents, similarly to the PROSA architecture.

Market-based resource allocation is a recursive, iterative process with announce-bid-award cycles. During RA the tasks are announced to the agents that control the resources, and they can bid for the available jobs. A typical market-based system would work as follows: if a new job arrives at the system, a new order agent is created and associated with that job. An order agent or a group of cooperating order agents announces a sequence of operations and the resource agents can bid for that sequence. Only resource agents being able to do at least the first operation of that job are allowed to bid. Before an agent bids, it gathers information about the possible costs of making that sequence. If the sequence contains only one operation, the agent has all the information it needs, however, if the sequence contains other operations as well, which probably cannot be processed by the machine of the agent, it starts to search for subcontractors. It becomes a partial order agent and announces the remaining part of the sequence. The other resource agents, which can do the next operation, may bid for the remaining operation sequence. Consequently, a recursive announce-bid process begins. In the end, when all the possible costs of that (partial) job are known, the agent bids. If it made the highest bid (in a given time-frame), the agent (and its subcontractors) get the job.

A disadvantage is that during this mechanism, the jobs or tasks are, usually, announced one by one, which can lead to myopic behavior and, therefore, guaranteeing an optimal or even an approximately good solution is often very hard. Regarding adaptive behavior, market-based RA is often less robust than swarm-optimization methods, since, e.g. if a resource breaks down it is very likely that a large part of the negotiation process has to be restarted.

3.4. Problem Decomposition

The idea of divide-and-conquer is often applied in order to decrease computational complexity in combinatorial optimization problems. The main idea is to decompose the problem and solve the resulting sub-problems independently. In most cases calculating the sub-solutions can be done in a distributed way [12].

These approaches can be effectively applied in many cases, however, defining a decomposition which guarantees both efficient speedup together with the property that combining the optimal solutions of the sub-problems results in a global optimal solution is very demanding. Therefore, when we apply decomposition, we usually have to give up optimality and be satisfied with fast but sometimes far-from-optimal solutions. Moreover, it is hard to make these systems robust against disturbances. Tracking environmental changes can be often accomplished by the complete recalculation of the whole solution only.

3.5. Distributed Constraint-Satisfaction

Resource allocation problems (at least their deterministic variants) can be often formulated as constraint-satisfaction problems [13]. In this case, they aim at solving the problem formulated as follows:

$$\begin{aligned} \text{optimize} \quad & f(x_1, x_2, \ldots, x_n), \\ \text{subject to} \quad & g_j(x_1, x_2, \ldots, x_n) \leq b_j, \end{aligned}$$

where $x_i \in \Omega_i$, $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$. Functions f and g_j are real-valued and so are $b_j \in \mathbb{R}$. Most resource allocation problems, for example, resource constrained project scheduling, can be even formulated as a linear programming problem, which formulation can be written as

$$\begin{aligned} \text{optimize} \quad & c^T x, \\ \text{subject to} \quad & Ax \leq b, \end{aligned}$$

where $A \in \mathbb{R}^{m \times n}$, $x, c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and c^T denotes the transpose of c. Then, distributed variants of constrained optimization approaches can be used to

compute a solution. In that case, a close-to-optimal solution is often guaranteed, however, the computation time is usually large. The main problems with these approaches are that they cannot take uncertainties into account and, moreover, they are not robust against noises and disturbances.

4. Machine Learning and Resource Control

Machine learning techniques represent a promising new way to deal with resource allocation problems in complex, uncertain and changing environments. These problems can be often formulated as Markov decision processes and they can be solved by *Reinforcement Learning* (RL) algorithms [14, 15, 16, 17].

Now, we propose an RL-based adaptive sampler to compute an approximately optimal resource control policy in a distributed way. The sampling is done by iteratively simulating the resource control process. After each trial the policy is refined through recursive updates on the value function using the actual result of the simulation. Thus, from an abstract point of view, the optimization is accomplished through adaptive sampling. In order to achieve this, the RAP must be reformulated as a controlled Markov process.

4.1. Markov Decision Processes

Sequential decision making under uncertainty is often modelled by MDPs. This section contains the basic definitions and some preliminaries. By a (finite, discrete-time, stationary, fully observable) *Markov Decision Process* (MDP) we mean a stochastic system that can be characterized by an 8-tuple $\langle \mathbb{X}, \mathbb{T}, \mathbb{A}, \mathcal{A}, p, g, \alpha, \beta \rangle$, where the components are: $\mathbb{X}$ is a finite set of discrete *states*, $\mathbb{T} \subseteq \mathbb{X}$ is a set of *terminal states*, $\mathbb{A}$ is a finite set of control *actions*. $\mathcal{A} : \mathbb{X} \to \mathcal{P}(\mathbb{A})$ is the *availability function* that renders each state a set of actions available in that state where $\mathcal{P}$ denotes the power set. The *transition function* is given by $p : \mathbb{X} \times \mathbb{A} \to \Delta(\mathbb{X})$ where $\Delta(\mathbb{X})$ is the space of probability distributions over $\mathbb{X}$. Let us denote by $p(y|x, a)$ the probability of arrival at state y after executing action $a \in \mathcal{A}(x)$ in state x. The *immediate cost function* is defined by $g : \mathbb{X} \times \mathbb{A} \times \mathbb{X} \to \mathbb{R}$, where $g(x, a, y)$ is the cost of arrival at state y after taking action $a \in \mathcal{A}(x)$ in state x. We consider discounted MDPs and the *discount rate* is denoted by $\alpha \in [0, 1)$. Finally, $\beta \in \Delta(\mathbb{X})$ determines the *initial probability distribution* of the states in the stochastic system.

A (stationary, randomized, Markov) control *policy* is a function from states to probability distributions over actions, $\pi : \mathbb{X} \to \Delta(\mathbb{A})$. The initial probability distribution β, the transition probabilities p together with a control policy π completely determine the progress of the system in a stochastic sense, namely, it defines a homogeneous Markov chain on $\mathbb{X}$.

The *cost-to-go* function of a control policy is $Q^\pi : \mathbb{X} \times \mathbb{A} \to \mathbb{R}$, where $Q^\pi(x, a)$ gives the expected cumulative [discounted] costs when the system is in state x, it takes control action a and it follows policy π thereafter

$$Q^\pi(x, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \alpha^t G_t^\pi \;\middle|\; X_0 = x, A_0 = a \right], \tag{4.1}$$

where $G_t^\pi = g(X_t, A_t^\pi, X_{t+1})$, A_t^π is selected according to control policy π and the next state, X_{t+1}, has $p(X_t, A_t^\pi)$ probability distribution.

A policy $\pi_1 \leq \pi_2$ if and only if $\forall x \in \mathbb{X}, \forall a \in \mathbb{A} : Q^{\pi_1}(x, a) \leq Q^{\pi_2}(x, a)$. A policy is called *optimal* if it is better than or equal to all other control policies. The objective in MDPs is to compute a near-optimal policy.

There always exits at least one optimal (even stationary and deterministic) control policy. Although there may be many optimal policies, they all share the same unique optimal action-value function, denoted by Q^*. This function must satisfy a (Hamilton-Jacoby-) *Bellman type optimality equation* [18]:

$$Q^*(x, a) = \mathbb{E}\Big[g(x, a, Y) + \alpha \min_{B \in \mathcal{A}(Y)} Q^*(Y, B) \Big], \tag{4.2}$$

where Y is a random variable with $p(x, a)$ distribution.

From an action-value function it is straightforward to get a policy, for example, by selecting in each state in a greedy way an action producing minimal costs.

4.2. Adaptive Sampling

General RAPs with stochastic durations can be formulated as MDPs, as shown in [17]. Then, the challenge of finding a good policy can be accomplished by approximate Q-learning. In that case, the possible occurrences of the resource control process are iteratively simulated, starting from the initial stage of the resources. Each trial produces a sample trajectory that can be described as a sequence of state-action pairs. After each trial, the approximated values of the visited pairs are updated by the Q-learning rule.

The one-step Q-learning rule is $Q_{t+1} = TQ_t$, where

$$(TQ_t)(x, a) = (1 - \gamma_t(x, a))\, Q_t(x, a) + \gamma_t(x, a)(KQ_t)(x, a) \tag{4.3}$$

$$(KQ_t)(x, a) = g(x, a, Y) - Q_t(x, a) + \alpha \min_{b \in \mathcal{A}(Y)} Q_t(Y, b),$$

where Y and $g(x, a, Y)$ are random variables generated from the pair (x, a) by simulation, that is, according to probability distribution $p(x, a)$; the coefficients $\gamma_t(x, a)$ are called the *learning rate* and $\gamma_t(x, a) \neq 0$ only if (x, a) was visited during trial t. It is well-known [18] that if for all x and a:

$\sum_{t=1}^{\infty} \gamma_t(x,a) = \infty$ and $\sum_{t=1}^{\infty} \gamma_t^2(x,a) < \infty$, the Q-learning algorithm will converge with probability one to the optimal value function in the case of lookup table representation. Because the problem is acyclic, it is advised to apply *prioritized sweeping*, and perform the backups in an order opposite to which they appeared in during simulation, starting from a terminal state.

To balance between *exploration* and *exploitation*, and so to ensure the convergence of Q-learning, we can use the standard Boltzmann formula [18].

4.3. Cost-to-Go Approximation

In systems with large state spaces, the action-value function is usually approximated by a (typically parametric) function. Let us denote the space of action-value functions over $\mathbb{X} \times \mathbb{A}$ by $\mathbb{Q}(\mathbb{X} \times \mathbb{A})$. The method of fitted Q-learning arises when after each trial the action-value function is projected onto a suitable function space $\mathcal{F}$ with a possible error $\epsilon > 0$. The update rule becomes $Q_{t+1} = \Phi\, T\, Q_t$, where Φ denotes a projection operator to function space $\mathcal{F}$. In [17] support vector regression is suggested to effectively maintain the cost-to-go function. The value estimation then takes the form as follows

$$\tilde{Q}(x,a) = \sum_{i=1}^{l} (w_i^* - w_i) K(y_i, y) + b\,, \tag{4.4}$$

where K is an inner product kernel, $y = \phi(x,a)$ represents some peculiar features of x and a, w_i, w_i^* are the weights of the regression and b is a bias. As a kernel the usual choice is a Gaussian type function $K(y_1, y_2) = \exp(-\left\|y_1 - y_2\right\|^2 / \sigma^2)$ where $\sigma > 0$ is a user-defined parameter.

Partitioning the search space by decomposing the problem and applying limited-lookahead rollout algorithms in the initial stage can also speed up the computation of a near-optimal cost-to-go function considerably [17].

4.4. Distributed Sampling

In this section we investigate how the sampling presented can be distributed among several processors even if the value function is local to each processor.

If a common (global) storage is available to the processors, then it is straightforward to parallelize the sampling-based approximate cost-to-go function computation: each processor can search independently by making trials, however, they all share (read and write) the same global cost-to-go function. They update the value function estimations asynchronously.

A more complex situation arises when the memory is completely local to the processors, which is realistic if they are physically separated, e.g., in a GRID.

A way of dividing the computation of a good policy among several processors is possible when there is only one 'global' value function, however, it is stored in a distributed way. Each processor stores a part of the value function and it asks for estimations which it requires but does not have from the others. The applicability of this approach lies in the fact that the underlying MDP is acyclic and, thus, it can be effectively partitioned, for example, by starting the trials of each processor from a different starting state.

If the processors have their own completely local value functions, they can have widely different estimates on the optimal state-action values. In order to effectively compute a global value function, the processors should count how many times they updated the estimates of the different pairs. Finally, the values of the global Q-function can be combined from the individual estimates by a Boltzmann formula.

5. Experimental Results

In order to investigate our RL-based distributed resource control approach, numerical simulations were initiated and carried out.

First, the proposed approach was tested on Hurink's benchmark dataset [19]. It contains *Flexible Job-Shop (FJS)* scheduling problems with 6–30 jobs (30–225 tasks) and 5–15 resources. The performance measure is make-span, thus, the total completion time has to be minimized. These problems are 'hard', which means, e.g. that standard dispatching rules or heuristics perform poorly on them. This dataset consists of four subsets, each subset containing about 60 problems. The subsets (sdata, edata, rdata, vdata) differ in the ratio of resource interchangeability, shown in the 'flexib' column in Table 1. The other columns show the average error (avg err) and the standard deviation (std dev) after carrying out N iterations. The execution of 10 000 simulated trials (after on the average the system has achieved a solution with less than 5 % error) takes only a few seconds on an ordinary computer of today.

benchmark		**1000 iterations**		**5000 iterations**		**10 000 iterations**	
dataset	**flexib**	**avg err**	**std dev**	**avg err**	**std dev**	**avg err**	**std dev**
sdata	1.0	8.54 %	5.02 %	5.69 %	4.61 %	3.57 %	4.43 %
edata	1.2	12.37 %	8.26 %	8.03 %	6.12 %	5.26 %	4.92 %
rdata	2.0	16.14 %	7.98 %	11.41 %	7.37 %	7.14 %	5.38 %
vdata	5.0	10.18 %	5.91 %	7.73 %	4.73 %	3.49 %	3.56 %
average	2.3	11.81 %	6.79 %	8.21 %	5.70 %	4.86 %	4.57 %

Table 1. Performance (average error and deviation) on benchmark datasets

We initiated experiments on a simulated factory by modelling the structure of a real plant producing customized mass-products. We used randomly generated orders (jobs) with random due dates. The tasks and the process-plans of the jobs, however, covered real products. In this plant the machines require product-type dependent setup times, and another specialty of the plant is that, at some previously given time points, preemptions are allowed. The performance measure applied was to minimize the number of late jobs and an additional secondary performance measure was to minimize the total cumulative lateness, which can be applied to comparing two situations having the same number of late jobs. In Table 2 the convergence speed (average error and standard deviation) relative to the number of resources and tasks is demonstrated. The workload of the system was approximately 90 %. The results show that the suggested resource control algorithm can perform efficiently on large-scale problems, e.g. with 100 resources and 10 000 tasks.

configuration		**1000 iterations**		**5000 iterations**		**10 000 iterations**	
machs	**tasks**	**avg err**	**std dev**	**avg err**	**std dev**	**avg err**	**std dev**
6	30	4.01 %	2.24 %	3.03 %	1.92 %	2.12 %	1.85 %
16	140	4.26 %	2.32 %	3.28 %	2.12 %	2.45 %	1.98 %
25	280	7.05 %	2.55 %	4.14 %	2.16 %	3.61 %	2.06 %
30	560	7.56 %	3.56 %	5.96 %	2.47 %	4.57 %	2.12 %
50	2000	8.69 %	7.11 %	7.24 %	5.08 %	6.04 %	4.53 %
100	10000	15.07 %	11.89 %	10.31%	7.97 %	9.11 %	7.58 %

Table 2. Performance relative to the number of machines and tasks

We also investigated the parallelization of the method, namely, the speedup of the system relative to the number of processors. The average number of iterations was studied until the system could reach a solution with less than 5 % error on Hurink's dataset. We treated the average speed of a single processor as a unit (cf. with the data in Table 1). In Figure 2 the horizontal axis represents the number of processors applied, while the vertical axis shows the relative speedup achieved. We applied two kinds of parallelization: in the first case (dark gray bars), each processor could access a global value function. It means that all of the processors could read and write the same global action-value function, but otherwise, they searched independently. In that case the speedup was almost linear. In the second case (light gray bars), each processor had its own, completely local action-value function and, after the search was finished, these individual functions were combined. The experiments show that the computation of the RL-based resource control can be effectively distributed

even if there is not a commonly accessible action-value function available and each processor works locally with its own estimates.

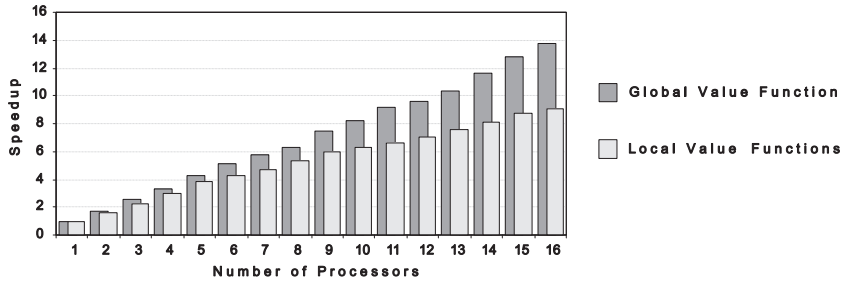


Figure 2. Distributed sampling: speedup relative to the number of processors

6. Concluding Remarks

Efficient allocation of scarce, reusable resources over time in uncertain and dynamic environments is an important problem that arises in many real world domains, such as production control. The paper examined some distributed RA approaches and presented an RL-based adaptive solution as well. The effectiveness of the latter approach was demonstrated by results of numerical simulation experiments on both benchmark and industry-related data.

There are several advantages why RL-based solutions are preferable to other kinds of distributed approaches described above. These favorable features are:

1. RL methods are *robust*, they essentially handle the problem under the presence of uncertainties, since they apply the theory of MDPs.
2. They can quickly *adapt* to unexpected changes in the environmental dynamics, such as breakdowns. This property can be explained by the Lipschitz type dependence of the optimal value function on the transition-probabilities and the immediate-cost function [20].
3. There are theoretical *guarantees* of finding optimal (or approximately optimal) solutions, at least asymptotically, in the limit.
4. Moreover, the actual convergence *speed* is usually high, especially in the case of applying distributed sampling or problem decomposition.
5. Additionally, the resulting distributed RL-based resource allocation *scales well* with the size of the problem. It can effectively handle large-scale problems without dramatic retrogression in the performance.
6. Finally, the proposed method constitutes an *any-time* solution, since the sampling can be stopped after any number of iterations.

Consequently, RL approaches have great potentials in dealing with real-world RAPs, since they can handle large-scale problems even in dynamically changing and uncertain environments. They seem to be one of the most promising approaches for distributed resource allocation in real-world domains.

Acknowledgements

The research presented was partially supported by the Hungarian Scientific Research Fund (OTKA) through the project "Production Structures as Complex Adaptive Systems". The paper also presented research results of the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office, and a grant Action de Recherche Concertée (ARC) of the Communauté Française de Belgique. The scientific responsibility rests with its authors. Balázs Csanád Csáji acknowledges the postdoctoral fellowship of the Université catholique de Louvain. The authors express their thanks to Tamás Kis for his contribution to the tests on industrial data.

REFERENCES

[1] Williamson, D. P., A., H. L., Hoogeveen, J. A., Hurkens, C. A. J., Lenstra, J. K., Sevastjanov, S. V., and Shmoys, D. B.: Short shop schedules. *Operations Research*, **45**, (1997), 288–294.

[2] Pinedo, M.: *Scheduling: Theory, Algorithms, and Systems*. Prentice-Hall, 2002.

[3] Perkins, J. R., Humes, C., and Kumar, P. R.: Distributed scheduling of flexible manufacturing systems: Stability and performance. *IEEE Transactions on Robotics and Automation*, **10**, (1994), 133–141.

[4] Baker, A. D.: A survey of factory control algorithms that can be implemented in a multi-agent heterarchy: Dispatching, scheduling, and pull. *Journal of Manufacturing Systems*, **17**, (1998), 297–320.

[5] Ueda, K., Márkus, A., Monostori, L., Kals, H. J. J., and Arai, T.: Emergent Synthesis Methodologies for Manufacturing. *Annals of the CIRP – Manufacturing Technology*, **50**, (2001), 535–551.

[6] Monostori, L., Váncza, J., and Kumara, S. R. T.: Agent-based systems for manufacturing. *Annals of the CIRP*, **55**(2), (2006), 697–720.

[7] Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., and Peeters, P.: Reference architecture for holonic manufacturing systems: PROSA. *Computers in Industry*, **37**, (1998), 255–274.

[8] Kennedy, J. and Eberhart, R. C.: Particle swarm optimization. *IEEE International Conference on Neural Networks*, **4**, (1995), 1942–1948.

[9] Moyson, F. and Manderick, B.: The collective behaviour of ants: an example of self-organization in massive parallelism. In *Proceedings of AAAI Spring Symposium on Parallel Models of Intelligence*, Stanford, California, 1988.

[10] Hadeli, Valckenaers, P., Kollingbaum, M., and Van Brussel, H.: Multi-agent coordination and control using stigmergy. *Computers in Industry*, **53**, (2004), 75–96.

[11] Márkus, A., Kis, T., Váncza, J., and Monostori, L.: A market approach to holonic manufacturing. *Annals of the CIRP*, **45**, (1996), 433–436.

[12] Wu, T., Ye, N., and Zhang, D.: Comparison of distributed methods for resource allocation. *International Journal of Production Research*, **43**, (2005), 515–536.

[13] Modi, P. J., Hyuckchul, J., Tambe, M., Shen, W., and Kulkarni, S.: Dynamic distributed resource allocation: Distributed constraint satisfaction approach. In *Pre-proceedings of the 8th International Workshop on Agent Theories, Architectures, and Languages*, 2001, pp. 181–193.

[14] Zhang, W. and Dietterich, T.: A reinforcement learning approach to job-shop scheduling. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 1995, pp. 1114–1120.

[15] Ueda, K., Hatono, I., Fujii, N., and Vaario, J.: Reinforcement learning approaches to biological manufacturing systems. *Annals of the CIRP – Manufacturing Technology*, **49**, (2000), 343–346.

[16] Aydin, M. E. and Öztemel, E.: Dynamic job-shop scheduling using reinforcement learning agents. *Robotics and Autonomous Systems*, **33**, (2000), 169–178.

[17] Csáji, B. C. and Monostori, L.: Adaptive stochastic resource control: A machine learning approach. *Journal of Artificial Intelligence Research (JAIR)*, **32**, (2008), 453–486.

[18] Bertsekas, D. P.: *Dynamic Programming and Optimal Control*. Athena Scientific, 2nd edn., 2001.

[19] Hurink, E., Jurisch, B., and Thole, M.: Tabu search for the job-shop scheduling problem with multi-purpose machines. *Operations Research Spectrum*, **15**, (1994), 205–215.

[20] Csáji, B. C. and Monostori, L.: Value function based reinforcement learning in changing Markovian environments. *Journal of Machine Learning Research (JMLR)*, **9**, (2008), 1679–1709.


Production Systems and Information Engineering
Volume 5 (2009), pp. 19-39.

SEMANTIC REPRESENTATION OF NATURAL LANGUAGE WITH EXTENDED CONCEPTUAL GRAPH

Erika Baksa-Varga
University of Miskolc, Hungary
Department of Information Technology
vargae@iit.uni-miskolc.hu

László Kovács
University of Miskolc, Hungary
Department of Information Technology
kovacs@iit.uni-miskolc.hu

[Received February 2009 and accepted April 2009]

Abstract. We intend to create an intelligent agent that is able to detect the objects and relations in its environment, and based on the received information is able to build up an internal knowledge base on the basis of which linguistic expressions can be formulated. The present investigations focus on modeling the agent's ability to assign semantic representations to its observations. For this purpose we have developed a graphical conceptual modeling language. In this article we examine the model's expressive power, that is to what extent it is able to model natural language semantics. The analysis is performed in comparison with predicate logic. In our view, first-order logic does not provide sufficient means for the translation of natural language expressions and we will argue that also higher-order logic needs some extensions.

Keywords: natural language semantics, semantic representation of languages, predicate logic, Extended Conceptual Graph

1. Introduction

The final goal of our research is to simulate a human agent who perceives signals from the environment and after processing the received information, is able to express the observations with linguistic symbols. We intend to create an intelligent agent – having the cognitive abilities of pattern recognition, association and generalization – that is able to detect the objects and relations in its environment, and based on the received information is able to build up an internal knowledge base on the basis of which linguistic expressions can be formulated. The schematic design of the system can be found in Figure 1.

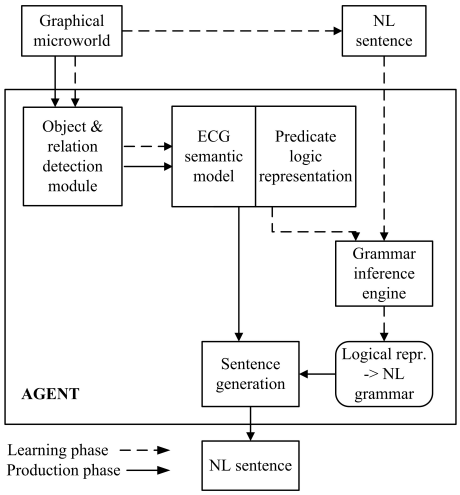


Figure 1. Operational system design of the agent

The project is strongly related to natural language processing (NLP). In NLP, the inference and adequate representation of natural language (NL) semantics is a crucial issue. A relatively new discipline, computational semantics has the aim to find techniques for automatically constructing semantic representations for expressions of human language. For this, the most basic issue is to agree on a semantic representation language. Practically, this should be a formalism with high expressive power and computational effectiveness. "In philosophy and linguistics the predicate calculus is used for analyzing the semantics and logic of natural language... The way expressions and structures contribute to the meaning of a natural language sentence is supposed to be determined and shown by means of its translation into the calculus [1]." The advantages of predicate logic (PL) are the use of a simple and exact notation and interpretation system, the standard formalism and general applicability, the ability for reasoning and rule validation, and its convertability to other symbolisms.

Despite the extensive research on representing NL semantics with PL, the relationship of the two systems is still not fully revealed. It is primarily a consequence of the NL system's complexity and ambiguity. Opinions in the related literature are quite diverse in terms of the suitability of first-order versus higher-order logic for NL representation. For example, [2] argues that first-order predicate logic (FOPL) offers an attractive compromise between the conflicting demands of expressivity and inferential effectiveness, and shows techniques for translating away modalities from intentional and temporal linguistic phenomena, as well as for handling plurals in FOPL. Also, [3] states

that FOPL is sufficient, since higher-order logical statements can be converted to FOPL formulas. On the other hand [4] and [5] prove the necessity of higher-order predicate logic (HOPL) in the field of quantification, while [1] makes a point on the unsuitability of FOPL in terms of representing plural referring expressions. In our view, FOPL does not provide sufficient means for the translation of NL expressions (see the analysis in Section 3) and in this article we will argue that also HOPL needs some extensions.

The article is organized as follows. In Section 2 we give the formal definition of the syntax and semantics of first-order predicate logic, then that of natural language. In Section 3 we examine the semantic equivalence of the NL and PL notation systems. For this purpose, we define first the semantic equivalence of two statements introducing the notion of composition preserving transformation. After that we analyze the properties of a semantic equivalence assignment between the two systems. In Section 4 we argue that first-order logic does not provide sufficient means for the representation of NL semantics, therefore we need to go beyond it. Higher-order logic is introduced, and we show that it also needs some extensions. In Section 5 we define an extended HOPL model and study the semantic equivalence assignment between this model and the NL system. In Section 6 we introduce the Extended Conceptual Graph (ECG) model as the graphical representation of the given extended HOPL model. Finally, we sum up the results of the present investigations concerning the expressiveness of the ECG model.

2. Formalizing Semantic Interpretation

2.1. Formal Definition of First-Order Predicate Logic

FOPL is a flexible, well-understood and computationally tractable approach to knowledge representation [6], which uses a wholly unambiguous formal language interpreted by mathematical structures. It is a system of deduction that extends propositional logic by allowing quantification over individuals of a given domain of discourse. The syntax of FOPL is built up of a vocabulary consisting of non-logical and logical symbols over a Σ alphabet. The set of non-logical symbols includes function symbols with a fixed arity ≥ 0, and the collection of variable and constant symbols. The set of logical symbols comprises predicate symbols with a fixed arity ≥ 0, the boolean connectives ($\wedge$, $\vee$, $\neg$, $\rightarrow$), and the quantifiers ($\forall$ and $\exists$). As its name implies, FOPL is organized around the notion of predicate. Predicates are symbols that refer to the relations that hold among some fixed number of objects in a given domain. Objects are represented by terms, which can be defined as constants, functions

or variables. FOPL constants refer to exactly one object, and are conventionally depicted as single capitalized letters. Functions also refer to unique objects, while variables, which are normally denoted by single lower-case letters, allow us to make statements about unnamed objects (free variables) and also to make statements about some/all objects in some arbitrary world being modelled (bound variables in the scope of a quantifier). Formally,

- all variable symbols are terms;
- if $t_1, \ldots, t_n$ are terms and f is a function symbol with arity n, then $f(t_1, \ldots, t_n)$ is also a term.

A statement is expressed in the form of formulas, which are defined as follows:

- If p is a predicate symbol with arity n, and $t_1, \ldots, t_n$ are terms, then $p(t_1, \ldots, t_n)$ is an atomic formula.
- If t_1 and t_2 are terms, then $t_1 = t_2$ is an atomic formula.
- If α and β are formulas, then so are $\neg\alpha$, $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, and $(\alpha \rightarrow \beta)$.
- If α is a formula, and x is a variable, then both $\forall x.\alpha$ and $\exists x.\alpha$ are formulas.
- A sentence is a formula without free variables.

The syntax of FOPL defines the set of well-formed formulas (WFFs), while the semantics of FOPL determines the truth value of an arbitrary formula in a given model or interpretation (an abstract realization of a situation). Formally, an interpretation $\mathfrak{I} = < \Delta, I >$ consists of a domain Δ and an assignment function I which assigns

- an f^I function with arity n to every function symbol f with arity n, where: $f^I : \Delta \times \ldots \times \Delta \mapsto \Delta$, and
- a p^I relation with arity n to every predicate symbol p with arity n, where: $p^I \subseteq \Delta \times \ldots \times \Delta$.

With the aid of interpretation an element of Δ can be assigned to every variable-free expression. Similarly, a truth value can be assigned to every sentence. For the interpretation of expressions with variables, and formulas with free variables a variable assignment function is required. This φ function assigns an element of Δ to each variable symbol x, so that $\varphi(x) \in \Delta$. Given an interpretation $\mathfrak{I} = < \Delta, I >$ and a variable assignment φ, the $t^{\varphi,I}$ meaning of an arbitrary term t is defined as:

- If x is a variable, then $x^{\varphi,I} = \varphi(x)$.
- If $t_1, \ldots, t_n$ are terms and f is a function symbol with arity n, then $f(t_1, \ldots, t_n)^{\varphi,I} = f^I(t_1^{\varphi,I}, \ldots, t_n^{\varphi,I})$.

Given an interpretation $\mathfrak{I} = < \Delta, I >$ and a variable assignment φ, the truth value of an arbitrary α formula is defined as $\mathfrak{I} \models_\varphi \alpha$, that is the interpretation satisfies the formula. Regarding the different types of formulas this definition is the following:

- $\mathfrak{I} \models_\varphi p(t_1, \ldots, t_n)$ iff $\langle d_1, \ldots, d_n \rangle \in p^I$ and $d_i = t_i^{\varphi, I}$.
- $\mathfrak{I} \models_\varphi t_1 = t_2$ iff $d_1, d_2 \in \Delta$ and for both $d_i = t_i^{\varphi, I}$ where $d_1 = d_2$.
- $\mathfrak{I} \models_\varphi \neg\alpha$ iff not $\mathfrak{I} \models_\varphi \alpha$.
- $\mathfrak{I} \models_\varphi \alpha \wedge \beta$ iff $\mathfrak{I} \models_\varphi \alpha$ and $\mathfrak{I} \models_\varphi \beta$.
- $\mathfrak{I} \models_\varphi \alpha \vee \beta$ iff $\mathfrak{I} \models_\varphi \alpha$ or $\mathfrak{I} \models_\varphi \beta$.
- $\mathfrak{I} \models_\varphi \alpha \rightarrow \beta$ iff not $\mathfrak{I} \models_\varphi \alpha$ or $\mathfrak{I} \models_\varphi \beta$.
- $\mathfrak{I} \models_\varphi \forall x.\alpha$ iff for all $d \in \Delta$ $\mathfrak{I} \models_{\varphi[x \mapsto d]} \alpha$.
- $\mathfrak{I} \models_\varphi \exists x.\alpha$ iff for some $d \in \Delta$ $\mathfrak{I} \models_{\varphi[x \mapsto d]} \alpha$.

Where $\varphi[x \mapsto d]$ is the variable assignment which assigns $d \in \Delta$ to x, while assigning the same value to every other variable as φ [7].

2.2. Formal Definition of Natural Language

The term 'natural language' (NL) refers to human languages which are not consciously invented, but naturally acquired by humans. All written natural languages build up of sequences of words which are finite sequences of symbols over a given alphabet. Syntax is the term which defines the set of rules telling us how words may be combined to form sentences. Formally,

- the main building blocks of NLs are characters (symbols) $c \in \Sigma$, where Σ denotes the finite character set (alphabet) of the language;
- words are finite sequences over Σ, that is every $w \in W \subseteq \Sigma^*$, where W denotes the set of words;
- sentences are finite sequences over W, that is every $s \in S \subseteq W^*$, where S denotes the set of sentences.

NLs are infinite recursive systems, hence on the basis of understanding a finite number of words we can understand and construct an infinity of sentences recursively applying the rules of syntax [8].

NL semantics is concerned with the relation between language and the 'world'. Hence, the meaning of a sentence determines the conditions under which it is true. Since, by definition sentences are finite sequences of words (which are the basic semantic units), and as a consequence of the recursive nature of language, the meaning of a word will determine what contribution it makes to the truth conditions of the sentences in which it occurs [2]. This is called

the principle of compositionality. For correct interpretation, however, we also need to have world knowledge. Without context, that is without defining the domain of discourse, many human language sentences could be assigned several meanings. This ambiguity may result from the lexical ambiguity of words, or from the syntactic ambiguity of sentences (word combinations). In other words, NL sentences build up of word constituents bearing a set of possible meanings which are made concrete by the actual context.

Thus, analogously to FOPL syntax and semantics, the syntax of NL defines the set of well-formed grammatical sentences (WGSs), while its semantics determines the truth value of a WGS in a given interpretation. An interpretation $\mathfrak{I} = < D_O, I >$ consists of a domain of objects D_O and an assignment function I which assigns

- an f^I function with arity n to every function symbol f with arity n, where: $f^I : D_{O_1} \times \ldots \times D_{O_m} \mapsto D_O$, and
- a p^I relation with arity n to every predicate symbol p with arity n, where: $p^I \subseteq D_{O_1} \times \ldots \times D_{O_m}$.

With the aid of interpretation an element of D_O can be assigned to every WGS constituent.

3. Examining the Semantic Equivalence of NL and PL Systems

3.1. Formal Definition of Semantic Equivalence

By definition, two statements in the same system are logically equivalent if, for all possible values of the variables involved, both statements are true or both are false. If α and β are equivalent, we write $\alpha \equiv \beta$. Formally, given an interpretation $\mathfrak{I} = < \Delta, I >$ and a variable assignment φ, formula α and formula β are equivalent if

- $\mathfrak{I} \models_\varphi \alpha$ and $\mathfrak{I} \models_\varphi \beta$, or
- not $\mathfrak{I} \models_\varphi \alpha$ and not $\mathfrak{I} \models_\varphi \beta$.

3.2. Semantic Equivalence of NL and FOPL Statements

In this section we intend to examine the semantic equivalence of NL and FOPL statements with 'true' logical value. Following from the definition of logical equivalence, this examination requires an interpretation $\mathfrak{I} = < \Delta, I >$, a variable assignment φ and two formulas: an NL WGS and an FOPL WFF. Let us suppose that the content words (those with lexical meaning) constituting

the NL sentence comprise the interpretation domain. The proposition(s) expressed by the NL sentence will be the predicate(s) of the FOPL formula, while the other constituents will be assigned to FOPL variables. It is easy to see that without interpretation and variable assignment the equivalence of the next two statements is undecidable.

1/a Jeg elsker deg.
1/b P(x, y).

Assuming you do not speak Norwegian (i.e. without interpretation), you are not able to understand 1/a, and hence you are not able to find out its truth value. Also, without interpretation and variable assignment 1/b can be rendered into any two-place predicate. Now, if we give the interpretation of the predicate and assign values to the variables, we get

2/a I love you.
2/b Love(I, you).

However, we are still not able to determine the statements' truth value, and hence their equivalence. This is because NL sentences are produced and perceived in concrete communicative contexts. Thus, "it is not just what a sentence means, but the fact that someone utters it plays a role in determining what its utterance conveys" [9]. In this case we need to specify who is referred to as 'I' and who is referred to as 'you'. Owning this knowledge then allows us to decide whether the two statements are semantically equivalent or not.

Another important question in our discussion is to what extent rendering NL sentences into logical notation should reflect the logical forms of those sentences. That is to say, "it is one thing for a sentence to be rendered into a logical formula, and quite another for the sentence itself to have a certain logical form" [9]. The difference is evident if we consider the following examples.

3/a There are students.
3/b $(\exists x)S(x)$.

4/a Some students are foreigners.
4/b $(\exists x)(S(x) \wedge F(x))$.
4/c $(\exists x)(F(S(x)))$.

3/b reveals the true structure of the existential proposition 3/a expresses. Thus this logical form of the sentence shows inherent properties of the sentence itself, therefore we can refer to it as a level of syntactic structure. On the other hand, 4/a does not express existential proposition and it does not

contain any sentential constituent corresponding to the conjunction in 4/b. On the other hand, 4/c is not a valid FOPL statement although its approach mirrors the true structure of the NL sentence.

In our discussion, we restrict our attention to logical forms reflecting syntactic structure. For handling discrepancies, we give the definition of the equivalence of two different notation systems by introducing the definition of a composition preserving transformation. Given two languages $L_1(F_1, O_1)$ and $L_2(F_2, O_2)$, where F denotes the set of formulas and O denotes the set of operations over F, the transformation $\tau : L_1 \to L_2$ is said to be composition preserving if

$$\tau(o(f_1, f_2, ...)) \equiv \tau(o)(\tau(f_1), \tau(f_2), ...), \tag{3.1}$$

i.e. $\tau(o(f_1, f_2, ...))$ and $\tau(o)(\tau(f_1), \tau(f_2), ...)$ are equivalent in all interpretations.

Without the criterion of composition preserving, an $ST(w_1, w_2, ...)$ general FOPL predicate could be assigned to any arbitrary $s = w_1, w_2, ...$ NL sentence (see example 5/c). In this case however, the semantic interpretation of the FOPL formula is not easier than that of the NL sentence.

3.3. Semantic Equivalence Assignment

The previous section defines what we mean by the semantic equivalence of two statements. If the two statements are in the same set of statements (A), then semantic equivalence is a binary relation over the given set, denoted by $R \subseteq A \times A$. If R is reflexive, symmetric and transitive, then is said to be an equivalence relation. In this section we consider two sets of statements: let NL denote the set of well-formed grammatical NL sentences and $FOPL$ denote the set of well-formed FOPL formulas. Examine the properties of semantic equivalence over both sets.

1. R is reflexive, if $\forall a \in A$ (aRa) holds.
2. R is symmetric, if $\forall a, b \in A$ $(aRb \Rightarrow bRa)$ holds. Thus, if a is the semantic equivalent of b, then the opposite is also true.
3. R is transitive, if $\forall a, b, c \in A$ $(aRb \wedge bRc \Rightarrow aRc)$ holds. Thus, if a is the semantic equivalent of b and b is the semantic equivalent of c, it entails that a is the semantic equivalent of c.

Taking the sets of NL sentences and FOPL formulas all three properties evidently hold, therefore semantic equivalence can be considered as equivalence relation over both sets. An equivalence relation divides a set into a number of non-empty, pairwise disjoint subsets (equivalence classes). The statement

sets constructed from these semantic equivalence classes are denoted by NL/R and $FOPL/R$, respectively. We define a semantic equivalence assignment n between these two sets as $n : NL/R \rightarrow FOPL/R$. Now we focus on studying the properties of this assignment in view of the criterion of composition preserving.

1. n is a mapping, if $\forall s \in NL/R$, $\exists f \in FOPL/R$ so that $(s, f) \in n$, and $\forall s \in NL/R$, $\forall f_1, f_2 \in FOPL/R$ $((s, f_1), (s, f_2) \in n \Rightarrow f_1 = f_2)$. Thus, every NL sentence should have a corresponding FOPL formula.
2. n is injective, if $\forall s_1, s_2 \in NL/R$, $\forall f \in FOPL/R$ $((s_1, f), (s_2, f) \in n \Rightarrow s_1 = s_2)$. Thus, every FOPL formula can have only one corresponding NL sentence (but it is not necessary to have any).
3. n is surjective, if $\forall f \in FOPL/R$, $\exists s \in NL/R$ so that $(s, f) \in n$. Thus, every FOPL formula should have one corresponding NL sentence.
4. n is bijective, if n is injective and surjective.

First, we need to prove that n is a mapping. Consider the following examples.

5/a You know I like sports.
5/b Know(you, Like(I, sports)).
5/c KnowLike(you, I, sports).

Here, the logical counterpart 5/b of the NL sentence is not a valid FOPL formula, because predicates are not allowed to be arguments of other predicates. On the other hand, 5/c is a well-formed FOPL formula, but it is not composition preserving.

6/a Most students like sports.
6/b (Most x: S(x)) Like(x, sports).

Although in this case the logical form respects the structural integrity of the quantified noun phrase, it is not a standard FOPL statement. Barwise and Cooper [5] have shown that the notation of FOPL is not adequate for symbolizing such quantificational expressions as 'most', 'many', 'several', 'few' (not mentioning numerical quantifiers and more complex quantificational expressions).

7/a Students travel home regularly.
7/b $(\forall x : S(x))$ Happens(regularly, Travel(x, home)).

The logical form here is not a valid statement even in extended versions of FOPL. The reason for this is that in FOPL it is not allowed to quantify over predicates.

Although we have not considered all linguistic phenomena, we could find some that cannot be represented in standard FOPL at all, or not with the precondition that we would like to keep the structure of the NL sentence. Therefore we can state that $n : NL/R \rightarrow FOPL/R$ is not a mapping. If we restrict though the set of NL sentences to those that can be represented in FOPL, we can prove that n' is still not an unambiguous mapping. Consider the next example.

8/a Every student read a book *(over the vacation).*
8/b $(\forall x)(\exists y(S(x) \wedge B(y) \wedge R(x,y))$.
8/c $(\exists y)(\forall x(S(x) \wedge B(y) \wedge R(x,y))$.

We can render the NL sentence either as 'every student read a separate book' as in 8/b, or as 'every student read the same book' as in 8/c. This phenomena is known as scope ambiguity, and results from the fact that NL, in opposition to FOPL, is structurally ambiguous [8]. As a consequence of the possibility of these kinds of multiple assignments n' is said to be a multivalued mapping. On the other hand, n" : $FOPL/R \rightarrow$ *restricted* NL/R would be a surjective mapping if we ignore the criterion of composition preserving. From this analysis we can conclude that the semantic content set $FOPL/R$ is able to cover is narrower than that of NL/R.

4. Higher-Order Logic

We can go beyond FOPL in two directions. On the one hand, we can introduce calculuses of higher order, in which propositions or propositional functions (and therefore sets) can appear as arguments to other functions. On the other hand, we can use higher (constructive and nonconstructive) methods like recursive numerical functions, symbolic structures, and semantic methods. In some ways intermediate between these are systems in which numbers are explicitly introduced (as primitives) into the domain of arguments [10]. From the above examples and examinations it is clear that we emphasize the introduction of higher-order calculus and numerical primitives.

4.1. Formal Definition of HOPL

The most obvious differences between HOPL and FOPL are that 1) HOPL uses variables that range over sets instead of discrete variables; and 2) in

HOPL predicates can be arguments of predicates and values of variables (i.e. quantification over predicates is allowed). In other words, higher-order logics allow for quantification not only of elements of the domain of discourse, but subsets of the domain of discourse, sets of such subsets, and other objects of higher type (such as relations between relations, functions from relations to relations between relations, etc.). The semantics are defined so that, rather than having a separate domain for each higher-type quantifier to range over, the quantifiers instead range over all objects of the appropriate type. Although higher-order logics are more expressive, allowing complete axiomatizations of structures, they do not satisfy analogues of the completeness and compactness theorems from first-order logic, and are thus less amenable to proof-theoretic analysis [11].

According to [12], a common approach to describing the syntax of a higher-order logic is to introduce some kind of typing scheme. One approach types first-order individuals with ι, sets of individuals with $\langle\iota\rangle$, sets of pairs of individuals with $\langle\iota\iota\rangle$, sets of sets of individuals with $\langle\langle\iota\rangle\rangle$, etc. Such a typing scheme, however, does not provide types for function symbols. A more general approach to typing is that used in the Simple Theory of Types [13]. Here again, the type ι is used to denote the set of first-order individuals, and the type o is used to denote the sort of booleans: {true, false}. In addition to these two types, it is possible to construct functional types: if σ and τ are types, then $\sigma \to \tau$ is the type of functions from objects of type σ to objects of type τ. Thus, an expression of type $\iota \to \iota$ represents a function from individuals to individuals. Similarly, an expression of type $\iota \to o$ represents a function from individuals to the booleans. Using characteristic functions to represent predicates, this latter type is used as the type of predicates whose one argument is an individual. Similarly, an expression that is of type o is defined to be a formula. Typed expressions are built by

- application, i.e. if M is of type $\sigma \to \tau$ and N is of type σ, then their application MN is of type τ, and
- abstraction, i.e. if x is a variable of type σ and M is of type τ, then the abstraction λxM is of type $\sigma \to \tau$.

Propositional connectives can be added to these terms by introducing the constants $\wedge$, $\vee$ and $\supset$ of type $o \to o \to o$ and $\neg$ of type $o \to o$. Quantification arises by adding (for each type σ) the constants $\forall_\sigma$ and $\exists_\sigma$ both of type $(\sigma \to o) \to o$. The intended denotation of $\forall_\sigma$ is the set that contains one element, namely the set of all terms of type σ; while the intended meaning of $\exists_\sigma$ is the collection of all non-empty subsets of type σ.

Higher-order logic can be interpreted over a pair $\langle \{D_\sigma\}_\sigma, J \rangle$, where σ ranges over all types. The set D_σ is the collection of all semantic values of type σ and J maps constants to particular objects in their typed domain. There are two major ways to interpret higher-order logic. A *standard model* is one in which the set $D_{\sigma\to\tau}$ is the set of all functions from D_σ to D_τ. Such models are completely determined by supplying only D_ι and J. If D_ι is denumerably infinite, then $D_{\iota\to o}$ is uncountable, thus standard models can be very large. As a consequence of Gödel's incompleteness theorem, the set of true formulas in such a model are not recursively axiomizable; i.e. there is no theorem proving procedure that could (even theoretically) uncover all true formulas [14]. In the *general (or Henkin) model* [15], however, it is possible for $D_{\sigma\to\tau}$ to be a proper subset of the set of all functions from D_σ to D_τ as long as there are enough functions to properly interpret all expressions of the language of type $\sigma \to \tau$. Hence this model is sound and complete.

4.2. Reasons for HOPL

The necessity of HOPL in representing NL semantics is proved in view of the arguments against it. Firstly, reification [6] is a technique used for representing all concepts that one wants to make statements about as objects in FOPL, instead of using higher-order predicates. In this case, however, new relations need to be introduced which in fact do not solve, but only shift the problem. Moreover, the resulting valid FOPL formulas will not be in accordance with the precondition of composition preserving. Secondly, [3] states that FOPL is sufficient, since HOPL formulas can be converted into FOPL formulas. In the proposed formalism, an arbitrary $P_1(P_2(x))$ second-order statement can be transformed into a $P_1(y_1) \wedge P_2(y_2) \wedge PR(y_1, y_2, x)$ FOPL formula; while $\forall p.P(x)$ is rendered into $\forall y.P(y) \to PR(y, x)$. This solution formally results in valid FOPL formulas, but the criterion of composition preserving is violated, which leads to the following problems. First, consider the example under 5. In 5/c we lose the syntactic structure of the NL sentence. This FOPL formula ignores the subordination relation between the NL constituents: all elements are at the same level, and the original structure is obscured. This problem can be eliminated by the use of higher-order predicates. In general, a higher-order predicate of order n takes one or more $(n-1)^{th}$-order predicates as arguments, where $n > 1$. Now, consider another example.

9/a I drink cold milk regularly.
9/b Happens(regularly, Drink(I, cold, milk)).
9/c Drink(I, regularly, cold, milk).

In 9/c not only the criterion of composition preserving is violated but one may draw the false conclusion that 'regularly' refers solely to the predicate 'drink'. In fact, it refers to the whole statement (see also example 7).

Example 6/b is another argument for the necessity of HOPL, since FOPL restricts the use of quantifiers to $\exists$ and $\forall$. A contemporary theory of quantification is the so-called Generalized Quantifier theory [4], [5]. This theory introduces many primitive quantifier expressions, as well as symbols for handling counting quantifiers.

For further information about second-order logic and its comparison with first-order logic we refer the reader to [16], [17] and [18]. For details about modal logic [19], [20], many-valued logic and fuzzy logic see [21].

Since variables are connected to domains in PL, HOPL expressions need to be embedded (see 5/b, 7/b). However, if we consider 9/b we can see that HOPL is not an adequate formalism either, because the structure under the predicate 'drink' is obscured: one may improperly conclude that 'cold' refers to 'drink' rather than 'milk'. Consequently the formalism needs further extensions.

5. Extension of HOPL

As we have declared previously, our project aims at developing an intelligent agent (see Figure 1) that is able to express its observations and states in NL sentences. In the first stage, we restrict the expressions to the observations which are related to definite, unambiguously interpretable situations. Consequently, the sentences describing these situations are factual assertions with true logical value. Therefore the following linguistic phenomena are beyond the scope of our investigations:

- if-then structures and conditionals,
- imperative, optative, exclamatory and interrogative sentences,
- probability and other certainty/uncertainty factors,
- intentional secondary meaning (pragmatics).

On the other hand, linguistic phenomena that need to be studied are as follows:

- domain types,
- referring expressions,
- adverbs,
- adjectives,

– numerical expressions and cardinality,
– quantification,
– logical connectives,
– historical (temporal) sequences, and
– causality.

For the composition preserving logical representation of the examined linguistic phenomena we propose the following HOPL extensions.

1) Arbitrary predicates (relations) are allowed, denoted by capitalized words. Domain types are assigned to the arguments of predicates, which specify the semantic roles these arguments play. The fixed set of roles (analogously to thematic roles [22] in linguistics) are associated with and determined by the predicate.

1.1/a Peter loves Mary.
1.1/b Love(Subject: Peter, Object: Mary).

2) Concepts are regarded as sets. Constants referring to specific objects (concepts) are single-element sets denoted by capitalized words, while constants referring to abstract concepts are multiple-element sets denoted by lower-case words. An element of a set a is denoted by the $isa(a)$ function (functions are denoted by lower-case words). We can refer to an object by the : operator.

2.1/a Peter reads a book.
2.1/b Read(Subject: Peter, Object: isa(book)).

2.2/a Peter reads a/the book Tom likes.
2.2/b Read(Subject: Peter, Object: isa(book):x |
Like(Subject: Tom, Object: x)).

From the set-based treatment of concepts follows that plural forms, when used for referencing objects in general, are represented as abstract concepts, i.e. multiple-element sets.

2.3/a Peter likes books.
2.3/b Like(Subject: Peter, Object: book).

3) By the representation of adverbs we should make a distinction between those that describe the circumstances of the action or state expressed by the predicate, and those that add extra conditions connected with the basic assertion. The latter is represented by the use of the *Happens* relation. The difference is clearly seen in the second example.

3.1/a Peter travels by train.

3.1/b Travel(Subject: Peter, Instrument: isa(train)).

3.2/a Peter often travels by train.

3.2/b Happens(Subject: Travel(Subject: Peter, Instrument: isa(train)), Time: Often).

3.2/c *Travel(Subject: Peter, Instrument: isa(train), Time: Often).*

The logical form in 3.2/c is incorrect, because from its truth does not follow that *Travel(Subject: Peter, Time: Often)* is true.

The following example is an illustration for the ambivalent nature of NL, where we cannot decide which predicate the adverb is linked to.

3.3/a I see you running today.

3.3/b See(Subject: I, Object: Run(Subject: You, Time: Today)).

3.3/c See(Subject: I, Object: Run(Subject: You), Time: Today).

4) Adjectives can be added to the assertion by the use of the *Property* relation.

4.1/a Peter reads a scientific book.

4.1/b Read(Subject: Peter, Object: isa(book):x | Property(Subject: x, Object: Scientific)).

5) For the treatment of numerical expressions we need to introduce numerical relations and numerical primitives, as well as the $some(a)$ function for creating a group of objects.

5.1/a Peter reads two books.

5.1/b Read(Subject: Peter, Object: some(isa(book)):x | Property(Subject: x, Object: Two)).

5.2/a Peter reads more books than magazines.

5.2/b Read(Subject: Peter, Object: (some(isa(book)):x, some(isa(magazine)):y) | More(Subject: x, Object: y)).

5.3/a Peter reads more books than Tom.

5.3/b (Read(Subject: Peter, Object: some(isa(book)):x), Read(Subject: Tom, Object: some(isa(book)):y) | More(Subject: x, Object: y)).

6) Existential and universal quantifiers are defined similarly by means of the $some(a)$ and $all(a)$ functions, respectively.

6.1/a There is a book on a/the table.

6.1/b Is(Subject: isa(book), Location: isa(table)).

6.2/a There are some books on a/the table.
6.2/b Is(Subject: some(isa(book)), Location: isa(table)).

6.3/a All books are on a/the table.
6.3/b Is(Subject: all(isa(book)), Location: isa(table)).

7) Logical operators can be applied to predicates or to arguments of predicates. When they refer to predicates we should note, that *and* means the presence of multiple predicates (they can be connected with the , operator), while *or* means the uncertainty of the observation (which is beyond the scope of our investigations).

7.1/a Peter reads and laughs.
7.1/b (Read(Subject: Peter), Laugh(Subject: Peter)).

7.2/a Peter reads not laughs. ≡ Peter reads.
7.2/b Peter does not read.
7.2/c NotRead(Subject: Peter).

In the latter example, case (a) demonstrates that we restrict our examinations to observations and the addition of extra information is not allowed. Case (b) states that Peter is not doing something without stating what he is doing. As a result, case (c) shows that an observation is uninterpretable without a specific predicate, thus *not* can only be allowed if included in the predicate.

The same applies when logical operators are related to arguments of predicates. Here *and* is represented by the grouping of the corresponding arguments, and *or* means uncertainty which is not covered by our investigations. Also, negation either expresses that an argument is not something without saying what it is, which is uninterpretable in our framework; or it states what the argument is, in which case the negation is an extra piece of information (e.g.: Peter reads not a book but a magazine. ≡ Peter reads a magazine.).

8) Temporal aspects can only be studied when several observations are compared on a historical basis. In this case the former observation(s) must have a tense preceding the latter observation(s). The observation at the end of the history demonstrates the actual (present) state of the system.

8.1/a Peter gives Tom a book. Tom reads the book.
8.1/b Give(Subject: Peter, Object: isa(book):x, Recipient: Tom) →
Read(Subject: Tom, Object: x).
8.1/c Tom reads the book that Peter gave him.

9) The examination of causes and results leads us back again to the *Happens* relation.

9.1/a Peter cannot sleep because Tom is dancing.

9.1/b Happens(Cause: Dance(Subject: Tom),
Result: NotSleep(Subject: Peter)).

From this analysis we can see that, in view of the criterion of composition preserving, the extended HOPL approximates NL better than the one without these extensions. Therefore, considering the assignment $m' : EHOPL/R \rightarrow NL/R$ (where $EHOPL/R$ denotes the set of extended HOPL statements constructed from semantic equivalence classes) we can state, that m' is a surjective mapping.

6. ECG: Graphical Representation of EHOPL

The present investigations concerning our project (see Figure 1) focus on modeling the agent's ability to assign semantic representations to its observations. For this purpose we have developed the Extended Conceptual Graph (ECG) model [23], a graphical conceptual modeling language that can be used to describe the semantics of an agent's internal knowledge model. In our model, the process of conceptualization occurs at two levels. At the primary level the direct and static mapping of the objects and relations within an observation takes place. At the extended or abstract level temporal and other complex relationship types are also managed. The main building blocks of the model are concepts, relationships, and containers which serve for structuring the model. The graphical representation of model elements is shown in Figure 2. The 'world' is built up of interconnected ECG model fragments representing separate observations, containing exactly one kernel predicate (denoted by *) and having 'true' truth value. The model is characterized by

- a predicate-centered schema language,
- the fine distinction between the different categories of concept and relationship types,
- the fixed set of elements with flexible semantic assignment, and
- the generality and reusability of basic model structures.

In the present article we examine the expressiveness of primary level ECG. Consequently, we will go through the linguistic phenomena identified in the previous section (except for temporal aspects, because they can only be studied at the extended level) and show that all of them can be represented in our model. In Equation 3.1 we defined a composition preserving transformation, and stated that two statements from different notation systems are said to

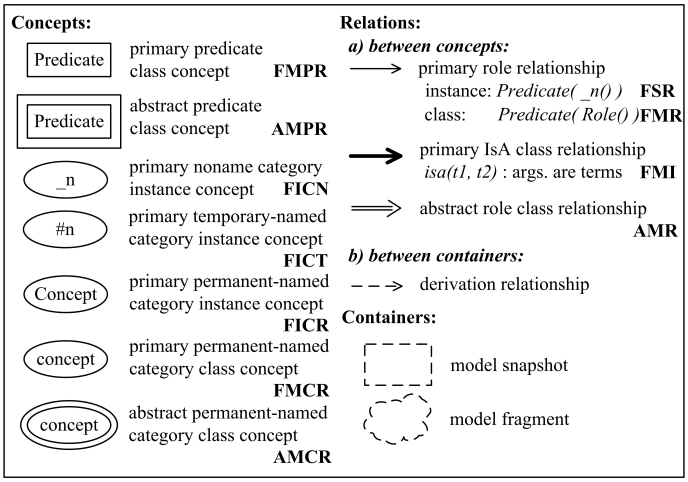


Figure 2. Components of the ECG model

be equivalent if there exists a composition preserving transformation between them by means of which the two formulas are equivalent in all interpretations. The following analysis specifies the composition preserving transformation of EHOPL formulas into graphical ECG structures.

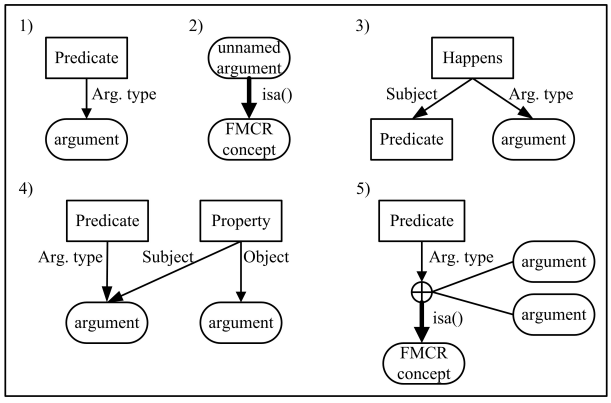


Figure 3. Basic ECG structures

Figure 3 shows the identified basic ECG structures. 1) illustrates a predicate with a typed argument, where types correspond to semantic roles. Arguments can be arbitrary ECG concepts, including predicate concepts as well. Objects are represented by different types of category concepts (see Figure 2).

Accordingly, we make a distinction between concepts referring to concrete objects (FICR), concepts referring to a collection of objects (FMCR), concepts referring to unreferenced unnamed objects (FICN), and concepts referring to referenced unnamed objects (FICT). The two latter serve for making a distinction between the use of indefinite and definite articles, respectively. 2) shows how an unnamed object is associated with a collection of named objects through the *isa*() relationship. Adverbs connected to the predicate are considered extra arguments of the predicate. On the other hand, 3) demonstrates how adverbs associated not only with the predicate itself but with the whole assertion are handled. Similarly, 4) displays the treatment of adjectives as arguments of the *Property* predicate. 5) shows how groups of objects can be composed. If an adjective indicating the cardinality of the group is also present then a *Property* predicate with an argument needs to be added to the construction. The handling of quantifiers and logical connectives originates in the previously discussed basic structures with the extension that also predicates can comprise a group. Causality can be traced back to 3) where the *Happens* predicate has a *Cause-type* and a *Result-type* predicate argument.

From these basic structures an ECG model, which is actually a semantic network, with arbitrary complexity can be built. For illustration, see Figure 4, which shows the ECG fragment for the observation "A black circle is in the white triangle".

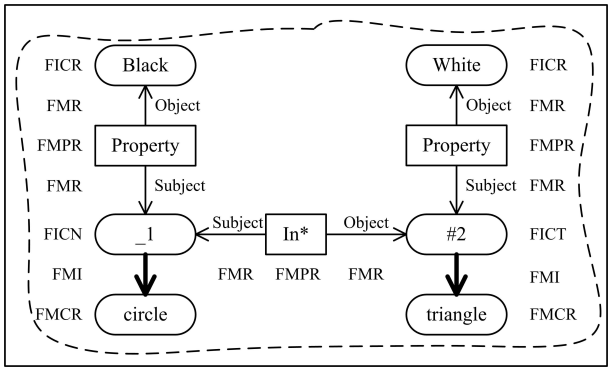


Figure 4. ECG fragment for an observation

This analysis proves that the assignment $e : EHOPL/R \rightarrow ECG$ (excluding temporal aspects) is a bijective function, therefore $e^{-1} : ECG \rightarrow EHOPL/R$ also exists. Thus we can state that the two formalisms are semantically equivalent, that is the same semantic content can be represented by both symbolisms. Therefore the ECG model can be viewed as the graphical counterpart of

the EHOPL language. As a consequence, the assignment $f' : ECG \to NL/R$ is a surjective mapping (just like m'). Note here that the set of ECG statements needs not be restricted to a set of semantic equivalence classes, because the ECG model is a semantic model constructed for representing the semantic content of a given situation.

7. Conclusion

The aim of the present article was the examination of primary level ECG's expressive power. In other words, we have been looking for an answer to the question: to what extent primary level ECG is able to model natural language semantics. Since the analysis was performed on a logical basis, this examination covered not less than the semantic comparison of natural language statements and logical formulas. We were interested in logically significant natural language expressions, and we have considered to what extent their semantics is captured by the logical behavior of their formal counterparts.

We can now draw the conclusion that the ECG model is able to grasp the semantic content of situations, and from the article we can see that every ECG statement can be rendered into an NL sentence. This assignment is unambiguous, that is every ECG statement can have only one corresponding NL formulation (with the assumption that semantically identical NL sentences are considered to be one). On the other hand, we can also state that every NL sentence can be approximated by an ECG model, if the pragmatic level of language is not taken into account. The ECG model is a recursive, compositional system: that is infinitely many statements can be constructed from the small finite set of model elements. Consequently, the more extended an ECG model is, the better it is able to approximate NL.

REFERENCES

[1] Ben-Yami, H.: *Logic & Natural Language*. Ashgate, 2004, ISBN 0-7546-3743-3.

[2] Blackburn, P. and Bos, J.: Computational semantics. *Theoria*, **18**, (2003), 27–45.

[3] Peregrin, J.: What does one need when she needs 'higher-order' logic? In *Filosofia*, LOGICA'96, Praha, 1997.

[4] Higginbotham, J. and May, R.: Questions, quantifiers and crossing. *Linguistic Review*, **1**, (1981), 41–79.

[5] Barwise, J. and Cooper, R.: Generalized quantifiers and natural language. *Linguistics and Philosophy*, **4**, (1981), 159–219.

[6] Jurafsky, D. and Martin, J. H.: *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*. Prentice-Hall, 2nd edn., 2008.

[7] Szeredi, P.: Az ontológiakezelés matematikai alapjai. Ontosz Klub, 2008.

[8] Keenan, E.: How much logic is built into natural language? In *Fifteenth Amsterdam Colloquium*, ILLC, University of Amsterdam, 2005, pp. 39–45.

[9] Bach, K.: *A Companion to Philosophical Logic*, chap. Language, logic, and form. Blackwell Publishers, 2002.

[10] Curry, H. B.: *Foundations of Mathematical Logic*. Dover Publications, Inc., New York, 1977, ISBN 0-486-63462-0.

[11] Shapiro, S.: *The Blackwell Guide to Philosophical Logic*, chap. Classical logic II: Higher order logic. Blackwell Publishers, 2001.

[12] Miller, D.: *Encyclopedia of Artificial Intelligence*, chap. Logic, Higher-order. 1991.

[13] Church, A.: A formulation of the Simple Theory of Types. *Journal of Symbolic Logic*, **5**, (1940), 56–68.

[14] Andrews, P.: *An Introduction to Mathematical Logic and Type Theory*. Academic Press, 1986.

[15] Henkin, L.: Completeness in the Theory of Types. *Journal of Symbolic Logic*, **15**, (1950), 81–91.

[16] Hinman, P. G.: *Fundamentals of Mathematical Logic*. A K Peters, 2005, ISBN 1-56881-262-0.

[17] Väänänen, J.: Second-order logic and foundations of mathematics. *The Bulletin of Symbolic Logic*, **7**(4), (2001), 504–520.

[18] Rossberg, M.: First-order logic, second-order logic, and completeness. In *First-Order Logic Revisited*, Logos Verlag Berlin, 2004, pp. 303–321.

[19] Gamut, L. T. F.: *Logic, Language, and Meaning. Volume 2. Intensional Logic and Logical Grammar*. The University of Chicago Press, 1991.

[20] van Benthem, J.: *The Logic of Time*. Kluwer Academic Publishers, 2nd edn., 1991.

[21] Jacquette, D. (ed.): *A Companion to Philosophical Logic*. Blackwell Publishers, 2002, ISBN 0-631-21671-5.

[22] Fillmore, C.: *Universals in Linguistic Theory*, chap. The Case for Case. New York: Holt, Rinehart and Winston, 1968.

[23] Baksa-Varga, E. and Kovács, L.: Knowledge base representation in a grammar induction system with Extended Conceptual Graph. *Scientific Bulletin of 'Politehnica' University of Timisoara, Romania*, **53**(67), (2008), 107–114.


Production Systems and Information Engineering
Volume 5 (2009), pp. 41-50.

ESTIMATION OF MISCLASSIFICATION ERROR USING BAYESIAN CLASSIFIERS

PÉTER BARABÁS
University of Miskolc, Hungary
Department of Information Technology
barabas@iit.uni-miskolc.hu

LÁSZLÓ KOVÁCS
University of Miskolc, Hungary
Department of Information Technology
kovacs@iit.uni-miskolc.hu

[Received January 2009 and accepted April 2009]

Abstract. Bayesian classifiers provide relatively good performance compared with other more complex algorithms. Misclassification ratio is very low for trained samples, but in the case of outliers the misclassification error may increase significantly. The usage of 'summation hack' method in Bayesian classification algorithm can reduce the misclassifications rate for untrained samples. The goal of this paper is to analyze the applicability of summation hack in Bayesian classifiers in general.

Keywords: Bayesian classifier, summation hack, polynomial distribution, misclassification error

1. Introduction

The Bayesian classification method is a generative statistical classifier. Studies comparing classification algorithms have found that the simple or Naive Bayesian classifier provides relatively good performance compared with other more complex algorithms. Accuracy of classification is a very important property of a classifier, a measure of which can be separated into two parts: a measure of accuracy in case of trained samples and a measure of accuracy in case of untrained samples. Naive Bayesian classification is generally very accurate in the first case since all testing samples are trained before and have no outliers; in the second case the efficiency is worse due to outliers. In [1], the role of outliers is examined in classification methods, the Naive Bayesian classification is reactive to outliers, and they can cause misclassification. Usage of summation hack can reduce the effect of outliers. The goal of our research is to analyze the generalization capability of Bayesian classification using summation hack. In the second part a short summary about Naive Bayesian classification is given. In the third part the concept of summation hack is introduced and examined. In the fourth part the classification methods are

analyzed considering the misclassification error. Finally, the test results and conclusions have been summarized in the last section.

It is assumed that the objects to be classified are described by n-dimensional pattern vectors $\boldsymbol{x} = (x_1,...,x_n) \in \boldsymbol{R}^n$. The dimensions correspond to the attributes of the objects. Every pattern vector is associated with a class label c, where the total number of classes is m. The class label c_i denotes that the object belongs to the i-th class. Thus, a classifier can be regarded as a function

$$g(\mathbf{x}) : R^n \rightarrow \{c_1,...,c_m\}. \tag{1.1}$$

The optimal classification function is aimed at minimizing the misclassification risk [2]. The risk value R depends on the probability of the different classes and on the misclassification cost of the classes:

$$R(g(\mathbf{x}) \mid \mathbf{x}) = \sum\nolimits_{c_j} b(g(\mathbf{x}) \rightarrow c_j) \cdot P(c_j \mid \mathbf{x}), \tag{1.2}$$

where $P(c_j \mid \boldsymbol{x})$ denotes the conditional probability of c_j for the pattern vector $\boldsymbol{x}$ and $b(c_i \rightarrow c_j)$ denotes the cost value of deciding in favor of c_i instead of the correct class c_j. The cost function b has usually the following simplified form:

$$b(c_i \rightarrow c_j) = \begin{cases} 0, if & c_i = c_j \\ 1, if & c_i \neq c_j \end{cases}. \tag{1.3}$$

Using this kind of function b, the misclassification error value can be given by

$$R(g(\mathbf{x}) \mid \mathbf{x}) = \sum\nolimits_{g(x) \neq c_j} P(c_j \mid \mathbf{x}). \tag{1.4}$$

The optimal classification function minimizes the value $R(g(\boldsymbol{x}) \mid \boldsymbol{x})$. As

$$\sum\nolimits_{c_j} P(c_j \mid \mathbf{x}) = 1, \tag{1.5}$$

thus if

$$P(g(\mathbf{x}) \mid \mathbf{x}) \rightarrow \max, \tag{1.6}$$

then the $R(g(\boldsymbol{x})|\boldsymbol{x})$ has a minimal value. The decision rule which minimizes the average risk is the Bayes' rule which assigns the $\boldsymbol{x}$ pattern vector to the class that has the greatest probability for $\boldsymbol{x}$[3].

2. Bayes classification

A Bayesian classifier is based on Bayes' theorem which relates to the conditional and marginal probabilities of two random events. Let A and B denote events. Conditional probability $P(A|B)$ is the probability of event A, given the occurrence of event B. Marginal probability is the unconditional probability $P(A)$ of event A, regardless of whether event B does or does not occur.

The simplified version of Bayesian theorem can be written for event A and B as follows:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}. \tag{2.1}$$

If $\bar{A}$ is the complementary event of A, called „not A". Let $A_1, A_2, A_3, \ldots$ be a partition of the event space. The general form of the theorem is given as:

$$P(A_i \mid B) = \frac{P(B \mid A_i)P(A_i)}{\sum_j P(B \mid A_j)P(A_j)}. \tag{2.2}$$

Let $C = \{c_k\}$ denote the set of classes. The observable properties of the objects are described by vector $\boldsymbol{x}$. An object with properties $\boldsymbol{x}$ has to be classified into the class for which the $P(c_k|\boldsymbol{x})$ probability is maximal. On the basis of Bayes' theorem:

$$P(c_k \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid c_k)P(c_k)}{P(\mathbf{x})}. \tag{2.3}$$

Since $P(\boldsymbol{x})$ is the same for all k we have to maximize only the expression $P(\boldsymbol{x}|c_k)P(c_k)$. The value $P(c_k)$ is given a priori or can be appreciated with relative frequencies from the samples. According to the assumption of Naive Bayes classification the attributes in a given class are conditionally independent of every other attribute. So the joint probability model can be expressed as

$$P(c_k, x_1, \ldots, x_n) = P(c_k)\prod_{i=1}^{n} P(x_i \mid c_k). \tag{2.4}$$

Using the above equation the probability of class c_k for an object featured by vector $\mathbf{x}$ is equal to

$$P(c_k \mid \mathbf{x}) = \frac{P(c_k)\prod_{i=1}^{n} P(x_i \mid c_k)}{P(\mathbf{x})}. \tag{2.5}$$

For the case where $P(c^*|\boldsymbol{x})$ is maximal the corresponding class label [5] is:

$$c^* = \operatorname{argmax}_{c \in C}\{P(c \mid \mathbf{x})\} = \operatorname{argmax}_{c \in C}\left\{P(c)\prod_{i=1}^{n} P(x_i \mid c)\right\}. \tag{2.6}$$

If a given class and feature never occur together in the training set, then the relative frequency will be zero. Thus, the total probability is also set to zero. One of the simplest solutions of this problem is to add 1 to all occurrences of the given attribute. In case of a large number of samples the distortion of probabilities is marginal and the information loss through the zero tag can be eliminated successfully. This technique is called Laplace estimation [4]. A more refined solution is to add p_k instead of 1 to the relative frequencies, where p_k is the relative frequency of k^{th} attribute value in the global teaching set, not only in the set belonging to class c_i.

3. Summation hack

Outliers in the classification can indicate faulty data which cause misclassification. The use of summation hack is an optional method to reduce the misclassification error. Summation hack is an ad-hoc replacement of a product by a sum in a probabilistic expression [1]. This hack is usually explained as a device to cope with outliers, with no formal derivation. This note shows that the hack does make sense probabilistically, and can be best thought of as replacing an outlier-sensitive likelihood with an outlier-tolerant one.

Let us define a vector $\boldsymbol{x}$ with components $x_1,x_2,\ldots,x_n$ and a class c. In Bayes classification where the vector values are conditionally independent:

$$P(\mathbf{x} \mid c) = \prod_{i=1}^{n} P(x_i \mid c). \tag{3.1}$$

In this case the probability is sensitive to outliers in individual dimensions so if any $P(x_i|c)$ value is equal to 0, the product will be zero. Using summation hack we get the following:

$$P(\mathbf{x} \mid c) \approx \sum_{i=1}^{n} P(x_i \mid c). \tag{3.2}$$

In this case the result will be zero if and only if all $p(x_i|c)$ values are equal to 0. Using (2.9) and (3.2) the computing of winner class is based upon the following formula:

$$c^* = \operatorname{argmax}_{c \in C} \{P(c \mid \mathbf{x})\} \approx \operatorname{argmax}_{c \in C} \left\{ P(c) \sum_{i=1}^{n} P(x_i \mid c) \right\}, \tag{3.3}$$

Applying summation hack the error of classification can be reduced. In every equation above the frequency probabilities are replaced with their approximated values, where

$$P(e) = \lim_{n \to \infty} \frac{n_e}{n_t}, \tag{3.4}$$

and n_t is the total number of trials and n_e is the number of trials where event e occurred. If the number of test events approaches infinity, the relative frequency value will converge to the probability value. In many classification tasks; a small number of samples is given [6], the number of tests is low, so a larger approximation error will arise in the calculations. We can write the probability as follows:

$$\frac{n_{x_i = v \mid c_j}}{n_{x_i}} = P(x_i = v \mid c_j) + \Delta_i, \tag{3.5}$$

where Δ_i means the error of approximation. The cumulated classification error in case of summation hack can be computed by the summation of the error elements. This error value differs from the classification error for the product of probabilities as it is calculated by the following form:

$$\prod_{i=1}^{n} (P(x_i = v \mid c_j) + \Delta_i) - \prod_{i=1}^{n} (P(x_i = v \mid c_j)). \tag{3.6}$$

4. Analysis of approximation error

The main cause of misclassification is the error of the approximated probability values shown in formula (3.4). To calculate the error value, the following model is applied. Let $\{c\}$ be the set of classes, and $\{a'\}$ the set of attributes where an attribute may be of vector value. A test case is described by a (a',c) pair where c denotes the class related to the a' attribute. The unknown probability that a_i belongs to c_j is denoted by p_{ij}. The relative frequency of the event that a_i belongs to c_j is denoted by g_{ij}. In the calculations p_{ij} are approximated by g_{ij}. The classification of the attribute can be regarded as a stochastic event, where $P(p_{ij}, g_{ij})$ denotes the probability that g_{ij} will be used in the calculations instead of g_{ij}.

Let $X(x_i)$ be a k-dimensional stochastic variable, where x_i denotes the number of attributes classified as c_i. X has a polynomial distribution:

$$P(x_1 = n_1, x_2 = n_2, ..., x_k = n_k) = \frac{N!}{n_1! n_2! ... n_k!} P_1^{n_1} P_2^{n_2} ... P_k^{n_k}, \tag{4.1}$$

where

$$N = \sum_{i=1}^{k} n_i, \sum_{i=1}^{k} P_i^{n_i} = 1. \tag{4.2}$$

A given $\boldsymbol{g}(n_1,n_2,...,n_r)$ frequency value has different P probabilities for the different $\boldsymbol{p}(p_1,p_2,...,p_r)$ probability tuples. The $\boldsymbol{p}(p_1,p_2,...,p_r)$ with maximal P value is assumed to be the real probability value tuple. As the maximum likelihood approximation of the probability is the frequency value, the relative frequencies are the best approximations of real probabilities:

$$P_i^{n_i} \approx \frac{n_i}{N}. \tag{4.3}$$

The probability of other $\boldsymbol{p}$ vectors can also be calculated with this formula. For the case $n = 2$ the resulting P distribution function is shown in Fig. 1.

In the next step, the approximation error of product $\prod P_i$ is calculated. It is clear that the larger the difference between $\boldsymbol{p}$ and $\boldsymbol{g}$, the higher the error value is. On the other hand, the lower the difference between p and g, the higher probability of this pair is. In the investigation, the average error value is calculated in the following way:

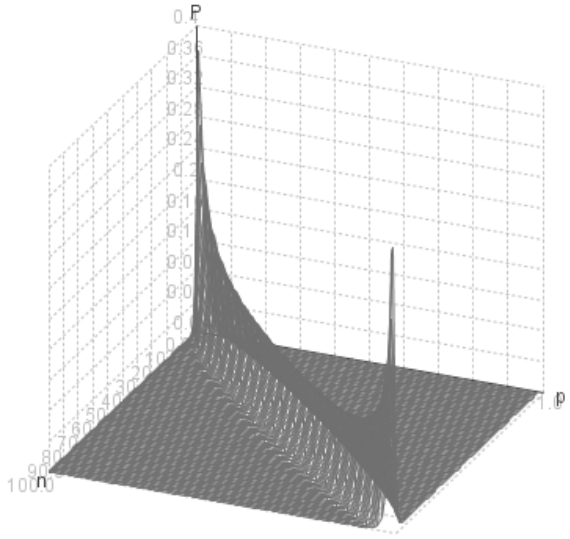
$$\varepsilon(\mathbf{g}) = \sum_{\mathbf{p}} P(\mathbf{p},\mathbf{g}) \cdot \varepsilon(\mathbf{p},\mathbf{g}), \tag{4.4}$$

where ε denotes the error value, where

$\varepsilon(\boldsymbol{p},\boldsymbol{g})$: the error value of matching $\boldsymbol{p}$ with $\boldsymbol{g}$,

$P(\boldsymbol{p},\boldsymbol{g})$: the probability of matching $\boldsymbol{p}$ with $\boldsymbol{g}$ and

$\varepsilon(\boldsymbol{g})$: the average error related to frequency vector $\boldsymbol{g}$.



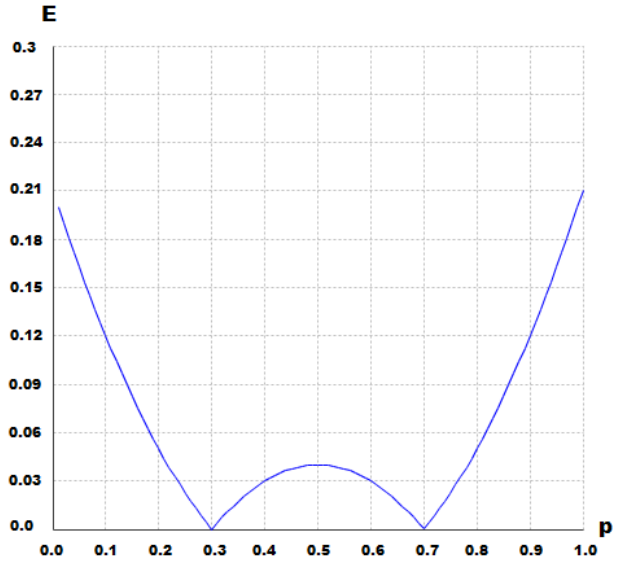
,

Figure 1. Probability function for the case n=2

In the test case, the error formula for p_i and the mean value of error can be computed as follows:

$$\varepsilon(\mathbf{p},\mathbf{g}) = p_1(1-p_1) - \frac{n_1}{N}(1-\frac{n_1}{N}), \tag{4.5}$$

Fig. 2 shows the error function for the test binomial case. The number of attempts is 100 where the number of attributes belonging to class c_1 is 30. In the Figure, can be seen the minimum error is in case of p=0.3. Since the function is symmetric, another minimum point can be found at p=0.7.



,

Figure 2. Error function for the case p=0.3

In Naive Bayesian classifier the accuracy depends strongly on the number of attempts. The larger the test pool, the better the accuracy is. In Fig. 3 the mean value error function can be seen for different N values. The results show that for a small number of N values the use of summation hack can improve the accuracy but for a larger test pool the Naive Bayesian classifier is the dominant one.

5. Test results

In first tests [7] the reference points were generated with uniform distribution in space. The winner was the Naive Bayesian classifier in teaching and testing phase equally. The teaching accuracy had values from 80% to 100% depending on environment parameters. Using summation hack this accuracy decreased by about 10%. The testing accuracy is far lower, it is between 40 and 70 percent in case of Naive Bayesian classifier and lower using summation hack. The relatively large range of result values can be explained by the overtraining of the model which can be controlled by the correct choice of environment parameters.

In later tests the reference points were generated sparsely, so the space has a small region with a relatively large number of reference points and outside this region there are only a few reference points.

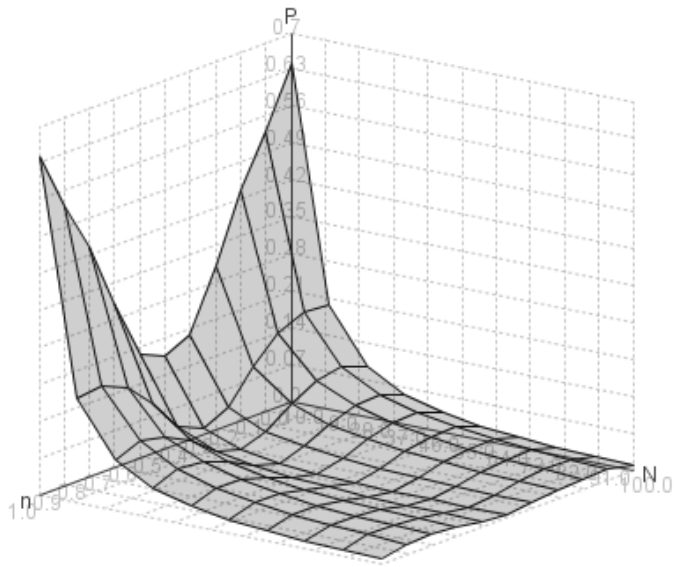


Figure 3. Mean value error function for different (n_1, N) values

In the case of this distribution, the accuracy of classifiers has changed. The teaching accuracy of Naive Bayesian classifier remained highly similar to other cases and the use of summation hack brought up the accuracy to the Naive Bayesian. In the testing phase the experience shows that in some cases the summation hack solution can improve the efficiency of classification and in many cases it exceeds the Naive Bayesian. It confirms the assumptions that the usage of summation hack in Bayesian classification can increase accuracy when the samples contain a great number of untrained attribute values.

The accuracy of classification depends on many parameters of the environment. One of the most important factors is the maximum attribute value parameter. Fig. 4 shows the accuracy functions for the following maximum attribute parameter values: 20 (NB20,SH20), 100 (NB100,SH100) and 500 (NB500,SH500). The notation NB is for Naive Bayesian algorithm and SH for the modified Bayesian algorithm. The accuracy of both algorithms has increased with increasing the size of the training set.

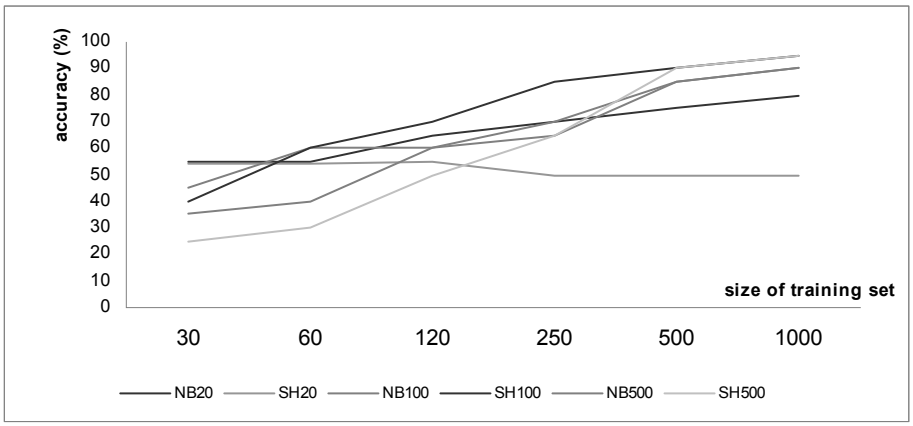


Figure 4. Relative accuracy of algorithms according to number of teaching samples

6. Conclusions

Summation hack is an alternative for the Naive Bayesian classifier with larger probability approximation errors. Taking a decision tree as a reference classifier, we have compared the Naive Bayesian classifier with the Bayesian classifier using summation hack. The test results show that both methods can yield the same accuracy as the decision tree method has in the case of large training sets.

REFERENCES

[1] THOMAS P. MINKA: *The 'summation hack' as an outlier model*, technical note, August 22, 2003

[2] HOLSTROM L, KOISTIEN P, LAAKSONEN J., OJA E: *Neural and Statistical Classifiers - Taxonomy and Two Case Studies*, IEEE Trans. On Neural Networks, Vol 8, No 1, 1997.

[3] KOVÁCS L., TERSTYÁNSZKI G.: *Improved Classification Algorithm for the Counter Propagation Network*, Proceedings of IJCNN 2000, Como, Italy.

[4] JOAQUIM P. MARQUES DE SÁ: *Applied Statistics Using SPSS*, Statistica, Matlab and R, Springer, 2007, pp. 223-268

[5] FUCHUN PENG, DALE SHUURMANS, SHAOJUN WANG: *Augmenting Naive Bayes Classifiers with Statistical Language Models*, Information Retrieval, 7, Kluwer Academic Publishers, 2004, Netherlands, pp. 314-345

[6] ROBERT P.W. DUNN: *Small sample size generalization*, 9th Scandinavian Conference of Image Analysis, June 6-9, 1995, Uppsala, Sweden

[7] BARABÁS P., KOVÁCS L.: *Usability of summation hack in Bayes Classification*, 9th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics, November 6-8, 2008, Budapest, Hungary


Production Systems and Information Engineering
Volume 5 (2009), pp. 51-66.

DEVELOPING MODELS BASED ON REAL ENVIRONMENTS

TAMÁS BÁKAI
University of Miskolc, Hungary
Regional University Knowledge Centre
retbakai@gold.uni-miskolc.hu

[Received January 2009 and accepted April 2009]

Abstract. All of the applications of information engineering are based on a correct model of the environment the applications represent. Unlike artificial environments, real environments cannot be modelled correctly. Firstly because the behaviour of the real environments also depends on the behaviour of other environments also. These relations cannot be revealed at the time the model is constructed. Therefore the boundaries of a real environment cannot be defined correctly. The only thing the model-designer can do is to define a model describing the behaviour of the environment to be modelled with no inconsistencies at the time the model is constructed. Therefore each model based on real environments needs to be redesigned continuously as the time passes to provide their consistency. It is only feasible using adaptive knowledge-intensive modelling tools. This paper shows a new concept for modelling real environment.

Keywords: system identification, behaviour description, knowledge-intensive modelling tool

1. Problem declaration

Nowadays more and more fields of our life are supported by applications of information engineering. These applications have to work not only in artificial environments, but in real environments as well. The main difference between artificial and real environments based on the boundaries of environment needs to be modelled. Both the boundaries and the behaviour of an artificial environment are defined by the model designer only. The object structure and the set of rules describing the behaviour of these environments can be defined at the time the model is constructed. Some models designed for working in artificial environments can learn rules and can extend their knowledge-base with those rules, but cannot invalidate a rule defined by the model designer. It is because the behaviour of artificial environments does not change. In artificial environments the set of rules is finite and therefore the behaviour of these environments can be transformed into a deterministic model easily. Unfortunately the applications based on artificial environments cannot work in real environments correctly. It is because the object

structure and the set of rules representing the behaviour of a real environment cannot be determined at the time the model is constructed. The behaviour of a real environment depends on the behaviour of other real environments also and these dependencies cannot be revealed correctly until they appear. Therefore the set of rules of a real environment cannot be described correctly. The only thing the model designer can do is to define a model which represents the correct behaviour of the real environment needed to be modelled at the time the model is constructed. Besides the indefinable boundaries the modelling of real environments also suffers from the complexity of their state space. For revealing the problem based on the complexity of real environments let us consider an environment with *n* pieces of state descriptor variables. If each of these variables has only two values, then the state space of this environment has 2^n independent states. The rules representing the behaviour of the real environment need to be revealed in such a large state-space. In the worst case it means 2^n pieces of rules for describing the behaviour of one of the state descriptor variables of the environment. For describing the behaviour of the whole environment $n*2^n$ pieces of rules have to be modelled. Table 1 shows an example of an environment with three state descriptor variables, each with two values.

Table 1. Complexity of the behaviour description of an environment containing three state descriptor variables (A, B, C), each with two values

conditions			conclusion
A	B	C	A
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

a,

conditions			conclusion
A	B	C	B
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

b,

conditions			conclusion
A	B	C	C
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

c,

Usually the state descriptor variables have more than two values. In the general case the complexity of behaviour description can be calculated by multipling the number of the values of each state descriptor variable of that environment and multipling the result by the number of the state descriptor variables of the environment. For example if the environment consists of three state descriptor variables with three, two and four values, then the complexity of the behaviour description of this environment is (3*2*4)*3 = 72 pieces of rules in the worst case. For example a medium sized environment has some hundreds of state description variables. The behaviour revealing cannot be modelled with the common

algorithms in so large a state-space. Artificial environments with large state spaces (like game of chess) can be grasped by common algorithms because both the object structure and the set of rules of these environments are determined by the model designer only. Therefore the large state space of these environments cannot be revealed to find the necessary rules.

If the application is required to work in a real environment then the object structure and the set of rules of the environment need to be revealed. Unfortunately, these features cannot be determined correctly because of the complexity of the real environments and their dependencies on other environments. Therefore the real environment cannot be transformed into a deterministic model correctly. At this point I need to mention that theoretically there is a deterministic real environment (see [2]). Its state space consists of all of the elementary particles in the universe. Probably information engineering will never handle this complex environment so the real environments can be held as these would be stochastic. It means that the state of a real environment at time t cannot be determined by the state of that environment at time $t-1$ and the set of rules of that environment only. The most the designer can do in order to eliminate the inconsistencies is to widen the boundaries of the real environment needed to be modelled until the environment behaves if it were deterministic. By this the behaviour of that environment at time t can be described correctly. To provide the consistency of the model as the time passes the model is needed to be redefined continuously. It is only possible using knowledge-intensive learning concept. This paper shows a new concept for solving this problem efficiently.

2. Modelling the object structure of the environment

The object structure of an environment can be modelled in several ways. This paper uses the most common and popular object-oriented designing concept [3]. According to this concept the object-structure of the environment needed to be modelled is described by object-attribute-value triple. Each object in this model represents a set of properties belonging to an elementary unit of the environment. For example in the game of chess it can be a figure on the chess-board. Each property belongs to an object named the attribute of that object. Each attribute represents an observable property in the environment. An attribute of the figure on the chess-board is its position. Each attribute has some values. While the objects and the attributes are virtual elements in the object-structure (whose only rule is to categorize the observable elements) a value can be observed by a sensor. For example a value can be a colour value (black, white, red, ...), a temperature value (0°C, 20°C, ...) or in case of the position attribute of the figures on the chess-board it can be a coordinate point. Each attribute has one marked value among it values at any time. This value represents the state of the attribute it belongs to at that time. The set of the active values of each attribute in the environment at a given time represents the state of the environment at that time. As the time passes a marked

value may lose its marked state and an unmarked value may become marked, but at any time each attribute has one and only one marked value. This process takes place at discrete time periods determined by the sampling frequency of the modelling tool only. Each sample is a state of the environment. Two consecutive samples give a change of states of the environment.

Some of the objects of an environment can do activities. For example a figure on the chess-board can move. An object which can do an activity is called an agent. The activities are similar to attributes from the point of view of the description of the states of the environment. The state of the environment at time t is given not only by the state of the attributes of the environment at time t but by the state of the activities at that time also. The attributes and the activities represent the dimensions of the environment. The values of the dimensions represent the points that can be reached along these dimensions. Each activity has two values. One for representing the fact the activity is done and one for representing the fact the activity is not done. The main difference between the attributes and the activities is that the values of the activities cannot be observed. Each agent knows the state of its activities but cannot observe the state of the activities of another agent. Therefore at first approach each agent considers the objects of the environment as these would be agents with no activities. The state of the activities can be deduced from their effect on the state of the attributes of the environment.

The designer tool in this paper is modelled as an intelligent agent trying to reveal the behaviour of the real environment needed to be modelled. The designer tool may have activities to interact with the environment it reveals. Using these activities the agent can lead the environment into states that have not observed yet. This possibility allows the agent to reveal new parts of the environment. The designer tool is modelled with one agent (containing all of the activities the designer tool can do) and a set of objects (representing the sensors with which the designer tool can observe the environment). Unlike this, the environment is modelled with a set of agents without objects. It is because any of the objects of the real environment may have activities.

Figure 1 shows the object structure of an environment with two objects. The first one is an agent which represents the inner properties of the developing tool and the second one is an object in the environment. The agent has a sensor called $A_1.At_1$ for describing the state of the agent, and two activities called $A_1.Ac_1$ and $A_1.Ac_2$ for describing the activities the agent can do. The developing-tool has a sensor called $O_1.At_1$ for describing the state of the environment. Both of the attributes in the object structure have two values.

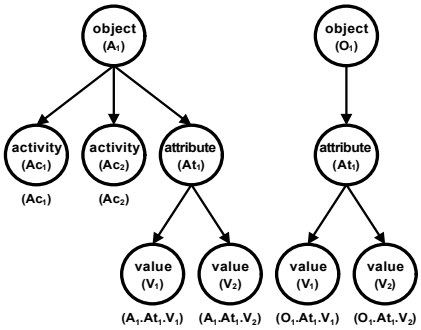


Figure 1. Object structure of an environment with two objects

In general the attributes of a real environment have many values. For example if one of the sensors of the designer tool is a thermometer then it can be modelled with an attribute. If this thermometer can measure the temperature in the range [-20...40°C] with the precision of 0.1°C, then that attribute has 600 values. The great amount of values enlarges the state space of the environment significantly, therefore makes the designer tool unworkable. To handle this problem only the values that have become active values are modelled. It decreases the complexity of the object structure considerably but does not make the model incorrect. The values which never become active are not existing values of the attribute.

The designer tool records the state of the environment to be modelled at each discrete time period according to a sampling rate. The resulting sequence is the history of the environment. The behaviour of a real environment can be revealed correctly using a history with infinite pieces of records. In practice it is not possible, but in general the more pieces of records there are in the history, the more correct the model. On the other hand, increasing the sampling frequency increases the correctness of the model as well. Each of these possibilities increases the size of the history significantly. One of the greatest problems the designer must eliminate in modelling a real environment is the great amount of information needed to be handled. The continuously growing number of records in the history requires storage and processing units with continuously growing capacity. For modelling the values that have become active at least once can decrease the amount of information needed to be stored in the history but in case of real environments it is not enough. The solution of this problem is oblivion. Each adaptive knowledge-intensive system needs oblivion to follow the changes in the behaviour of the real environment it models. For example imagine a railroad schedule. After it has changed, the old version has to be forgotten in order for the designer-tool to work properly. The history with oblivion in this paper is modelled as follows. Each value records a timestamp each time it becomes the active-value of the attribute it belongs to. This time-stamp is called satisfaction. Each satisfaction represents the time it occurred, the duration it is memorable in the short-term memory and the duration it is memorable in the history. The duration a satisfaction is memorable in

the history is greater than or equal to the duration that satisfaction is memorable in the short-term memory. A satisfaction is removed from the history if the duration it is memorable in the history expires. If none of the satisfactions of a value is memorable in the short-term memory, it removes all of the satisfactions of that value from the history. This rule models the oblivion process of intelligent beings. According to this if an event occurs frequently, then the occurrences of this event in the past are memorable for a long time, but if an event does not occur for a while, then all of the occurrences of this event become forgotten. The developing tool can handle different durations for each satisfaction. Therefore the oblivion of intelligent beings can be modelled with higher precision. For determining these durations is a developing possibility for the future. At this point constant durations are determined for all of the satisfactions of the values. A value which loses all of its satisfactions will be removed from the object structure. Similarly an attribute which loses all of its values will be removed from the object structure. An activity does not lose any of its values but if it is not done for a while (defined by the short-term memory) then that activity is removed from the object structure. An object or an agent which loses all of its dimensions will be removed from the object structure. Modelling of the values that have became active at least once only and oblivion extend the abilities of the designer tool to model real environments.

3. Revealing the behaviour of the environment

3.1. The concept of the revealing behaviour

In case of deterministic environments the state of the environment at time t can be calculated by the state of that environment at time $t-1$. Therefore the set of rules that describes the behaviour of a deterministic environment can be revealed correctly. For example a wrist-watch is a deterministic environment because its state at time t is determined unambiguously by its state at time $t-1$. No matter how complex a deterministic environment is there is a possibility (at least theoretically) to reveal the set of rules that represents the behaviour of that environment correctly. Unlike this, the state of a stochastic environment at time t cannot be calculated by its state at time $t-1$ only. It is because stochastic environments are partly observable only. It means that there is always a set of state descriptor variables which produce an effect on the environment but are not modelled. For example the traffic lights are a stochastic environment from the point of view of the traffic. If its state at time t is red then its state at time $t+1$ can be red or green equally. Because the traffic lights are an artificial environment (that is its set of rules is defined only by its designer) then its stochastic behaviour can be transformed into deterministic unambiguously. For example a counter can be put beside the traffic-lights which counts downwards until zero. The zero induces the switch of colours. Therefore if the boundaries of the traffic lights environment are extended by this counter (as a new state descriptor variable), then the state of the traffic lights at time t can be calculated by its state at time $t-1$ correctly.

Unfortunately real environments remain always stochastic. The most the designer can do is to determine the boundaries of the environment needed to be modelled with which the behaviour of the real environment can be described as if it were deterministic up to the actual date. Unfortunately the rules that describe the behaviour of a real environment up to time t correctly may become incorrect at time $t+1$. In real environments no one can guarantee that the rules revealed are correct. The most that can be guaranteed is that the rules revealed describe the behaviour of the environment up to the actual date correctly. Intelligent beings trying to reveal the behaviour of the environment around them work on this basis. We cannot be sure that our equations that describe the environment around us are correct. Each time we encounter a contradiction modify the equations to describe the environment correctly up to the actual date.

Therefore if according to experience up to time t each time the sun shines a rainbow can be seen then the following rule that describes the visibility of the rainbow is correct.

IF the sun shines **THEN** a rainbow can be seen. (3.1)

According to our experience we know that this rule is not correct. It is because we have a large amount of experience in our history. Our experience makes us to find a more correct formula but no one can guarantee that our formula is correct and that then cannot come new experience bringing contradiction. If new experience contains a sun shining and a rainbow cannot be found in the sky then the previous rule has to be reconstructed. For this the agent tries to extend the condition part of the rule with one state descriptor variable it can observe. For example if the agent has a thermometer then it tries to use the temperature to eliminate the contradiction of the rule. If the history contains that each time the sun shone and the temperature was lower than 20°C and the rainbow was visible then the following rule can be constructed.

IF the sun shines **AND** the temperature is lower than 20°C **THEN** a rainbow can be seen. (3.2)

Taking this with each state descriptor variable the agent can observe, the agent creates a set of new rules which describe the visibility of the rainbow correctly. If the agent has a sensor for observing the rain then maybe one of these rules is the following.

IF the sun shines **AND** it rains **THEN** a rainbow can be seen. (3.3)

If the agent does not have a sensor for observing the rain then it is possible that the agent cannot find another sensor among the sensors the condition part of the inconsistent rule is extended with that rule becoming correct. In this case the agent tries to find two sensors to extend the condition part of the incorrect rule. If it is

possible then the problem is solved up to the actual date, but if it is not possible then a state of more and more sensors is needed to be taken in the condition part of the rule. Following this concept some problems arise:

- the states of all of the sensors the agent has are in the condition part of the rule but the contradiction cannot be eliminated
- the contradiction of the incorrect rule is eliminated successfully but there are numerous state descriptor variables in the condition part of the rule.

The solution of these problems implemented in the designer tool will be shown in Section 4 of the paper.

3.2. Using initial knowledge in learning systems

Implementing initial knowledge into learning-applications is popular practice today. It makes the application more efficient in revealing the behaviour of the environment because it protects it from learning numerous misleading rules. In spite of this the developing tool described in the paper avoids applying initial knowledge. It is because the application designer is an agent too whose knowledge about the real environment cannot be correct. The initial knowledge describes some of the rules of the real environment and these rules describe the behaviour of the environment at the time the application is constructed only. Therefore it helps the application in the near future only. As time passes the initial rules become more and more inconsistent and these inconsistencies make the application unable to work.

3.3. Methods implemented methods for revealing the behaviour

This section shows the method implemented in the developing tool for revealing the behaviour of real environments. For this let us consider the object structure shown in Figure 1. This object structure contains two values for each dimension. It is the object structure of one of the simplest models that can be constructed and it is to demonstrate the learning process of the developing tool in the simplest way only. The concept shown is applicable for handling attributes with more than two values also. As Figure 1 shows, this environment has four independent states. These are $[A_1.At_1.V_1, O_1.At_1.V_1]$, $[A_1.At_1.V_1, O_1.At_1.V_2]$, $[A_1.At_1.V_2, O_1.At_1.V_1]$, $[A_1.At_1.V_2, O_1.At_1.V_2]$. The agent in this environment can do four activities. These are [nothing], $[Ac_1]$, $[Ac_2]$, [Ac_1 and Ac_2]. Let the behaviour of the environment the developing-tool has to reveal be as follows:

- Ac_1 changes the state of the attribute $[A_1.At_1]$
- Ac_2 changes the state of the attribute $[O_1.At_1]$

Let the state of the environment at the beginning of the investigation is [$A_1.At_1.V_1$, $O_1.At_1.V_1$]. Figure 2 shows the time sequence of the changes of states in the environment the developing tool will observe.

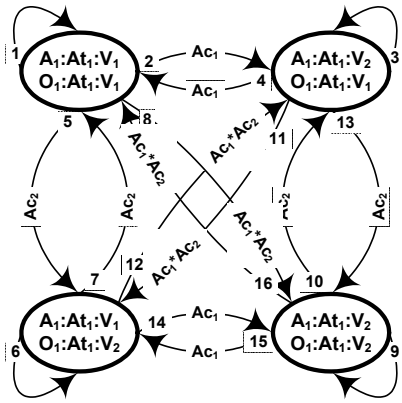


Figure 2. Time sequence of the changes of states in the environment the developing tool will observe

In Figure 2 the four ellipses represent the states of the environment and the arrows represent the activities the agent does. At the middle of each arrow the name of the activity - that the arrow represents - can be seen. An arrow with no activity name represents the null activity, that is no activity is done during that change of states. The number at the beginning of each arrow represents the serial number of the change of states according to the time sequence. Following the sequence of these changes of states the values in the object structure of the environment are satisfied as shown in Figure 3.

Satisfactions of "A1.At1"

Back

	V1			V2	
Date	STM	Memorability	Date	STM	Memorability
0	40 (24)	100 (84)	2	42 (26)	102 (86)
1	41 (25)	101 (85)	3	43 (27)	103 (87)
4	44 (28)	104 (88)	8	48 (32)	108 (92)
5	45 (29)	105 (89)	9	49 (33)	109 (93)
6	46 (30)	106 (90)	10	50 (34)	110 (94)
7	47 (31)	107 (91)	12	52 (36)	112 (96)
11	51 (35)	111 (95)	13	53 (37)	113 (97)
14	54 (38)	114 (98)	15	55 (39)	115 (99)
16	56 (40)	116 (100)			

Satisfactions of "O1.At1"

Back

	V1			V2	
Date	STM	Memorability	Date	STM	Memorability
0	40 (24)	100 (84)	5	45 (29)	105 (89)
1	41 (25)	101 (85)	6	46 (30)	106 (90)
2	42 (26)	102 (86)	8	48 (32)	108 (92)
3	43 (27)	103 (87)	9	49 (33)	109 (93)
4	44 (28)	104 (88)	11	51 (35)	111 (95)
7	47 (31)	107 (91)	13	53 (37)	113 (97)
10	50 (34)	110 (94)	14	54 (38)	114 (98)
12	52 (36)	112 (96)	15	55 (39)	115 (99)
16	56 (40)	116 (100)			

Satisfactions of "A1.Ac1"

Back

	Active			Inactive	
Date	STM	Memorability	Date	STM	Memorability
1	41 (24)	101 (84)	2	42 (25)	102 (85)
3	43 (26)	103 (86)	4	44 (27)	104 (87)
7	47 (30)	107 (90)	5	45 (28)	105 (88)
10	50 (33)	110 (93)	6	46 (29)	106 (89)
11	51 (34)	111 (94)	8	48 (31)	108 (91)
13	53 (36)	113 (96)	9	49 (32)	109 (92)
14	54 (37)	114 (97)	12	52 (35)	112 (95)
15	55 (38)	115 (98)	16	56 (39)	116 (99)

Satisfactions of "A1.Ac2"

Back

	Active			Inactive	
Date	STM	Memorability	Date	STM	Memorability
4	44 (27)	104 (87)	5	45 (28)	105 (88)
6	46 (29)	106 (89)	8	48 (31)	108 (91)
7	47 (30)	107 (90)	13	53 (36)	113 (96)
9	49 (32)	109 (92)	14	54 (37)	114 (97)
10	50 (33)	110 (93)	16	56 (39)	116 (99)
11	51 (34)	111 (94)			
12	52 (35)	112 (95)			
15	55 (38)	115 (98)			

Figure 3. The satisfactions of the values in the object-structure of the environment

The header of the table in each screenshot contains the name of the values of the dimension the screenshot represent. Each column belongs to a value representing the satisfactions of that value. The columns called *Date* contain the date in the environment the satisfaction appeared. The number without brackets in the columns called *STM* represents the date the satisfaction remains memorable in the short-term memory. The number within brackets in these columns represents the time that remains until this date. The number without brackets in the columns named *Memorability* represents the date the satisfaction remains memorable. The number within brackets in these columns represents the time that remains until this date.

This presentation deals with the topic of revealing the rules that represent the behaviour of the attributes in the object structure of the environment only. At the first approach the environment is regarded as if it were deterministic. That is the state of the environment at time t can be calculated by the state of the environment at time $t-1$ and the activities done at time $t-1$. If at time t this supposition fails that makes the environment deterministic until time t by the methods described in Section 4. Revealing the rules that represent the behaviour of the activities of the agents is a development possibility. At each state of the environment each agent calculates the set of next states thet can be reached by its activities. Supposing that each agent chooses the state best for itself the set of activities can be determined.

According to the concept of the revealing behaviour this paper follows, the rules that describe the behaviour of the attributes of the environment are determined by the hypothesis space of these attributes. A hypothesis space is a set of consequences the agent observes during its experiences [1]. In the hypothesis-space the agent can classify the different consequences using the conditions that caused those consequences in order to form distinct subsets of the same consequences. The consequences in each subset represent the same consequence which was observed at different time. The condition of each subset can be obtained by the logical AND relation needed to form that subset. The rule describing a consequence can be formed by the logical OR relations of the condition of each subset representing that consequence. Figure 4 shows an example of a hypothesis space.

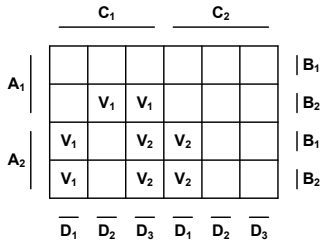


Figure 4. Example of a hypothesis space

Each place in the hypothesis space shown in Figure 4 represents an experience. The letters in the places represent the consequences of those experiences. The consequence in this case is the value that became the active value of the attribute whose hypothesis space is represented in Figure 4. The letters next to the table represent the dimensions (attributes or activities) of the environment. The index of the dimensions denotes the serial number of the value of that dimension. Each of these values represent the condition (observed by sensors) at the time the experience was recorded. In this example the letters *A*, *B* and *C* represent the dimensions of the environment with two values. Letter *D* represent a dimension of the environment with three values. Each dimension divides the hypothesis space into so many pieces as the number of its values. The number of places represented by the logical AND relation of values belonging to different dimensions can be calculated by dividing the number of places in the hypothesis space by the number that can be calculated by multiplying the number of values of the dimensions those values belong to. For example the number of places represented by A_1 **AND** D_1 in case of the hypothesis space shown in Figure 4 is 24/(2*3) = 4 pieces of places. The condition of the subsets containing the same consequences is as follows:

$$\begin{aligned} &C_1 \text{ AND } D_1 \Rightarrow V_1 \\ &A_1 \Rightarrow V_1 \\ &A_2 \text{ AND } D_3 \Rightarrow V_2 \\ &C_2 \Rightarrow V_2 \end{aligned} \tag{3.4}$$

As this example shows the places that are not experienced can be taken into any subset. It follows the concept intelligent beings reveal the behaviour of the environment. If each time the sun shone a rainbow was visible then the state of the sensor indicating the state of the sun is enough to describe the visibility of the rainbow. Other sensors are not needed for this until a consequent experience appears.

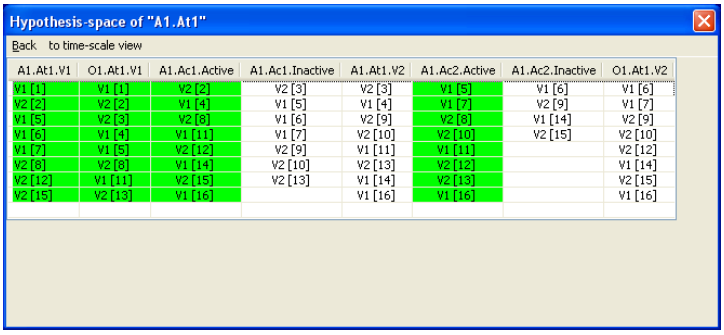
The rule representing the behaviour of a value of an attribute can be determined by the logical OR relation of the condition of the subsets containing the satisfactions of that value. For example the rules as describing the behaviour of the attribute whose hypothesis-space is shown in Figure 4 are the follows.

IF (C_1 **AND** D_1) **OR** A_1 **THEN** V_1

IF (A_2 **AND** D_3) **OR** C_2 **THEN** V_2 (3.5)

The rules generated by this concept can be simplified using existing algorithms therefore this paper does not detail this issue.

Figure 4 shows the simplest representation of hypothesis spaces for understanding its role only. However this representation wastes memory significantly. It is because it reserves places for all of the experiences with different conditions. In general cases the number of the experiences appearing in an environment as much smaller than the number of the experiences with different conditions in that environment. Therefore the representation of the hypothesis spaces shown in Figure 4 can be used for environments with few state descriptor variables only. For handling environments with as many state descriptor variables as possible a new representation of the hypothesis spaces had to be developed. This new representation reserves places for the experiences that have been appeared at least once only. This new representation eliminates the waste of memory and therefore maximizes the number of the state descriptor variables. In this new representation the consequences of the experiences are organized in lists according to its conditions. The hypothesis spaces of the experiences shown in Figure 3 can be seen in Figure 5.



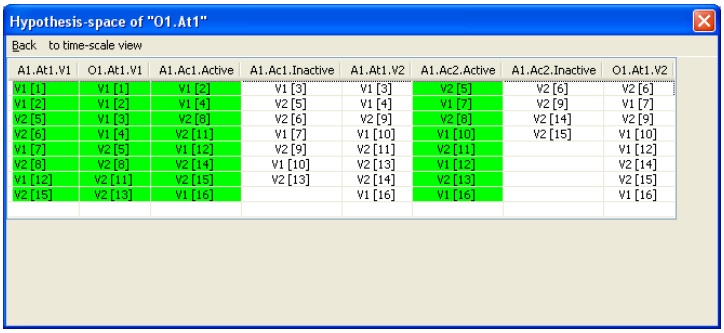


Figure 5. Hypothesis-spaces of the attributes of the environment shown in Figure 3

This representation uses no more memory space than needed for describing the hypothesis spaces. The header of the tables contains the conditions of the experiences. In each column the satisfactions of the values of the attribute whose hypothesis space is represented by the table are listed under the condition which was satisfied at the previous state of the environment where the satisfactions appeared. The number behind each consequence in square brackets represents the date that the satisfaction appeared. The green background of a column indicates that the condition associated with that column is satisfied at the actual state of the environment. If a column contains the satisfactions of the same value then the condition of that column represents the cause of the satisfaction of that value. For example according to the hypothesis spaces shown in Figure 5 each time the activity called $A_1.Ac_2$ was not done the value called $O_1.At_1.V_2$ was the active value of its attribute at the next time. Therefore the following rule can be constructed.

IF $A_1.Ac_2.\text{Inactive}$ **THEN** $O_1.At_1.V_1$ (3.6)

If a column in this table contains the satisfactions of different values of the attribute whose hypothesis space that table represents then two or more columns had to be combined in order to obtain a new column with the satisfactions of the same value.

For example in Figure 5 the column with condition $A_1.At_1.V_1$ contains the satisfactions of the values $A_1.At_1.V_1$ and $A_1.At_1.V_2$. If this column is combined with the one whose condition is $A_1.Ac_1.Active$, then the resulting column contains the satisfactions of the value $A_1.At_1.V_2$ only. Therefore the following rule can be constructed.

IF $A_1.At_1.V_1$ **AND** $A_1.Ac_1.\text{Active}$ **THEN** $A_1.At_1.V_2$ (3.7)

If the condition of a resulting column contains at least two values belonging to the same dimension, then that column does not contain a satisfaction. If the resulting column contains the satisfactions of the same value, then this column does not need to be combined with other columns. The rule represents the behaviour of a value of an attribute hat be formed by taking the condition of the resulting rules leading to the consequence of that value into logical OR relation with each other. Using this concept the logical rules representing the behaviour of the environment shown in Figure 3 will be as follows.

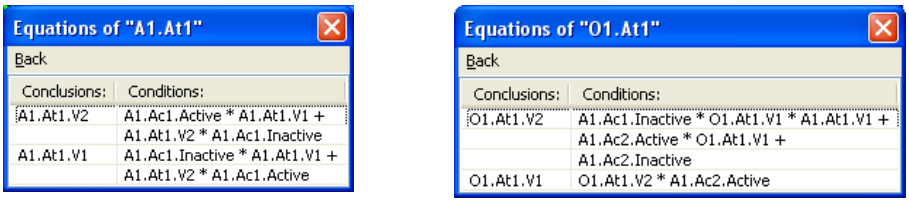


Figure 6. Rules representing the behaviour of the environment

The rules shown in Figure 6 are not equal to the ones expected according to the sequence of the changes of states shown in Figure 2. It is because the agent handles the values observed at least once. The value $A_1.At_1.V_2$ can be observed at time 2 first and the value $O_1.At_1.V_2$ can be observed at time 5 first. Therefore the behaviour of these values cannot be observed until that time. For modelling the behaviour of the values observed at least once from the initial state of the environment is a development possibility. At this point the rules describing the planned behaviour can be revealed by recording more changes of states. Figure 7 shows the rule representing the completed behaviour of the environment.

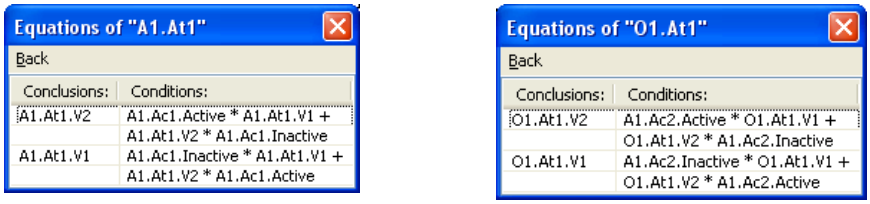


Figure 7. Rules representing the completed behaviour of the environment

4. Eliminating the inconsistencies of the observable environment

If two states of the environment with the same state description lead to states with different state descriptions, then the environment is stochastic. It is because the state of the environment at time *t* cannot be determined by the state of this environment at time *t-1* only. In practice the environment is handled as if it were be stochastic if the condition part of the rules representing the behaviour of the

environment contains numerous state descriptor variables also. This helps to avoid misleading rules. If the agent does not have a sensor to measure a relevant attribute of the environment, then the changes of states become inconsistent. If these sensors cannot be installed, then the agent has to eliminate the inconsistencies by itself. Figure 8 shows a sequence of the changes of states containing inconsistencies. In this Figure each letter with uppercase represents a state of the environment and each letter with lowercase represents the set of activities done by the agents.

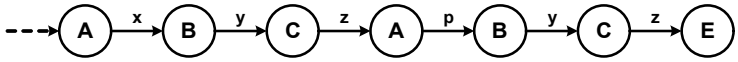


Figure 8. Example of inconsistent changes of states

As Figure 8 shows two different states can be reached from the state denoted by *C* with the same activity. This makes the environment inconsistent. For eliminating the inconsistency an inner state descriptor variable has to be created to distinguish the states be denoted by *C*. Let the values of the inner state descriptor variables denoted by numbers. Therefore the states denoted by *C* can be differentiated into a state denoted by C_1 and a state denoted by C_2.

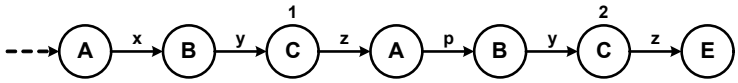


Figure 9. Elimination of some inconsistencies

It eliminates the inconsistency appearing at the states *C* but creates a new inconsistency at states denoted by *B*. These states also have to be distinguished by the inner state descriptor variable as shown in Figure 10.

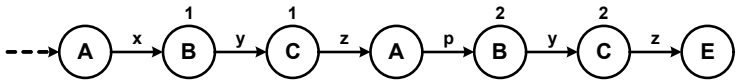


Figure 10. Elimination of some inconsistencies

This method has to be followed until all of the inconsistencies of the changes of states become eliminated.

REFERENCES

[1] RUSSELL, S., NORVIG, P.: *Mesterséges intelligencia modern megközelítésben (2., átdolgozott kiadás). Published by Pearson Education Inc. 2003.*

[2] Szabó, L.: *A nyitott jövő problémája, Véletlen, kauzalitás és determinizmus a fizikában. Typotex Kiadó. 2004.*

[3] KONDOROSI, K., LÁSZLÓ, Z., SZIRMAY-KALOS, L.: *Objektum-orientált szoftverfejlesztés*. ComputerBooks, Budapest, 1999.

[4] GILL, A.: *Introduction to the Theory of Finite-state Machines. New York, McGraw-Hill,1962*.

[5] AHO, A.V., HOPCROFT, J.E., ULLMAN, J.D.: *The Design and Analysis of Computer Algorithms*. Menlo Park, CA: Addison-Wesley, 1974.

[6] CHOW, T.S.: *Testing software design modelled by finite-state machines.* In IEEE Trans. Software Eng., vol. SE-4, no.3, pp. 178-187, Mar.1978.

[7] GOBERSHTEIN, S.M.: *Check words for the state of a finite automation*. In Kibernetika, No. 1, pp. 46-49, 1974.

[8] FRIEDMAN, A.D., MENON, P.R.: *Fault Detection in Digital Circuits*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1971.


Production Systems and Information Engineering
Volume 5 (2009), pp. 67-80.

ANALYSIS OF DYNAMOMETER SIGNALS OF INTERRUPTED CUTTING BY TIME-FREQUENCY LOCALIZATION METHODS

Tamás Kovács
Kalmár Sándor Institute of Information Technology, Kecskemét College
6000 Kecskemét, Hungary
kovacs.tamas@gamf.kefo.hu

Edit Csizmás
Kalmár Sándor Institute of Information Technology, Kecskemét College
6000 Kecskemét, Hungary
csizmas.edit@gamf.kefo.hu

András Szabó
Institute of Metal and Polymer Processing Technology, Kecskemét College
6000 Kecskemét, Hungary
szabo.andras@gamf.kefo.hu

[Received November 2008 and accepted April 2009]

Abstract. In this work the force sensor signals of an interrupted metal cutting process were analyzed by continuous wavelets (CWT) and Hilbert-Huang Transformation (HHT). The main purpose was to characterize the signal through the typical behavior of its dominant frequencies and, at the same time, to compare the performances of the two frequency localizing methods applied. It was found that the low dominant frequencies are approximately constant while the values of the high dominant frequencies (above one kHz) show strong fluctuations. Here we show that the fluctuations revealed by the CWT method are caused partly by numeric effects of the method itself and partly by real variations in the frequency values. These variations are responsible for the non-stationary behavior of the signal. Regarding the methods applied, it was found that the wavelet analysis method is capable of tracing fast varying frequencies of the signal with values close to each other by the proper choice of the central frequency parameter of the mother wavelet. In comparison with the HHT, the wavelet method proved much more robust in this case.

Keywords: Continuous Wavelet Transform, Hilbert-Huang Transform, interrupted cutting

1. Introduction

The analysis of vibration signals is a useful method in research on cutting or other machining processes. A milling or a turning tool together with the

work-piece form a complex mechanical system containing different vibrating sub-systems, the frequencies of which carry relevant information about the whole system. Fourier and wavelet transforms, and recently the Hilbert-Huang Transform (HHT) are the main tools that can be applied in frequency determination. One of the the main motivations behind these examinations is the possibility of tool wear monitoring, diagnostics of tool breakage or other disfunctionalities of the machine. Application of wavelets can be found in various areas of machining processes. In most cases wavelet-based low-pass, high-pass or band filters are developed for signal processing or analyzing purposes. Sheffer and Heyns [1] applied wavelet decomposition of the force signal along with Fourier transform for tool condition monitoring. Li et al. [2] and later Bhattacharyya et al. [3] used wavelet spectrum coefficients for on-line monitoring of the tool wear state in turning and milling processes. Recently Shi and Gindy [4] proposed wavelets to decompose sensory signals into static and dynamic components and extract features of tool malfunctions in various machining processes. The possibilities of localizing varying frequencies in the signal were investigated theoretically in details by Delprat et al. [5] and Torrésani [6].

The other motivation in the force signal analysis is related to the area of active vibration control by a sensor-actuator system. Recently El-Sinawi and Kashani [15] and later Al-Zaharnah [16] designed and implemented such a control system by magnetostrictive actuators in the case of interrupted cutting. Their system does not use any a priori information about the frequencies or other characteristics of the vibration sources. If we had this information or at least part of it, the performance of such a control system could be improved by involving stochastic prediction of the vibration to be damped.

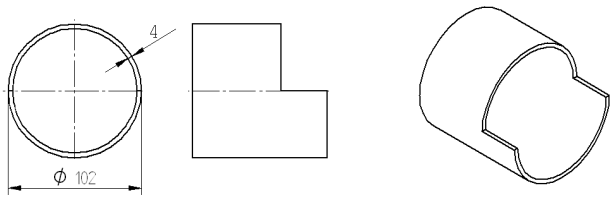


Figure 1. The truncated tube as work-piece

In this work the feed-force signal of an orthogonal interrupted cutting process is studied. In such processes the cutting tool collides with the work-piece with a certain frequency, and this launches the vibration modes of the sub-systems of the whole machine. In addition to this, in the continuous cutting phase

(between the entry and the exit of the cutting insert) there are various physical effects that similarly excite the vibration modes though with much smaller energy. Because of these impulse-like or continuous excitations the force sensor signal contains the frequencies of the activated vibration modes. Here we propose a Continuous Wavelet Transform (CWT)-based analysis of the vibrating force signals that is capable of tracing the fast varying frequencies of the different sub-systems, and, at the same time, it seems to be robust enough. A Hilbert-Huang Transform-based analysis of the signal is also performed in order to compare the two relevant methods.

2. Experimental

In the experimental part an orthogonal cutting arrangement was set by turning the free end of a structural steel tube, the end of which was truncated as shown in Fig. 1. The external diameter and the wall thickness of the tube was 102 mm and 4 mm, respectively. The machine applied was a general purpose double engine (2x5.5 kW) lathe equipped with T25M coated cemented carbide inserts. The geometry of the insert is characterized by a rake angle 5^o and inclination angle 0^o. The feed rate was 0.1 mm per revolution, and the rotations per minute (RPM) of the machine was 270. The applied cutting speed was calculated as

$$v = \pi D n, \tag{2.1}$$

where D is the diameter of the tube and n is the RPM value. By means of Equation (2.1) the value of v was approximately 85 m/min. The magnitude of the force components was measured by a Kistler dynamometer, which was connected to a PCI National Instruments data acquisition card. The sampling frequency of the dynamometer was very high ($200kHz$) in order to get a good resolution of the force signal. The data were processed by LabView 7.1 software. In Fig. 2 the signal produced by the feed force can be seen.

3. Stationarity of the signal

As mentioned above, the analysis of the force signal is important for inventing proper stochastic prediction methods, which can help the vibration control system. The aim of these methods could be summarized as predicting the signal value a certain time in advance. In the present area this time should be a few tenths of milliseconds, since the reaction times of the magnetostrictive or piezoelectric actuators are in this order of magnitude. In the case of stochastic signals the most common technique is the linear prediction method [13]. This

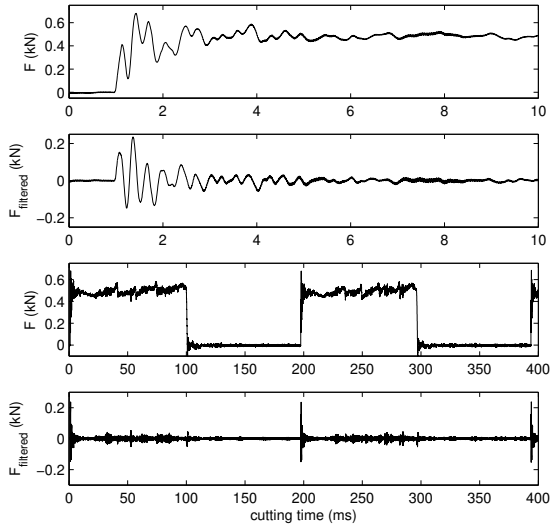


Figure 2. The original and the filtered (see eq. (3.3)) feed force signal measured in the first 10 and 400 milliseconds (two whole turning cycles) at the cutting speed 85 m/min and feed ratio 0.1 mm/rev. The zero of the time axis is taken to be one millisecond before the first workpiece-tool impact.

method assumes that the signal is at least weakly stationary. Among others, the weak stationarity demands that the signal's autocorrelation, defined by

$$X(t, L) = \langle F(t)F(t + L)\rangle\,, \tag{3.1}$$

is independent of time t and depends on only the time lag L. In the definition above $F(t)$ denotes the time dependent signal and the brackets $\langle\rangle$ stand for the normalized mean value. In order to characterize the weak stationarity of $F(t)$ we calculate the approximate value of the autocorrelation inside any time interval of length T, where T is large enough to get a reliable approximation. In mathematical terms, we determine the function defined by

$$X_T(t, L) = \langle F(t)F(t + L)\rangle_{[t-T/2,t+T/2]} = \frac{\int_{t-T/2}^{t+T/2} F(t)F(t + L)dt}{\int_{t-T/2}^{t+T/2}(F(t))^2 dt}. \tag{3.2}$$

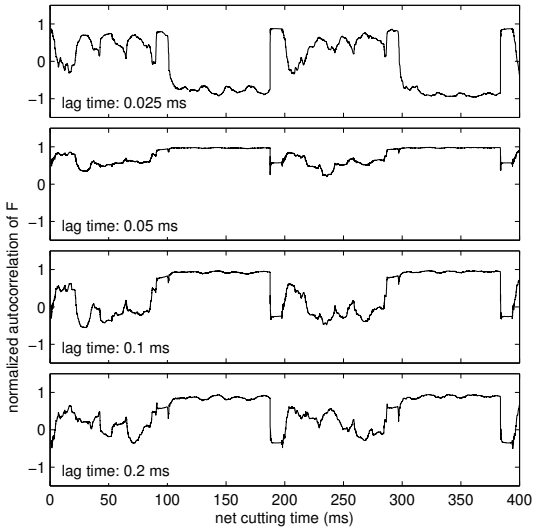


Figure 3. The autocorrelation function of the feed force signal at cutting speed 85 m/min and feed 0.1 mm/rev measured during the first 400 milliseconds (two whole turning cycles) for four different values of the time lag. The time periods of the free running can be seen as relatively constant plateaus in the autocorrelation graph.

By means of the condition mentioned, if the signal is weakly stationary then $X_T(t, L)$ should be a constant function with respect to variable t at any fixed value of L. Since we are interested in only the high frequency component of the signal (in the order of 1 kHz), first the low frequency part is removed from the signal by subtracting its sliding window average, that is

$$F_{filtered}(t) = F(t) - \frac{1}{W}\int_{t-W/2}^{t+W/2} F(\tau)d\tau, \tag{3.3}$$

where the length of the averaging time window (W) was chosen to be 0.1 milliseconds. The function $X_T(t, L)$ was calculated with the filtered feed force signal of six consecutive turning cycles, which means about 600 milliseconds net cutting time (i.e. omitting the free running time intervals), for the fixed values of time lag L of 0.025, 0.05, 0.1 and 0.2 milliseconds. The length of time segment T was chosen to be 10 milliseconds. The resulting four autocorrelation functions (for the four different lag times) can be seen in Fig. 3. The time segments of the free running of the tool were cut out from the signal. Since the reaction times of the piezoelectric or magnetostrictive actuators are a few tenth

of milliseconds, the last two graphs are the most interesting when $L = 0.1$ and 0.2 milliseconds. It can be seen that the graphed autocorrelation functions are far from being constant at any values of the time lag, so the signal should be considered non-stationary. In the case of the smallest lag ($L = 0.025$) there are periodic sharp deflections at the transient phases when the tool collides with the workpiece, but the fluctuations in other time intervals are also remarkable though not so outstanding as in the transient phase. For higher time lags the fluctuations became so intensive that the periodic deflections of the transient phase cannot be observed. These results indicate that the linear predictive methods with constant coefficients cannot be applied successfully for such cutting force signals, moreover, because of the fast variation of the autocorrelation value it is difficult to find even a time dependent adaptive linear prediction method. What are the basic reasons of such a bad non-stationary behavior of the signal? Since the normalized autocorrelation is more or less independent of the variations of the amplitude [14], the answer should be searched for in the variations of the dominant frequencies in the high frequency domain.

4. The tools of the frequency analysis

For accomplishing the wavelet analysis of the signal the symmetric Morlet function was chosen as mother wavelet, defined by

$$\phi(t) = \frac{1}{\sqrt{2\pi}}\, exp\left(-\frac{1}{2}t^2 + i\omega_0 t\right). \tag{4.1}$$

The mother wavelet is scaled by a parameter a so as to get a characteristic frequency ω_0/a, and it is shifted in the time domain by a parameter b. Thus a set of wavelets (child wavelets) with different frequencies and time-positions is obtained as follows

$$\phi_{a,b}(t) = \frac{1}{\sqrt{2\pi a}}\, exp\left[-\frac{1}{2}\left(\frac{t-b}{a}\right)^2 + i\omega_0\left(\frac{t-b}{a}\right)\right]. \tag{4.2}$$

The coefficients of the CWT are obtained by calculating the convolution integral:

$$\hat{F}(a,b) = \int_0^\infty \phi^*_{a,b}(t)\,(F(t) + iH[F(t)])\,dt, \tag{4.3}$$

where $H[F(t)]$ is the Hilbert transform of the signal and * stands for the complex conjugation. The expression $F(t)+iH[F(t)]$ is generally addressed as the

analytical form of $F(t)$. The most simple way of determining the dominant modes in the frequency and time scale is to find the local maximum places of $\left|\hat{F}(a,b)\right|^2$, which is regarded as the energy density function of the vibration [5],[6],[7]. The choice of the parameter ω_0 in the mother wavelet is crucial in the present task. This parameter determines the number of observable periods in the mother and also in the child wavelets. It can be easily seen that for bigger values of ω_0 (many periods in the wavelet) the selectivity of the wavelets in the frequency domain is better, i.e., the $\delta\omega$ error of frequency localization is smaller but the δt error of that in the time domain is bigger, and the contrary case is true when we choose a small value of ω_0 (few periods in the wavelet). Unfortunately, the two errors cannot be small at the same time, similarly to Heisenberg's law in quantum mechanics. In this work the $\hat{F}(a,b)$ spectra were calculated with more different (small and big) parameter values so as to get good resolution in the time and frequency domain, though not at the same time.

The HHT is based on completely different theoretical methods. In the first phase the so-called Empirical Mode Decomposition (EMD) decomposes the signal into approximately monochromatic, though possibly frequency and amplitude modulated components [8]. Then the instant frequencies are obtained by derivating the phase (argument) of the analytical components with respect to time, that is

$$\omega_k(t) = \frac{d}{dt}\, arg\left(C_k(t) + iH[C_k(t)]\right), \tag{4.4}$$

where $C_k(t)$ is the kth component. There are a number of improvements of the original algorithm, especially regarding the EMD [9],[10],[11]. Here we apply the algorithm recently proposed by Rilling et al. [10]. The HHT, being a differential method, is able to give the instant frequency at a specific point of the time-scale, while the "instant" frequency obtained by CWT is localized in a finite time interval, which the envelope of wavelet spans over. However, a serious drawback of this method is that the EMD is of limited capability when components with frequencies close to each other are to be decomposed [8],[14].

5. Results and discussion

The results of the CWT and HHT analysis of the signal in Fig. 2 are presented here. The Fourier Transform of the signal measured during a complete turning cycle (approximately 80 ms) is seen in Fig. 4. There are four dominant

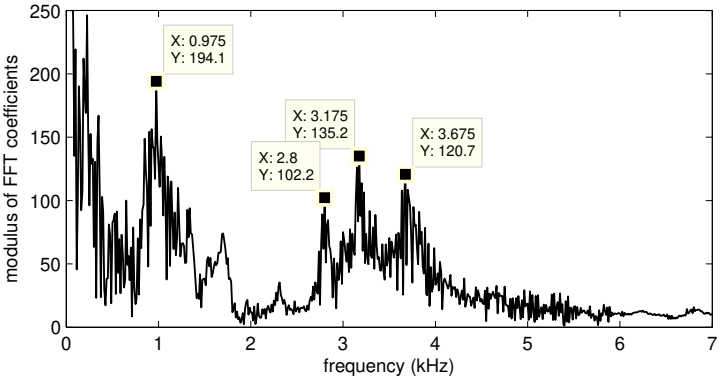


Figure 4. Fourier Transform of the analytical form of the feed force signal at cutting speed 85 m/min and feed 0.1 mm/rev measured during a complete turning cycle (80 milliseconds)

peaks in the region of higher frequencies at $f_1 = 1.0$, $f_2 = 2.8$, $f_3 = 3.2$ and $f_4 = 3.7$ kHz, where the frequency f is related to the angular frequency ω as $f = \omega/(2\pi)$. The latter three values are relatively close to each other, all of them being in the band of 2.5–4 kHz. Separating frequencies with values close to each other is a hard task for any known time-frequency localization methods [8]. In the case of a signal that consists of several components with frequency values constant but close to each other the CWT, because of the imperfect resolution in the frequency domain, leads to more or less fluctuating time-frequency functions. Therefore, in the case of the present force signal we should expect some fluctuations in the frequency values, which is not 'real' but a purely numeric effect.

In order to get a qualitative picture about this kind of numeric effect, an $F_{test}(t)$ test signal is constructed as a sum of three harmonic signals with the constant frequencies f_1, f_2, f_3 and f_4, and the ratios between their amplitudes are chosen to be the same as those of the FFT coefficients in Fig. 4, i.e.

$$F_{test}(t) = \sum_{i=1,2,3,4} a_i cos(2\pi f_i t), \tag{5.1}$$

where $a_1 = 1.9$, $a_2 = 1.0$, $a_3 = 1.4$ and $a_4 = 1.2$. This signal belongs to a dynamic system of four undamped harmonic subsystems without coupling. Fig. 5 shows the results obtained by applying the CWT-based method detailed in the previous section on the test signal with the ω_0 values of 10, 15, 20 and 25. The CWT results are generally given by a level curve or color-coded graph of

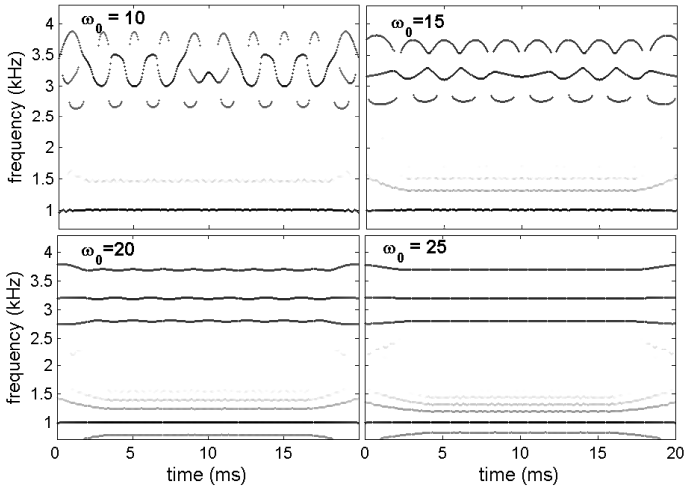


Figure 5. Local maximum places (at each fixed time value) of the energy density function obtained by CWT of the test signal given by (5.1) for four different values of ω_0. The tone of the marker specs is linearly proportional to the logarithm of the local energy density.

the energy density function $\left|\hat{F}(a,b)\right|^2$. In the present work, however, only the places of the local maxima belonging to a fixed value of time are plotted in the time-frequency plane indicating the energy density at the local maxima by the darkness of the marker. To be more specific, the tones of the marker specs are linearly proportional to the logarithm of the local energy density, the highest and the lowest energy peaks corresponding to pitch black and white tones. Though this way of the presentation carries poorer information compared to the conventional ones, it gives a clearer picture about the behavior of the dominant frequencies. The fluctuations of the frequency values in these graphs are obviously caused by the numeric effect described above. However, these numeric fluctuations almost vanish when the value of the ω_0 parameter is increased to 20, and there is no considerable change for the further increase of ω_0. Therefore, for the value of 20 of ω_0 the resolution is satisfactory in separating the present dominant frequency values with the measured ratios of the amplitudes. However, the case is different for the lowest frequency component. Since it is far enough from the other three in the frequency scale, its frequency is measured to be approximately constant even for the smallest value of ω_0. Besides, false components appear for any values of ω_0 above the frequency f_1, which is due to an aliasing effect. The higher ω_0 is, the more pronounced this effect is.

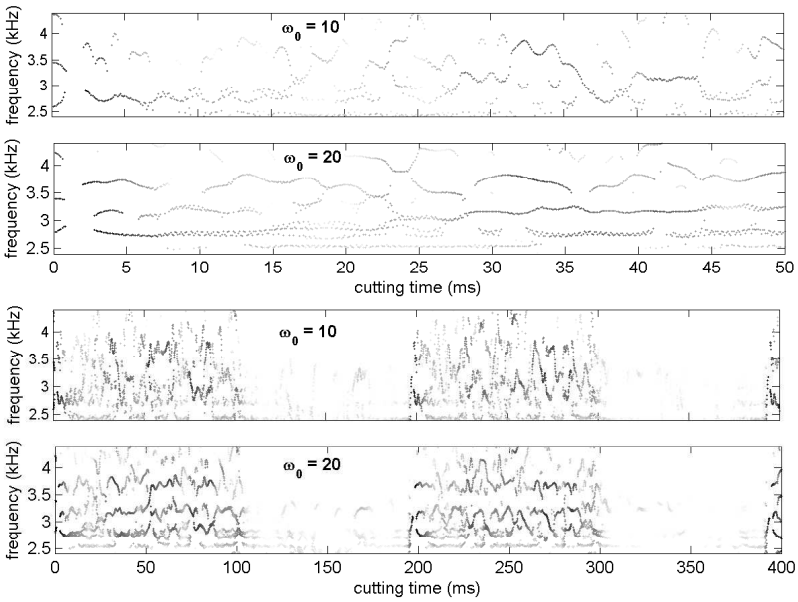


Figure 6. Local maximum places (at each fixed time value) of the energy density function obtained by CWT of the measured signal in the first 10 and 400 milliseconds for two different values of ω_0. The chosen frequency region in the present case was 2.5-4.5 kHz. The tone of the marker specs is linearly proportional to the logarithm of the local energy density.

The feed force signal investigated was subject to the same CWT calculations with values of 10 and 20 of the self frequency parameter ω_0. Fig. 6 and Fig. 7 show the results of the calculations for the higher and lower frequency regions. The figures were constructed by the same method as that applied in the case of Fig. 5. By means of experiences learned from the CWT of the test signal, the graphs of $\omega_0 = 10$ are contaminated by numerical fluctuations if there are frequency values close to each other, and, in addition to this, there can be 'real' variations of the frequency values. Nevertheless, the numeric fluctuations almost vanish when the ω_0 self frequency parameter is increased to 20. In the figure of the lower frequency region (Fig. 7) we can discover the dominant frequency of $f_1 = 1.0$ kHz together with false aliasing frequencies. There are no serious fluctuations of any kind, which means that there are no real dominant frequencies close to f_1, and this component of the signal has approximately constant frequency. Regarding the higher frequency region the case is completely different: there are large fluctuations of the three dominant frequency values for both cases of $\omega_0 = 10$ and $\omega_0 = 20$. The false (only

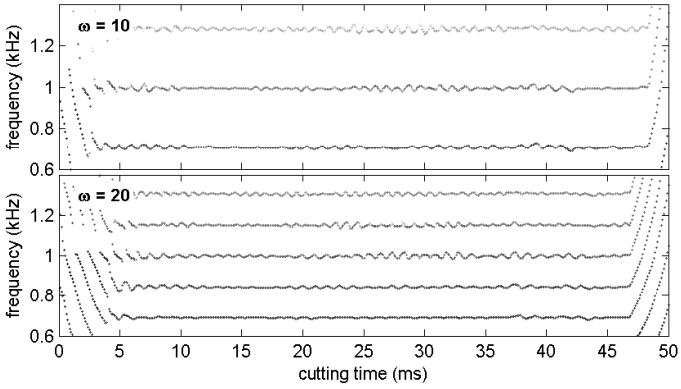


Figure 7. Local maximum places (at each fixed time value) of the energy density function obtained by CWT of the measured signal for two different values of ω_0 in the frequency region 0.6-1.4 kHz. The tone of the marker specs is linearly proportional to the logarithm of the local energy density.

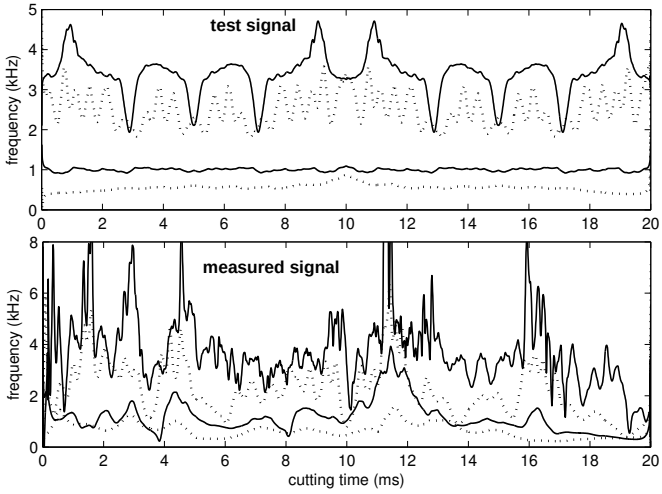


Figure 8. Frequencies of the four relevant components in the first 20 millisecond segment of the test and the measured signals obtained by HHT (continuous and dashed lines)

numeric) fluctuations were expected for $\omega_0 = 10$ because of the frequency values being relatively close to each other, however, when $\omega_0 = 20$ there are still serious fluctuations. This means that the vibration modes in the upper frequency region have physically varying frequencies. This real variation in

the frequency values can cause the non-stationary behavior of the force signal. It is also interesting that the energy seems to flow between the main vibration modes (remember, the black line segments are the high energy parts). This indicates that there are couplings between the three vibration modes. This can be one of the causes of the frequency modulations. Obviously, there may be other non-linear physical phenomena behind the frequency modulations as well. The investigation of the specific physical causes is beyond the scope of this paper.

For the sake of the comparison of the two relevant methods the HHT transform of the test and the measured signal at hand were calculated as well. The algorithm published electronically as a MATLAB package by Rilling et al. [10] was used here with stopping error limits ten times smaller than the default values set in the program package. Fig. 8 shows the time-frequency functions of the four components obtained by the HHT in the frequency region above 1 kHz. The fluctuations of the frequencies given by this method are much higher, which is partly due to the high precision local property of the HHT. In other words, this method is much more sensitive to the different kinds of noises and therefore seems to be much less robust than the CWT. It is to be noted that the HHT method does not give constant frequency-time functions even for the constant frequencies of the test signal (upper figure). Recently Rilling and Flandrin [12] investigated the theoretical limitations of EMD and they found that this method in its present form cannot separate two harmonic signals if the frequencies are close to each other or the lower frequency component has a considerably bigger amplitude than the higher one. This is the other reason for the wide fluctuations of the frequency values detected by the HHT. In the case of the test signal the two lower frequencies are distinguishable while the upper two are not. The reason for this is that the upper two frequencies are relatively too close to each other to apply the EMD method successfully (for a quantitative analysis see [12]). In addition to this, in the graph corresponding to the measured signal it is impossible to discover the dominant separate frequency peaks observed in the FFT graph, while the CWT method can identify these dominant frequencies.

6. Conclusions

The measured interrupted cutting force signal proved to be non-stationary with relatively fast varying autocorrelation, and this can be a serious drawback in the stochastic prediction of the signal. The CWT examinations showed that non-stationarity is caused by the considerable frequency modulations of the vibration modes in the frequency region above 2 kHz. An effective prediction

of the signal in this frequency region can be successful only if the physical reasons of the frequency fluctuations are analyzed in details. The CWT method proposed in the present work can be a helpful tool for such a physical analysis, since with its help the time evolution of the main frequency values and the energy content of the different modes can be monitored effectively. Naturally, to obtain a complete picture about the dynamics of the system, other types (acoustic and accelerometer) of measurements are also needed. This is left to a future work.

From the point of view of the numerical methods the results led us to the conclusion that the CWT should remain an important tool in time-frequency analysis problems, especially when the signal contains frequency and amplitude modulated vibrations in a relatively narrow frequency band. Besides, calculating the wavelet spectrograms for higher values of the self frequency parameter (ω_0) of the mother wavelet proved to be useful in separating real (physical) frequency variations from numeric fluctuations.

Acknowledgements

The support provided by the Annual Research Foundation of the Kecskemét College is acknowledged. We would like to thank Norbert Földvári for his help in editing our figures.

REFERENCES

[1] Sheffer, C., Heyns, P. S.: *Wear monitoring in turning operations using vibration and strain measurements*, Mechanical Systems and Signal Processing, **15/6**, (2001), 1185–1202.

[2] Li, X., Guan, X. P.: *Time-frequency analysis-based minor cutting edge fracture detection during end milling*, Mechanical Systems and Signal Processing, **18**, (2004), 1485–1496.

[3] Bhattacharyya, P., Sengupta, D., Mukhopadhyay, S.: *Cutting force-based real-time estimation of tool wear in face milling using a combination of signal processing techniques*, Mechanical Systems and Signal Processing, **21**, (2007), 2665–2683.

[4] Shi, D., Gindy, N. N.: *Development of an online machining process monitoring system: Application in hard turning*, Sensors and Actuators A, **135**, (2007), 405–414.

[5] Delprat, N., Escudié, B., Guillemain, P., Kronland-Martinet, R., Tchamitchian, Ph., Torrésani, B.: *Asymptotic Wavelet and Gabor Analysis: Extraction of Instantaneous Frequencies*, IEEE Trans. Inf. Th., **38**, (1992), 644–664.

[6] Torrésani, B.: *Time-Frequency analysis from geometry to signal processing*, Proceedings of the COPROMATH conference (November 1999, Cotonou, Benin), J. Govaerts, N. Hounkonnou and W. A. Lester Eds., World Scientific, (2000), 74–96.

[7] Carmona, R., Hwang, W. L., Torrésani, B.: *Characterization of Signals by the Ridges of their Wavelet Transform*, IEEE Trans. Signal Processing, **45/10**, (1997), 2586–2590.

[8] Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., Yen, N., Tung, C. C., Liu, H. H.: *The empirical mode decomposition and the Hilbert Spectrum for nonlinear and non-stationary time series analysis*, Proc. R. Soc. Lond. A, **454**, (1998), 903–995.

[9] Peng, Z. K., Tse, P. W., Chu, F. L. : *An improved Hilbert-Huang transform and its application in vibration signal analysis*, J. Sound and Vibration, **286**, (2005), 187–205.

[10] Rilling, G., Flandrin, P., Goncalves, P.: *On Empirical Mode Decomposition and its algorithms*, IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing NSIP-03, Grado (I), (2003).

[11] Yinfeng, D., Yingmin, L., Mingkui, X., Ming, L.: *Analysis of earthquake ground motions using an improved Hilbert-Huang transform*, Soil Dynamics and Earthquake Engineering, **28**, (2008) 7–19.

[12] Rilling, G., Flandrin, P.: *One or Two Frequencies? The Empirical Mode Decomposition Answers*, IEEE Trans. Signal Processing, **56/1**, (2008), 85–95.

[13] Hayes, M. H.: *Statistical Digital Signal Processing and Modeling.* , J. Wiley & Sons, Inc., New York, (1996).

[14] Rubio, E. M., Teti, R., Baciu, I. L.: *Advanced Signal Processing in Acoustic Emission Monitoring for Machining Technology*, Intelligent Production Machines and Systems,Ed. by D.T.Pham, E.E.Eldukhri and A.J.Soroka, Elsevier Ltd., Cardiff, UK, (2006).

[15] El-Sinawi, A. H. and Kashani, R.: *Improving Surface Roughness in Turning Using Optimal Control of Tool's Radial Position*, J. Materials Processing Technology,**167**, (2005), 54–61.

[16] Al-Zaharnah, I. T.: *Suppressing Vibrations of Machining Processes in Both Feed and Radial Directions Using an Optimal Control Strategy: The Case of Interrupted Cutting*, J. Materials Processing Technology,**172**, (2006), 305–310.

Production Systems and Information Engineering
Volume 5 (2009), pp. 81-94.

SOLVING MULTI-OBJECTIVE PRODUCTION SCHEDULING PROBLEMS USING A NEW APPROACH

GYULA KULCSÁR
University of Miskolc, Hungary
Department of Information Engineering
kulcsar@ait.iit.uni-miskolc.hu

MÓNIKA KULCSÁRNÉ FORRAI
University of Miskolc, Hungary
Department of Automation
kulcsfm@mazsola.iit.uni-miskolc.hu

[Received March 2009 and accepted April 2009]

Abstract. Most companies use a proactive approach to schedule production orders and jobs at shop floor level. To make a near-optimal schedule for a shop with different types of machines and many operations is a very important but complicated task because of the very large number of alternative solution in the searching space. Advanced scheduling models, good heuristics and fast improving algorithms are required to satisfy constraints and to optimize production performances. The aim of the paper is to outline a new modelling and solving approach connected to discrete production scheduling and rescheduling.

Keywords: scheduling, multi-objective optimization, simulation, production

1. Introduction

Today, production engineering and management utilize more and more computer-integrated application systems to support decision making. Software systems have been applied to manage discrete production processes. These can be classified into four hierarchical groups according to the supported fields: (1) Enterprise Resources Planning (ERP), (2) Computer Aided Production Engineering (CAPE), (3) Manufacturing Execution Systems (MES) and (4) Manufacturing Automation (MA). This paper focuses on the detailed scheduling function of Manufacturing Execution Systems. At MES level, the main purpose of the fine scheduler is to initiate a detailed schedule to meet the master plan defined at ERP level. The scheduler gets the current data of dependent orders, products, resource environment and others technological constraints (tools, operations, buffers, materials handling and so on). The shop floor management defines the production goals and their priorities. Obviously, the management may declare different goals at different

times. The scheduler has to provide a feasible sequence of jobs which meets the management's goal. The result of the scheduling is a detailed production programme which declares the releasing sequence of jobs, assigns all the necessary resources to them and proposes the starting time of operations. It must not break any of the hard constraints but has to meet the predefined goals. The computation time of the scheduling process is also an important issue especially with a large number of internal orders, jobs, operations, resources, technological variants and constraints.

In discrete manufacturing, series of goods are produced. A series (batch or lot) can include highly different number of pieces, from a single product (e.g. special part, complex or unique equipment) to thousands or millions of the same product (simple parts or goods). In discrete manufacturing operations are executed on discrete, separated machines and workplaces. Depending on the arrangement of machines, buffers and transportation devices, manufacturing systems may have a line type or group type layout. In essence the execution of the operations for mass production or customized mass production requires the exact predefinition of the routing of the operations. This kind of model is called flow shop (FS) model.

In a FS model there are machines in a theoretical line structure, placed one after the other in the order predefined by the technology. The model is highly influenced by the presence of buffers between machines. If the capacity of the buffers is zero, then the system is a strictly synchronized transfer system, othervise the system is called asynchronous transfer line system. A permutation flow shop model is a special variant of the FS models, it means that the job-queues in front of each machine operate according to the FIFO (First In First Out) principle.

In the literature, different advanced variants of the classical FS models can be found. One of the main groups of these models is the flexible flow shop (FFS) scheme [1, 4, 6, 7, 8, 9, 14, 16]. The FFS environment consists of stages which represent the fundamental operation-type units of the manufacturing / assembly system. At each stage one or more identical machines work in parallel. Each job has to be processed at each stage on any of the parallel concurrent machines. In respect of production performance, both the allocation of machines and the order of jobs are of great importance. Lots of flexible flow shop models are known in the literature, but most of them use only one performance measure (objective function). Usually the latest finishing time (makespan) of the released jobs appears as goal function of optimization for make to stock (MTS) manufacturing. Frequently one of objective functions related to due date plays the main role in scheduling models for supporting make to order (MTO) manufacturing. Only a few of the models deals with multi-objective cases which are more important in flexible and agile manufacturing. According to the customized mass production (CMP) paradigm the firms plan their production partially for external direct orders. Additionally, to reach better delivery capabilities they make forecasts for manufacturing to make components, master units or semi-finished products for

stock. The flexible scheduling models – known in the literature – do not meet all the requirements of CMP to the expected extent. The existing models often disregard the machine processing abilities, alternative technological routes, limited availability time of the machines, limited buffer capacities, shared machine tools, so an improvement and extension of flexible flow shop models are required.

2. A New Advanced Fine Scheduling Approach

2.1. An Extended Flexible Flow Shop Model (EFFS)

The discrete manufacturing process examined produces various consumer goods. By means of forecasting tools which consider external orders, market trends, seasonal characteristics a set of internal orders has been created by the production planners. Each order defines the required number of identical products of a certain product type, which should be manufactured by the predefined time. The logistic unit is the palette at the shop floor level, which can take one or more products. Internal orders consist of one or more (i.e. whole number of) palettes. Depending on the product type, palettes carry a predefined number of identical products. **O**rders can be considered the set of palettes to manufacture, where the number of palettes depends on the ordered product quantity and the capacity of the palettes. The model being shown in this section applies manufacturing / assembly machine objects (individual machines and/or machine lines). Machine lines perform several technological steps (TS). Each TS means a sequence of elementary operations and cannot be interrupted. Consequently a TS is the smallest allocation unit during the scheduling. A job means one or more palette of an internal order with technological steps to be executed in a predefined sequence. The nature of flexible manufacturing is that same goods can be manufactured using alternative materials, components, machines or technological routes. The capacity of the buffers placed among the machines can be zero, limited or not limited. The limit size of a buffer may depend on the product types.

In EFFS model, every machine can be characterized by product sequence dependent setup times, availability time frames, various production rates depending on product types, and capability for performing a single step or a sequence of steps for certain products. Machines can be arranged into machine groups according to processing ability. A machine group is a set of machines that can execute the same execution step (sequence of technological steps). This point of view, a given final product or a semi-finished product can be produced differently using different sequences of machine groups which the required components are taken through. The flow shop nature of the model means that each execution route may consists of one or more execution steps, but the common part of the execution steps has to be an empty set (overlapped technological step is not allowed). Moreover the sequence of technological steps is determined by a strict direction and an execution

step has to be included in all the technological steps which are between the first and last steps of the machine.

In order to formulate the new class of scheduling problems described above, the well-known formal specification $\alpha|\beta|\gamma$ is used, where α denotes machine environment descriptors**,** β denotes processing characteristics and constraints, and γ denotes the list of objective functions**.** An extended Flexible Flow Shop (EFFS) scheduling model can be described as follows:

$$\mathrm{Fx}, \mathrm{M_g}, \mathrm{Q_{i,m}}, \mathrm{Set_{i,j,m}}, \mathrm{Cal_m}, \mathrm{B_{m,p}}, \mathrm{TR_{m,n}} \left| \mathrm{R_i}, \mathrm{D_i}, \mathrm{Exe_i}, \mathrm{A_i} \right| \mathrm{f_1}, \mathrm{f_2}, ..., \mathrm{f_K} \tag{2.1}$$

$$\alpha = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7\} \tag{2.2}$$

α_1: Type of operation (technology step) sequence at shop floor level. *F*: Flow Shop, *x*: the maximum number of operations.

α_2: Type of machine environment. M_g: group of multi-purpose machines which can execute one or more operations in a given sequence. Each machine group can include distinct parallel machines.

α_3: Type of alternative machines. $Q_{i,m}$: unrelated parallel machines with job dependent production rates.

α_4: Type of machine setups. $Set_{i,j,m}$: job sequence and machine dependent setup times.

α_5: Special resource constraint. Cal_m: machine availability time intervals.

α_6: Special buffer constraint. $B_{m,p}$: product type dependent capacity of the buffer placed in front of machines.

α_7: Transportation time. $TR_{m,n}$: job travelling time between given machines.

$$\beta = \{\beta_1, \beta_2, \beta_3, \beta_4\} \tag{2.3}$$

β_1: Constrained released time of jobs. R_i: the earliest start time of jobs.

β_2: Constrained due date of jobs. D_i: the latest completion time of jobs.

β_3: Type of manufacturing processes. Exe_i: required type and sequence of technology steps for jobs.

β_4: Constrained resource assignments. A_i: set of suitable machines for jobs.

$$\gamma = \{\gamma_1, \gamma_2, ..., \gamma_K\} \tag{2.4}$$

The goodness and quality of the schedule can be evaluated using the numerical result of the objective function. Some examples are listed in Table 1. Real manufacturing environments may require various objective functions declared.

The extended flexible flow shop scheduling problem is difficult to solve because of its combinatorial nature. The model inherits the difficulties of the classical flow shop and the flexible flow shop models. Additionally, numerous odd features appear because of special extensions.

Table 1. Typical objective functions for detailed scheduling

C_{max}	Finishing time of last job (makespan) to be min.
L_{max}	Max lateness to be min.
T_{max}	Max tardiness to be min.
S_{max}	Max square distance of differences to due dates to be min.
$\sum(C_i-r_i)$	Sum of throughput times to be min.
$\sum L_i$	Sum of lateness times to be min.
$\sum T_i$	Sum of tardiness times to be min.
$\sum U_m$	Sum of machine utilization to be max.
$\sum v_i(C_i-r_i)$	Weighted sum of flow times to be min.
$\sum v_i L_i$	Weighted sum of lateness times to be min.
$\sum v_i T_i$	Weighted sum of tardiness times to be min.
N_{WIP}	Average number of work in progress to be min.
N_{SET}	Number of setups to be min.
$\sum T_{SET}$	Sum of setup times to be min.

2.2. Comparison of Solutions Based on Relative Quality

In the literature, different approaches can be found considering multi-objective scheduling problems, as they are surveyed i.e. in [3, 11]. In this section a new approach will be shown. According to our proposal the relative goodness of a solution is more important than its absolute goodness. The basis of the approach is the following: the relative goodness of the selected solution is measured by comparing it with another solution in the feasible solution space.

Let S be the search space under consideration. It is the set of all possible solutions of the problem. Suppose that a number of objective functions $f_1, \ldots, f_K$ is given such that:

$$f_k : S \to \Re^+ \cup \{0\}, \ \forall k \in \{1,2,...,K\} . \tag{2.5}$$

The problem is to find an $s \in S$ that minimizes every $f_k(s)$. This is known as a multi-objective optimization problem. In the majority of cases, it is not possible to find a solution to a multi-objective optimization problem. Successfully minimizing one of the component objective functions will typically increase the value of another one. So we must find solutions that represent a compromise between the various criteria used to evaluate the quality of solutions.

Let s_x, $s_y \in S$ be two solutions. The function F is defined to express the quality of s_y compared to s_x as a real number:

$$F : S^2 \to \Re\,,\ F(s_x, s_y) = \sum_{k=1}^{K} (w_k \cdot D(f_k(s_x), f_k(s_y)))\,. \tag{2.6}$$

The definition of function D is the following:

$$D : \Re^2 \to \Re\,,\ a,b \in \Re\,,\ D(a,b) = \begin{cases} 0, if\ \max(a,b) = 0 \\ \dfrac{b-a}{\max(a,b)} \cdot 100,\ otherwise \end{cases}. \tag{2.7}$$

The *max(a,b)* used in (2.7) denotes an operator:

$$\max : \Re^2 \to \Re\,,\ \max(a,b) = \begin{cases} a,\ if\ a > b \\ b,\ otherwise \end{cases}. \tag{2.8}$$

Moreover, coefficient w_k which is an integer value within range *[0, 1, …, W]* is used in order to express the importance of any component objective function f_k. It is allowed that the decision maker sets the actual priority of each objective function independently.

Using the function F, two solutions s_x, $s_y \in S$ are compared as follows:

s_x is a better solution than s_y ($s_x < s_y$ is true) if

$$F(s_x, s_y) > 0\,. \tag{2.9}$$

s_x and s_y are equal ($s_x = s_y$ is true) if

$$F(s_x, s_y) = 0\,. \tag{2.10}$$

s_x is a worse solution than s_y ($s_x > s_y$ is true) if

$$F(s_x, s_y) < 0\,. \tag{2.11}$$

These definitions of the relational operators are suitable for applying in meta-heuristics like taboo search, simulated annealing and genetic algorithms to solve multi-objective combinatorial optimization problems.

In this model, the *max* operator (2.8), which means the basis of the comparison, may be replaced by a general function *g* but the essence of the proposed approach does not change. The relative qualification of two solutions remains if the new function *g* is characterized by the following simple feature (2.12):

$$g : \Re^2 \to \Re \,,\ g(a,b) = g(b,a) \,. \tag{2.12}$$

The mathematical model described above was developed to establish a method for managing the objective functions and evaluating the relative quality of the feasible solutions by comparing them to each other. The model can be widely used for solving multi-objective combinatorial optimization problems which include objective functions characterised by dynamically varying importance, different dimension and value range.

2.3. Numerical Simulation and Evaluation of Schedules

Manufacturing processes can be characterized by general state variables. In respect of production management the most important three state variables for measuring performance are as follows: (1) stock level, (2) capacity utilization and (3) readiness for delivery [12]. For solving scheduling problems we use fast computer simulation to evaluate the schedules. It considers the availability of the individual machines for a given time window and the required setup times between the series. Using the machine-job assignments the processing time of tasks can be calculated. For each job and each internal order the starting and finishing time can be defined by using the time data of all the tasks. By means of the simulation the objective functions can be evaluated as well.

The proposed algorithm means numerical simulation of the production to calculate the time data of the execution of tasks. Inputs of the simulation consist of jobs, machines, their assignments, sequences of jobs on machines, buffer capacities, abilities of machines, availabilities of machines, transportation time of jobs between machines. Simulation of a given job on an intermediate machine requires, among other things, the completion time of the job on the previous machine and the previous job on the machine, moreover the shop floor environment has lots of junctions of the possible routes. So it has to define the sequence of tasks in which the calculation can be performed correctly. To satisfy this requirement we developed a fast process oriented algorithm that works in an event driven way.

The time values of a given job (task) on an assigned machine are mainly determined by (1) the constraint start time of the job (in point of view of components availabilities), (2) completion time of the job on the previous machine,

(3) completion time of the previous job on the machine, (4) transportation time of the job taken from the previous machine, (5) setup time of the job on the machine, (6) the availability of the machine (availability time frames and product dependent production intensities), (6) actual state of the buffer at back of machine and (7) availability of tools needed (which can be shared).

The numerical tracking of the product-palettes supplies the time data of the manufacturing steps such as starting time, setup time, processing time and finishing time of tasks, jobs and internal orders. The simulator extends the predefined schedule (job-resource assignments and job sequences on machines) to a fine schedule by calculating and assigning time data. Consequently the simulation is able to transform the original searching space into a reduced space by solving the timing problem.

2.4. Integrated Fine Scheduling and Rescheduling Software

Meta-heuristics (i.e.: genetic algorithms, simulated annealing and taboo search) are becoming more and more successful methods for optimization problems that are too complex to be solved using deterministic techniques [3, 7, 10, 11, 15]. In general, the scheduling task consists of batching, assigning, sequencing and timing because of the complexity of the problem. We developed a new integrated approach to solve all these sub-problems as a whole without decomposition. In this approach, all the issues are answered simultaneously (Figure 1).

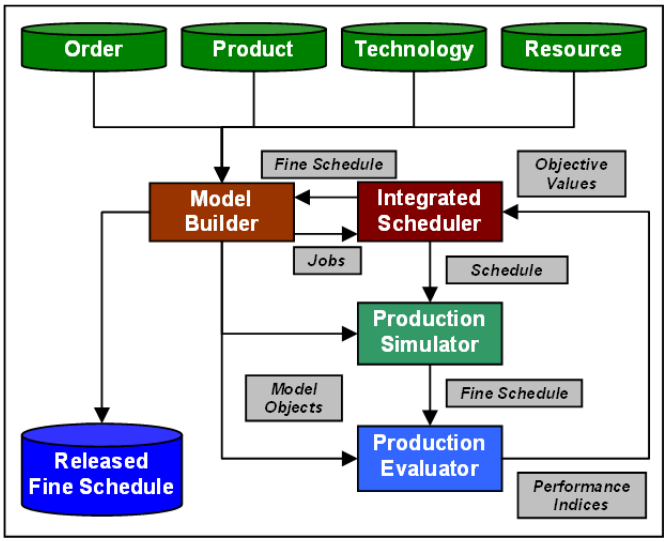


Figure 1. Integrated fine scheduling approach

The new integrated approach based method supports the decision making of joining and/or dividing production orders; the calculation of the manufacturing lot

sizes dynamically; the selection of the alternative technological routes; the allocation of machine resources; the definition of manufacturing tasks and the scheduling of its execution processes. This method uses heuristic algorithms, searching techniques and problem space transformation based on discrete events type simulation.

To accelerate the computation an indexed data model has been elaborated. The data structure supports the association of two or more different types of arrays. The model builder creates the full indexed data model which includes the possible technology and resource alternatives.

In the approach applied the product-pallet plays the role of the basic scheduling unit. Each production order consists of pallets that mean individual jobs (one or more pieces with execution steps required). The production batch sizes are formed dynamically by scheduling the jobs on machines.

In order to create a detailed schedule for the production of each internal order, it is necessary for each job: (1) to be assigned to one of the possible routes, (2) to be assigned to one of the possible machines at each possible machine group according to selected route, (3) to fix its position in the queue of each selected machine, and (4) to fix its starting time on each selected machine.

The solving algorithms are integrated into a scheduler engine. Two classes of heuristic algorithms are used in two phases. In the first phase, constructive algorithms based on combined heuristic priority rules create good initial solutions. In the second phase, iterative searching algorithms improve the best solution according to the multiple objectives. The method focuses on creating near-optimal feasible schedules considering multiple objectives and it is based on a special taboo search variant. A certain number of neighbours of the current schedule are generated at random successively by using priority controlled neighbouring operators. These operators create new feasible schedules by modifying resource allocations and job sequences. It is not necessary to check the feasibility of the generated solutions because the neighbouring operators make valid modifications by choosing allowed alternatives from the indexed model structure. Moreover, an advanced structure of taboo-list is used. Taboo-list contains the schedules that have been visited in the recent past (less than a given number of moves ago). Schedules in the taboo-list are excluded from the neighbourhood of the actual solution.

The objective functions concerning schedules are evaluated by the production simulator which represents the discrete production environment (machines with their capabilities, buffers with their capacities and others technological constraints). The production evaluator measures the performance of the fine schedules by calculating management indices based on job, order and machine data. The mathematical model proposed in Section 2.2 for relative qualification is used for comparing the generated schedules according to multiple objectives. The best schedule becomes the initial solution of the next loop of the searching algorithm.

When the scheduling process has been finished or stopped by the user, the current best schedule is returned.

In managing real production systems, different types of uncertainty may occur e.g. machine failure or breakdown, missing material or components, under-estimation of processing time, job priority or due date changes and so on. Different rescheduling methods can be used according to the effects of the unexpected events: time shift rescheduling, partial rescheduling or complete rescheduling [2, 13]. Rescheduling is a process of updating an existing production schedule in response to disruptions or creating a new one if the current schedule has become infeasible.

Our approach is able to solve rescheduling tasks using multi-objective searching algorithms similarly to predictive scheduling (Figure 2). The aim of rescheduling is to find a schedule, which (1) considers the modified circumstances, (2) is near-optimal according to some predefined criterion and (3) is as close as possible to the original one.

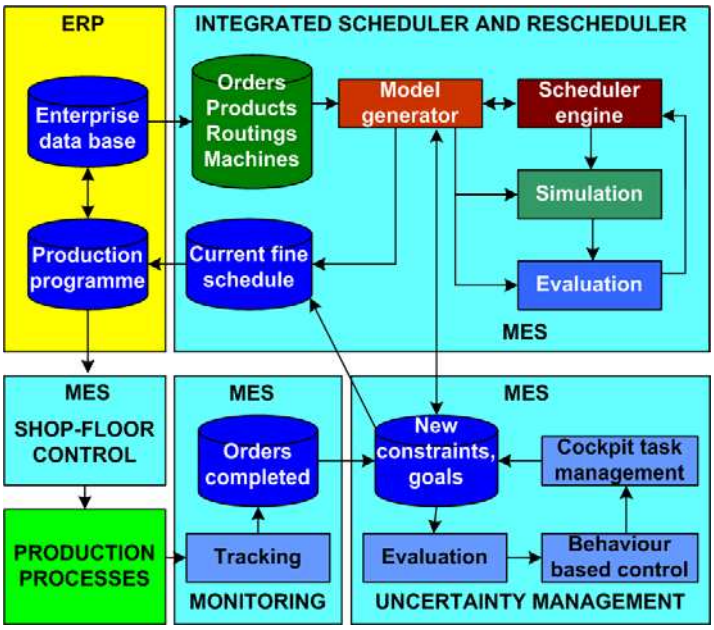


Figure 2. Integrated scheduling and rescheduling system

It is required of rescheduling methods to consider new demands that are added to the predictive scheduling problem. The last released schedule appears as a new input element of the rescheduling system and it is very important to preserve this initial schedule as much as possible to maintain the system stability. For this purpose we defined new qualitative indices (i.e. related to setup and due date) for

supporting the comparison of schedule changes. We consider a great number of special rescheduling constraints. Some examples are as follows: All jobs which are already finished when the rescheduling process starts are not changeable but can affect the other jobs and orders. The manufacturing tasks of jobs running at the starting time of the rescheduling process must not be interrupted and their possible execution route and parallel machines (and other alternatives) can differ from the original possibilities. Finished or running jobs on resources are known therefore they have to be considered in the future. All production orders starting after the rescheduling process can be considered in their original status. To satisfy these constraints the software uses freezing techniques. The main classes of these techniques are as follows: (1) to freeze jobs, (2) to freeze internal orders and (3) to freeze machines. The advanced functionalities of the software help to satisfy the requirements of shop floor control in practise.

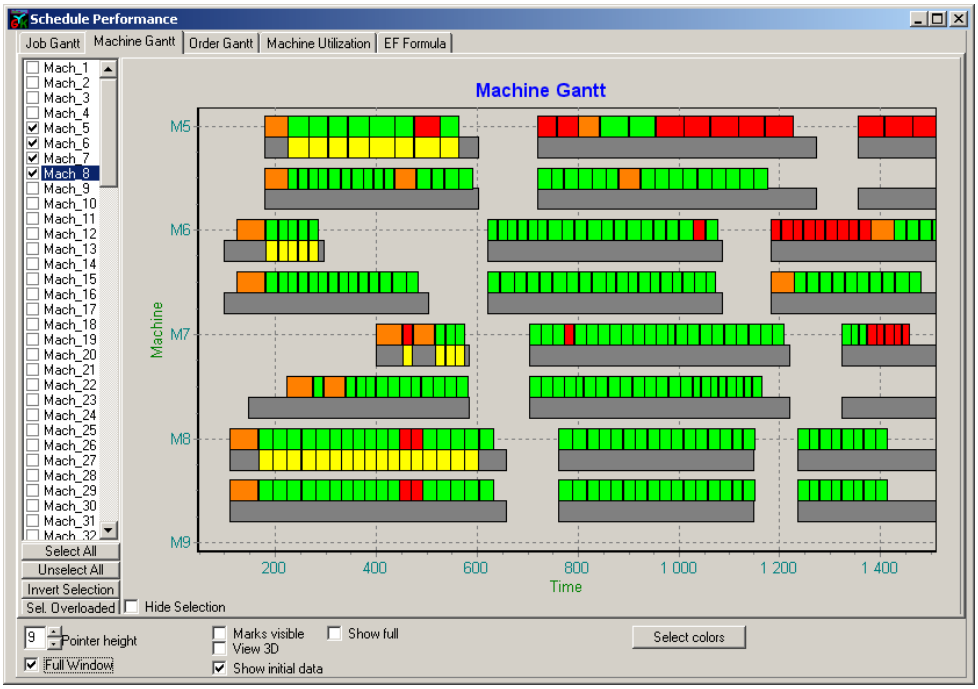


Figure 3. Visualization of the actual and planned status of the jobs

To increase the flexibility and effectiveness of the scheduling process, an advanced software module for supporting the user interactions has been developed. A graphical user interface provides useful charts, diagrams, tables and reports to show aggregate and detailed information of the production fine schedule (i.e. Figure 3). The scheduling process is also scrutinized and checked on the screen, the user can modify at runtime the control priority and parameters of searching

algorithms (Figure 4). In addition, the user can suspend the automatic process and edit the actual schedule by using the available operation tools manually (Figure 5). Similarly to the neighbouring operators, the usage of the manual planning tools can produce only valid and feasible solutions.

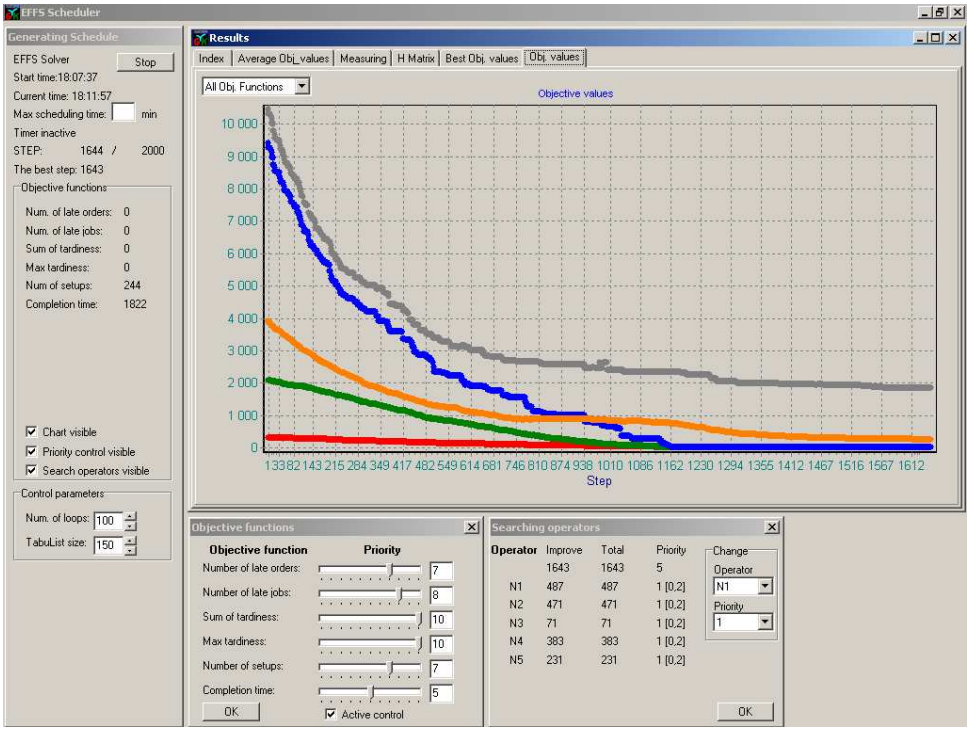


Figure 4. A typical user interface of the scheduling system

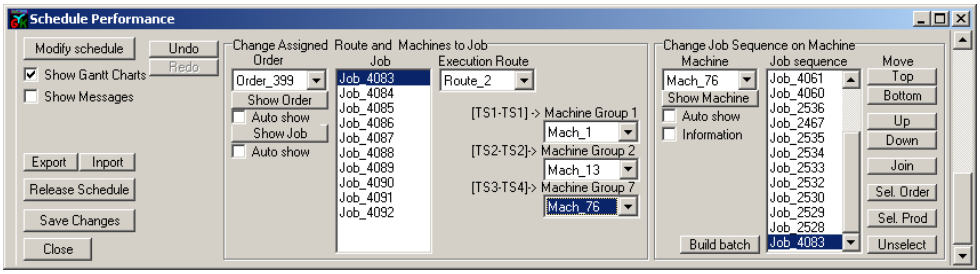


Figure 5. Operation tools for editing schedules manually

The running results produced on sample tasks show that the EFFS model developed and its solution methods are suitable for supporting various production planning and control tasks that fit in the defined category of tasks. Based on the

results of the tests which are executed on small size problem instances, we can say that the method is able to find the optimal solution. The results are validated by comparison with the result of an optimal method based on enumeration technique. The method is able to solve large size problems effectively in a reasonable amount of time [5]. Demo version of the implemented software with built-in problem generator and some input data are available on the Web at the following address: http://ait.iit.uni-miskolc.hu/~kulcsar/EFFS_Sch_Demo.zip.

3. Conclusions

The paper describes the proposition and application of a new method for solving multi-objective scheduling and rescheduling problems. It is based on new interpretation and usage of relational operators for comparing quality of schedules in searching algorithms. After developing the software prototype, the concept is successfully tested on extended flexible flow shop scheduling and rescheduling problems considering multiple objectives and constraints. The results obtained and the problem independent nature of the approach are encouraging for the application of the method in other multi-objective optimization problems. Future research work will be carried out studying the effect of changes in the manufacturing environment and investigating an flexible job shop scheduling model (FJSP) and heuristic solving procedures which can apply the approach proposed.

Acknowledgements

Early period (2004-2007) of the research and development was partially supported by the NODT project entitled “VITAL, Real Time Cooperative Enterprises” (National Office for Development and Technology founded by the Hungarian Government, Grant No.: 2/010/2004, project leader: László Monostori).

REFERENCES

[1] ALLAHVERDI, A., NG, C. T., CHENG, T. C. E., KOVALIOV, M. Y.: *A Survey of Scheduling Problems with Setup Times or Costs*. European Journal of Operational Research, 187, (3), pp. 985-1032, 2008.

[2] AYTUG, H., LAWLEY, M. A., MCKAY, K., MOHAN, S., UZSOY, R.: *Executing Production Schedules in the Face of Uncertainties: A Review and some Future Directions*. European Journal of Operational Research, 161, pp. 86-110, 2005.

[3] BAYKASOĞLU, A., ÖZBAKIR, L., DERELI, T.: *Multiple Dispatching Rule Based Heuristic for Multi-Objective Scheduling of Job Shops Using Tabu Search*. In Proceedings of the 5th International Conference on Managing Innovations in Manufacturing, pp. 396-402, Milwaukee, USA, 2002.

[4] KIS, T, PESCH, E.: *A Review of Exact Solution Methods for the Non-Preemptive Multiprocessor Flowshop Problem*. European Journal of Operational Research, 164, (3), pp. 573-695, 2005.

[5] KULCSÁR, GY., ERDÉLYI, F.: *A New Approach to Solve Multi-Objective Scheduling and Rescheduling Tasks*. International Journal of Computational Intelligence Research, 3, (4), pp. 343-351, 2007.

[6] LINN, R., ZHANG, W.: *Hybrid Flow Shop Scheduling: A Survey*. Computers and Industrial Engineering, 37, (1-2), pp. 57-61, 1999.

[7] LOUKIL, T., TEGHEM, J., TUYTTENS, D.: *Solving Multi-Objective Production Scheduling Problems Using Metaheuristics*. European Journal of Operational Research, 161, pp. 42-61, 2005.

[8] PATERNINA-ARBOLEDA, C. D., MONTOYA-TORRES, J. R., ACERO-DOMINGUEZ, M. J., HERRERA-HERNANDEZ, M. C.: *Scheduling Jobs on a k -stage Flexible Flow-Shop*. Annals of Operations Research, 164, (1), pp. 29-40, 2008.

[9] QUADT, D., KUHN, H.: *A Taxonomy of Flexible Flow Line Scheduling Procedures*. Journal of Operational Research, 178, pp. 686-698, 2007.

[10] SBALZARINI, L. F., MÜLLER, S., KOUMOUTSKOS, P.: *Multiobjective Optimization Using Evolutionary Algorithms*. In Center of Turbulence Research, Proceedings of the Summer Program 2000, pp. 63-74, 2000.

[11] SMITH, K. L., EVERSON, R. M., FIELDSEND, J. E.: *Dominance Measures for Multi-Objective Simulated Annealing*. In Proceedings of Congress on Evolutionary Computation, pp. 23-30, 2004.

[12] TÓTH, T., ERDÉLYI, F., KULCSÁR, GY.: *Decision Supporting of Production Planning and Control by means of Key Production Performance Measuring Indicators*. Seventh International Symposium on Tools and Methods of Competitive Engineering, TMCE 2008, April 21–25, Kusadasi, Turkey, pp. 1201-1215, 2008.

[13] VIEIRA, G., HERMANN, J., LIN, E.: *Rescheduling Manufacturing Systems: A Framework of Strategies, Policies and Methods*. Journal of Scheduling, 6 (1), pp. 35-58, 2003.

[14] WANG, W.: *Flexible Flow Shop Scheduling: Optimum, Heuristics, and Artificial Intelligence Solutions*. Expert Systems, 22, (2), pp. 78-85, 2005.

[15] YAMADA, T.: *Studies on Metaheuristics for Jobshop and Flowshop Scheduling Problems*. PhD Thesis, Kyoto University, Japan, 2003.

[16] ZHU, X., WILHELM, W. E.: *Scheduling and Lot Sizing with Sequence-Dependent Setup: A Literature Review*. IIE Transactions, 38, pp. 987-1007, 2006.


Production Systems and Information Engineering
Volume 5 (2009), pp. 95-108.

AN EXTENDED NEWSVENDOR MODEL FOR SOLVING CAPACITY CONSTRAINT PROBLEMS IN A MULTI-ITEM, MULTI-PERIOD ENVIROMENT

PETER MILEFF
University of Miskolc, Hungary
Department of Information Technology
mileff@iit.uni-miskolc.hu

KAROLY NEHEZ
University of Miskolc, Hungary
Department of Information Engineering
nehez@ait.iit.uni-miskolc.hu

[Received January 2009 and accepted April 2009]

Abstract. In the past few years, the effective management of inventory control problems has become an increasingly critical issue for supplier companies. In this paper, on the basis of the needs a major Hungarian mass production company, we present an extension of an analytical inventory control model considering the condition of global capacity constraint. In a previous paper [6] we elaborated a model regarding the one customer - one supplier relation. Our aim is to determine an optimal holding-production policy of the supplier, which makes a cost-optimum stockpiling policy possible for an arbitrary long production time. We intend to show that, on the basis of our former results, the global capacity constraint satisfying policy can be determined with a new heuristic method.

Keywords: stockpiling policy, extended newsvendor model, global capacity constraint

1. Introduction

In the past 15 years, the business environment of companies in the field of mass production has changed. The demand for mass products has remained high but numerous new requirements have appeared on the market. Changes in the business environment influence engineering and logistic relations between companies and suppliers. The former, simple buying-selling (so-called 'cool') relation has become much 'warmer'. This means that cooperative and collaborative methods and activities have become the main objectives in SCM development. Relations between marketing organizations, end-product manufacturers and supplier companies can be very complicated and diverse in practice. This motivates a wide

examination of the available models and further investigation of effective decision supporting and planning methods.
The professional literature includes a wide variety of stockpiling models [6]. Later we will deal with one of the best known stochastic methods, the so-called 'newsvendor model'. The model is certainly among the most important models in the field of operations management. It is applied in a wide variety of stockpiling problems. In this paper we will examine an extended newsvendor model [7] on the basis of former results in a multi-product and capacity constraint case.
The properties of our extended newsvendor model make it possible to solve the inventory control problem of an arbitrary length production horizon can be solved analytically, which opens up new opportunities to model multi-product capacity constraint problems. Capacity constraint problems appear in almost all larger or smaller manufacturing companies. These firms often produce several products meeting customer demands. In case of dynamically and stochastically changing demands, capacity problems often appear. The question is always the same: how much should be produced? Of course the question is very simple, but the solution always belongs to type NP-hard.
The model conditions are identical to the conditions outlined in publications [5,6]. The larger part of the models solves the problem applying the dynamic programming or some kind of searching method (soft-computing). Due to the large searching space, these solutions require extremely long computing times in case of many products and long production time horizon. In our research we investigated capacity constraint problems ranging from one-product one-period to multi-products and multi-periods. This paper aims to present only some of these methods.

1.1 Related studies

Modelling and solving inventory control problems demand for efficiently has been existed since the establishment of the first industrial companies, factories and enterprises. The first successful publications appeared at the beginning of the 1950's. Since then a great number of papers have been published on stockpiling, which proves that the subject is up-to-date. The most important events related to the evolution of inventory control models are fully summarized in the paper by Hans-Joachim Girlich and Attila Chikán [1999]. The main stream research results are concerned with one-product, one-period deterministic models. These models aim to give an optimum policy in an analytical way in accordance with the objective function of the modelled reality. Multi-period deterministic and stochastic models applying multi-products were developed only in later years.

Another way of carrying out stockpiling policies is game theory approaches. Game theory provides effective methods for modelling the 'warming-up' process of the

supplier – end manufacturer and customer – vendor relations, which tend to be ever closer nowadays as well as for modelling their cooperation. Next the results of the past nearly 50 years are surveyed, an outstanding example being John von Neumann and Oskar Morgenster's famous book, the "Theory of Games and Economic Behavior" [16], which gave a new direction to the approach of inventory problems.

In the late 1950s, the problem of 'Optimal Inventory Policy' was analyzed by two important economists: Arrow and Marschak [13]. Karlin solved this problem with a dynamic programming method (The Structure of Dynamic Programing Models) [14]. Thirty-six years later, Alistair Milne [15] emphasized that one of the best papers in the area of production decisions and inventory analysis was the study by Arrow, Karlin and Scarf entitled 'Studies in the Mathematical Theory of Inventory and Production' [11]. Among the deterministic models, the Wagner-Within method minimizing the total cost plays an important role. It determines the optimal inventory level with O(n logn) calculation time for an n length finite time horizon.

The paper by Dvoretzky, Kiefer and Wolfowitz [17] examined the (S,s) type policy in the case of a fixed time interval and penalty cost. Nowadays the analysis of inventory-holding problems has become an important part of the management of supply chains. Many excellent publications have appeared concerning this topic [9,10,12], which apply the deterministic demand model [5].

Nowadays, regarding supply chain problems, the most prominent results are linked with the name of G.P. Cachon [2,3]. Stockpiling plays an important role in the management of supply chains. With the rapid evolution of information technology, ERP (Enterprise Resource Planning) and SCM (Supply Chain Management) application systems are gaining significance. Dynamic systems with many products are manageable with operations research models or constraint programming methods. However, solutions based on analytical results and heuristics are decisive in 'what if' type investigations and in the case of quick decisions.

1.2 An Extended Newsvendor Model

The classic newsvendor model cannot be applied properly to solve the tasks of customized mass production. The reason for this is high setup costs that cannot tackle multi-period problems where customer demand can vary stochastically. During our research we developed a new inventory control method, which gives the optimal solution for the problem in an analytic way, and ensures efficient stockpiling for the supplier.
Summarizing the main features of the model the objective function can be formulated as follows:

$$K_{123\ldots n}(q) = c_f + c_v[q - I] + hE\left[q - D_1\right]^+ + hE\left[q - D_1 - D_2\right]^+ + \ldots + \\ + hE\left[q - D_1 - D_2 - \ldots - D_n\right]^+ + pE\left[D_1 - q\right]^+ + pE\left[(D_1 + D_2) - q\right]^+ + \ldots + \\ + pE\left[(D_1 + D_2 + \ldots + D_{n-1}) - q\right]^+ + pE\left[D_n + \left[\ldots + \left[D_2 + \left[D_1 - q\right]^-\right]^-\right]^-\right]^+, \tag{1.1}$$

where the individual parameters are the following:

c_f – fixed cost. This cost always exists when the production of a series is started. [Ft / production]

c_v – variable cost. This cost type expresses the production cost of one product. [Ft / product]

p – penalty cost (or back order cost). If there is less raw material in the inventory than needed to satisfy the demands, this is the penalty cost of the unsatisfied orders. [Ft / product]

h – inventory and stock holding cost. [Ft / product]

D – this means the demand by the receiver for the product, which is an optional probability variable. [number / period]

$E[D]$ – expected value of the D stochastic variable.

q – product quantity in the inventory. The decision of the inventory control policy concerns the product quantity in the inventory after the product decision. This parameter includes the initial inventory as well. If nothing is produced, then this quantity is equal to the initial quantity, i.e. concerning the existing inventory.

I – initial inventory level. We assume that the supplier possesses I products in the inventory at the beginning of the demand of the delivery period.

n – number of periods

The new method is robust and adequately elegant (detailed in papers [6][7]), because the solution is independent from the type of the distributed function:

$$F_{123..n}(q^*) = \frac{p - c_v - hF_1(q^*) - hF_{12}(q^*) - hF_{123}(q^*) - \ldots - hF_{123..n-1}(q^*)}{h + p}, \tag{1.2}$$

where $F()$ represents the joint distribution function in compliance with the number of periods drawn together. q^*, which satisfies the equation, expresses how many finished products should be in the inventory at the time when customer demand appears with regard to n periods. Naturally, the critical inventory level, which was first mentioned by Herbert Scarf [1] for one-period production, can be applied in the case of joint production for n numbers of production cycles, as well. However, the present paper does not deal with this.

2. One Product, Multi-Period Model

When applying the global capacity constraint for the multi-period extended newsvendor model we take the characteristic features of the model and the period-based policy into consideration. Accordingly, two types of optimization methods can be distinguished: *service-level-based policy* and *cost-based policy*. Naturally, these policies are in contrast with each other. Only one of them can be considered in the stockpiling policy.

2.1. Cost-Based Policy

This approach models the type of supplier that is directly in connection with the market. In this policy, the main objective is to minimize the costs. It should be decided how many back-orders can be placed in the specified period of time. So the penalty cost is determined by the supplier itself [7]. Since in case of cost-based policy keeping the service level is not the main objective, the solution for the optimization of production costs can be *reducing the number of setups* or *taking the risk of penalty*.

The reduction of the number of jointly produced periods is not definitely the best solution and does not result in the minimization of costs. It is possible that paying a penalty cost is a cheaper way for the supplier.

Theorem 1: if the sum of the penalty cost appearing at producing the quantity according to the capacity constraint and the holding cost of quantity of the truncated period storing from the beginning of the time horizon, is lower than the cost of preparing a new setup, then taking the risk of the penalty is the proper policy.

We prove this theorem as follows. We denote the cost value of the back-orders by variable b. The following equation helps to decide what policy should be applied.

$$\text{if } \begin{cases} b + \sum_{i=1}^{a} h \cdot (C - q_a) \le c_f, \textit{ then take the risk}, \\ b + \sum_{i=1}^{a} h \cdot (C - q_a) > c_f, \textit{ then reduce the periods}. \end{cases}, \tag{2.1}$$

where h is a cumulative holding cost per product and $q_a < C$ is the quantity in compliance with the number of jointly produced periods. Variable a means the period number, q_a value is even smaller than capacity constraint C. Then

$$h \cdot a \cdot (C - q_a) = \sum_{i=1}^{a} h \cdot (C - q_a) \tag{2.2}$$

represents the holding cost, which appears as the difference between the quantity of capacity constraint and quantity of the reduced jointly produced periods. The formula means: if the value of $b + h \cdot a \cdot (C - q_a)$ is lower than a new setup cost (c_f), the cost will be minimized, if the supplier chooses to pay the penalty and produces the quantity of the capacity constraint. Otherwise reducing the number of the jointly produced periods is a good policy. Figure 1 shows the method of cost based policy.

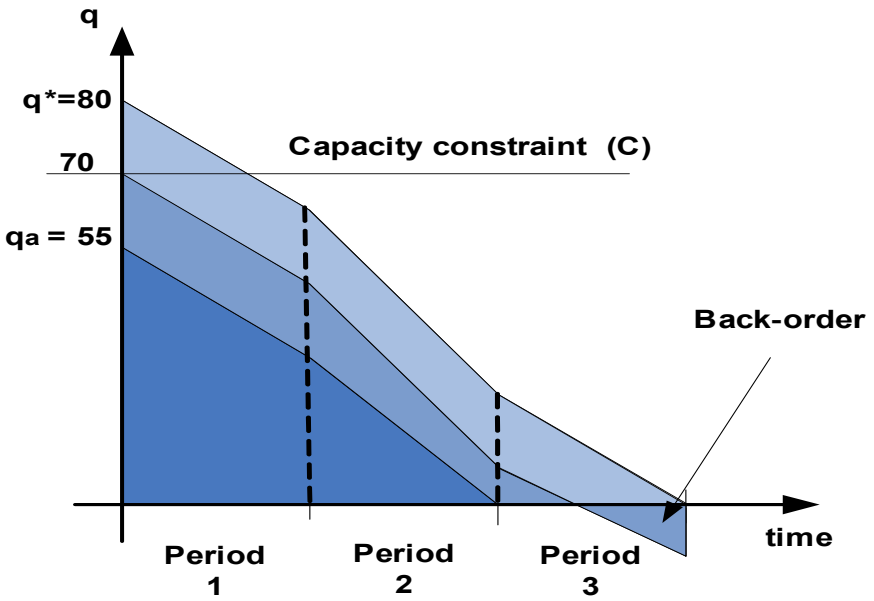


Figure 1. Applying capacity constraint in case of cost base policy

2.2. Service Level Based Policy

The main objective when choosing this policy is to ensure the predetermined *Service Level*. This level is determined in compliance with the objectives of the company. Applying capacity constraint means that the number of unsatisfied orders should be lower than the predetermined service level. Disregarding this important rule gets the relationship between the customer and the supplie at riskr.

The reduction of jointly produced periods is the solution for this problem. If the optimal quantity calculated with the extended newsvendor model exceeds the value of capacity constraint, then the predetermined service level can only be maintained if we reduce the number of jointly produced periods until the quantity satisfies the capacity condition according to the reduced period. This solution is justified by the unit cost variation curve which is further detailed in paper [7].

3. The Multi-Product, Multi-Period Model

Concerning this model, to solution capacity of constraint problems is most complicated. As a rule the ABC method is widely offered to solve the problem. On the basis of the Pareto diagram about the 'significance' distribution of the elements of a product set [4], several conclusions can be drawn. But the method does not give a proper answer to the questions arising while calculating the optimal stockpiling quantities.

Next we will present a new heuristic method to solve multi-product, multi-period and service-level-based capacity constraint optimization problems. We assume a global capacity constraint, which means that different products share one common production capacity and that the decisions of the inventory control policy are made for a long period of time.

We prefer in the solution presented the Service-Level-based policy. The objective is to determine the reduced number of jointly produced periods per product in a way that the sum of the total quantities should satisfy the capacity constraint condition.

The main idea behind this heuristic solution is the specific property of the unit cost of the products. Figure 2 shows the changes in the unit cost of a product against jointly produced periods.

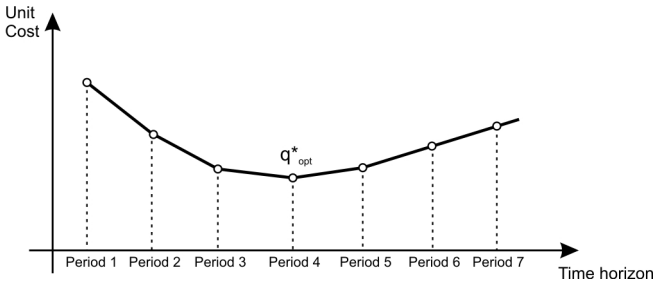


Figure 2. Unit cost changes in case of seven-period length time

Each product has a similar unit curve [7]. If the sum of the quantities of n number of products is larger than the value of capacity constraint then the solution should be changed. If

$$\sum_{i=1}^{n} u^i \cdot (q_{opt_j}^{i*} - I^i) \le C \text{, the solution is optimal.} \qquad (3.1)$$

In the equation i (i=1,…,n) means the number of products, $q_{opt_j}^{i*}$ is the optimal quantity of the product i: opt_j means the number of jointly produced periods and u^i represents the capacity usage of the product. I^i denotes the initial inventory of

product *i*. We should determine the number of setups to satisfy the minimal cost conditions.
First we should start examining the unit cost curve. The following figure shows the changes in the unit cost for 4 periods, as a modification of Figure 2.

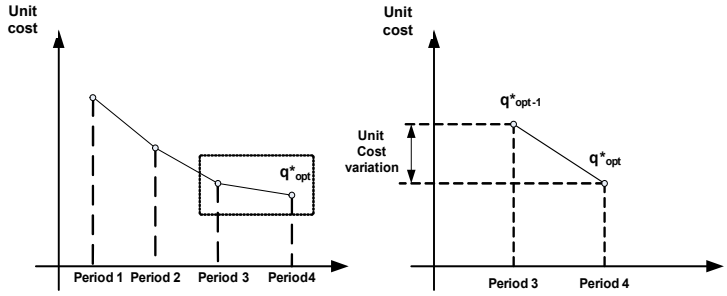


Figure 3. Increase in unit cost against jointly produced weeks

In Figure 3 it is easy to see, if the optimal number (4) of jointly produced periods is reduced to three periods, the value of the unit cost is bound to increase. This observation suggests the following:
Theorem 2: the capacity constraint can be regarded as the optimum solution when it is due to the reduction of jointly produced periods; there is a sum of minimum increases in the sum of the unit cost.

FKV_i denotes the sum of unit cost changes for product *i*. Then:

$$\sum_{i=1}^{n} FKV_i \longrightarrow \min . \tag{3.2}$$

Theorem 2 helps to find the optimal solution, but a searching method is necessary, with which we can calculate the sum of unit cost changes fast in a multi-product, multi-period environment. In the following we present a new and suitable algorithm.

3.1 Algorithm and Other Parts of the Method

The basic idea behind our algorithm is the existence of the optimal solution per product without the capacity constraint condition. The objective of the method is to move the searching space along the minimal unit cost changes, because as we have mentioned before, the optimal solution means the minimal sum of per-unit variation costs. The algorithm can be divided into three main parts: (1) checking the capacity constraint condition (2) selecting optimum modifications possible and (3) choosing a combination to obtain a better solution.

While searching for the proper solution, these steps are continually repeated until the optimal supplier policy can be found in compliance with the capacity constraint conditions.

3.1.1 Capacity constraint condition test

The first step of the method is to determine the optimal number of jointly produced periods based on the introduced unit cost model [8]. This operation is performed only once during the running at the beginning. After that it should be investigated if there are any products the production of which can be 'shifted'. This can be achieved by comparing the quantities in the inventory and the optimal quantities according to the jointly produced periods. Regarding the first period:

$$c_v\left(q_j^{i*} - I_1^i\right) \leq 0\,,\ j = 1,2,...,\ m\,,\ i = 1,2,...,\ n \tag{3.3}$$

If a product can be found for which this equation is true during the calculation of the unit costs, then its production can be shifted forward along the time. After this, these products do not take part in the further steps.
The next step is the evaluation of the following capacity constraint condition.

$$\sum_{j=1}^{n} u^i q^i_{opt_j - L_i} - I^i \leq C\ . \tag{3.4}$$

If the condition is satisfied, then we have the optimum solution. The equation has a new element L_i, which modifies the optimal number of jointly produced periods. L_i represents the solution vector and means the reduced jointly produced periods of the products in the iteration steps of the algorithm. At the beginning of the iteration, this is a zero vector. If the equation is not fulfilled the next step follows.

3.1.2 Selecting the optimum modifications possible

If the solution in the first step or in a previous iteration does not satisfy the capacity constraint, then a modification of the solution is required. In the second step of the algorithm we will choose the products which can be suitable in determining the optimal solution. Choosing the optimum modification possible always means the product which has a minimum unit cost variation. To determine this product the following steps must be taken: the formerly calculated optimal number of jointly produced periods is reduced virtually by one period considering the current modifications (L_i). For product i it means: $opt_j - (L_i + 1)$.

Before and after the reduction, we can calculate both the unit cost and changes in the unit cost variation.

We will choose only one product which has the minimum unit cost variation in this iteration step. If we have chosen product *i*, then we will increase the *i.th* element of the solution vector: $L_i = L_i + 1$. The following figure shows this reduction method.

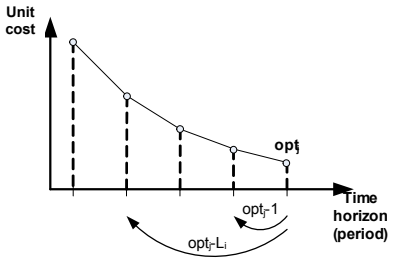


Figure 4. Reduction of the jointly produced periods against the solution vector

The product with the maximum unit cost variation will also be chosen if the combination was performed in the previous iteration. This choice constitutes the basis for the last step of the algorithm, which forbids infinite iteration loops (detailed in step 3).

3.1.3 Selecting a combination for a better solution

Selecting the minimal setup cost is not enough to find the best solution. There can be cases, when the sum of setup cost variation of two products can be substituted for per-setup cost variation of another product in order to achieve a better solution. Figure 5 presents changes in the setup cost of three products and the possibility of substitution.

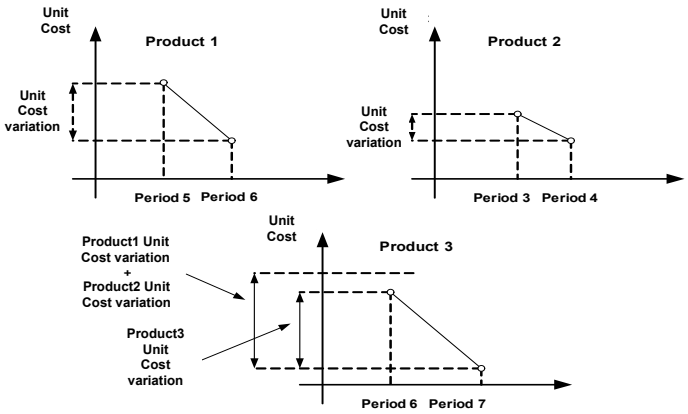


Figure 5. Comparison and substitution of setup cost increases

Explanation of Figure 5: because the originally computed solution does not meet the capacity constraint condition, the number of jointly produced periods is reduced by this algorithm. According to the first step of the algorithm, the product with the minimal variation value of unit cost will be chosen. We will choose the first product. Let us suppose that the solution after the reduction of jointly produced periods from six to five still does not meet the capacity constraint. In case of one product, the substitution phase cannot be explained, so the algorithm runs on. In the second step we will choose another product, with the minimal variation value of unit cost, which will be the second product now.

It is not sure that the reduction of jointly produced periods of the two chosen products is the optimal solution. Therefore we should examine if the sum of the increase of unit cost variations, resulting from the reduction of the jointly produced periods of the two chosen products, can be substituted for a smaller variation of unit cost. Figure 5 shows that the value of unit cost of product three is lower than the variations sum of product one and two.

This means that the variations sums regarding products one and two can be substituted by a reduction of jointly produced periods at product three. Let us examine what will happen if there are four products. In the next figure, the unit cost variations of product three and four can be seen. The situation shows one period decrease of its jointly produced periods.

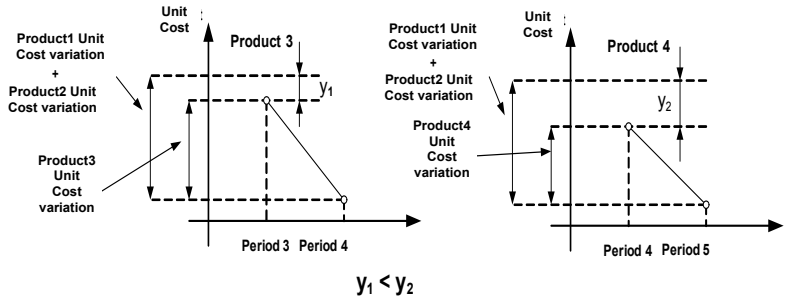


Figure 6. Substitution phase in case of more than three products

In case of more than three products, the question arises: which product should be chosen for substitution. In Figure 6 we can see that both variations of unit costs for product three and four are smaller than the variations sum for the first two products.

In this case the product should be chosen, where the variation is the farthest from the variations sum. The main reason for this is following: if the result of the substitution does not meet the capacity constraint, the algorithm in the next step chooses another product, with the minimal unit cost variation. In this example, the first product satisfies the condition. Let us suppose that this is the optimum

solution. If we choose product four, then the sum of unit cost variations is certainly smaller if we choose product three for substitution.
During the substitution process we use the product with maximum unit cost variation value found in the second step. This value and product constitute the basis of reference in the investigation of the unit cost variations. We will examine as reference the possibility of merging according to this value because it cannot be the chosen product.
If the substitution is carried out successfully it is necessary to prepare the next iteration. The first step is to modify the solution vector. The value in the vector belonging to the selected product should be set to zero. This ensures that the algorithm can move the search space along the changes in the minimal unit cost.
After this process the next iteration comes until the solution meets the capacity constraint condition.
Calculations in practice show clearly that to find the optimal solution we do not need a lot of iteration steps. In case of a product, the optimal number of jointly produced periods is about 7-8 periods. The analytic solution for the extended newsvendor model ensures high-performance calculation in an optional multi-product, multi-period environment for a long period of time.

4. Conclusion

In this paper we extended the previously elaborated and modified newsvendor model [6][7] with the condition of global capacity constraint. Based on the periodic feature of the model, two problem groups were distinguished and presented. For the most complex, multi-period, multi-product case a new heuristic method was elaborated. This model enables the determination of a cost-optimal stockpiling policy applying capacity constraint in case of an arbitrary product number and an arbitrary length of production time. Because of the specific approach of the new model, it can be used effectively in practice compared with other models.

Acknowledgements

The research and development summarized in this paper have been carried out by the Production Information Engineering and Research Team (PIERT) established at the Department of Information Engineering. The research is supported by the Hungarian Academy of Sciences and the Hungarian Government with the NKFP VITAL Grant. The financial support of the research by the aforementioned sources is gratefully acknowledged. Special thanks to Ferenc Erdélyi for his valuable comments and review work.

REFERENCES

[1] HAYRIYE, A., JIM, D., FOLEY, R. D., JOE, W.: *Newsvendor Notes*, ISyE 3232 Stochastic Manufacturing & Service Systems, 2004.

[2] CACHON, G. P.: *Competitive Supply Chain Inventory Management, Quantitative Models for Supply Chain Management*, International Series in Operations Research & Management Science, 17), Chapter 5, 2003.

[3] CACHON, G. P.: *Supply Chain Coordination with Contracts*. In de Kok, A. G., Graves, S. C. (eds): Supply Chain Management: Design, Coordination and Cooperation. Handbooks in Op. Res. and Man. Sci., 11, Elsevier, 2003, pp. 229-339.

[4] TAYLOR, A. D.: *Supply Chains A Managers Guide*, Addison Wesley, 2003.

[5] HANS-JOACHIM, G., CHIKÁN, A.: *The Origins of Dynamic Inventory Modelling under Uncertainty*, International Journal of Production Economics Volume 71, Issues 1-3, 1999, pp. 25-38.

[6] MILEFF, P.: *Kiterjesztett újságárus modell alkalmazása az igény szerinti tömeggyártás készletgazdálkodási problémáiban*, PhD thesis at Hatvany József Informatikai Tudományok Doktori Iskola, 2008.

[7] MILEFF, P., NEHEZ, K.: *An Extended Newsvendor Model for Customized Mass Production*, AOM - Advanced Modelling and Optimization. Electronic International Journal, Volume 8, Number 2, 2006, pp 169-186.

[8] MILEFF, P., NEHEZ, K.: *A new inventory control method for supply chain management*, UMTIK-2006, 12th International Conference on Machine Design and Production, Istanbul – Turkey, 2006, pp. 393-409.

[9] BRAHIMI, N., DAUZERE-PERES, S., NAJID, N. M., NORDLI, A.: *Single Item Lot Sizing Problems*, European Journal of Operational Research, 168, 2006, pp. 1-16.

[10] LEE, C. C., CHU, W. H. J.: *Who Should Control Inventory in a Supply Chain?*, European Journal of Operational Research, 164, 2005, pp. 158-172.

[11] ARROW, K.J., KARLIN, S., SCARF, H., *Studies in the Mathematical Theory of Inventory and Production*, Stanford University Press,1958.

[12] JULIEN, B., DAVID, S.: *The Logic of Logistics: Theory, Algorithms, and Applications for Logistics Management*, Springer PLACE of publication, Chapter 8-9, 1997.

[13] ARROW, K.J., HARRIS, T., MARSCHAK, J.: *Optimal inventory policy*, Econometrica 19: 250 – 272, 1951.

[14] KARLIN, S.: *The structure of dynamic programing models*, Naval Research Logistics Quarterly 2: 285 – 294, 1955.

[15] MILNE, A.: *The mathematical theory of inventory and production: The Stanford Studies after 36 years*, In Workshop, August 1994, Lake Balaton. ISIR, Budapest, 1996, 59 - 77.

[16] VON NEUMANN, J. AND MORGENSTERN, O.: *Theory of Games and Economic Behavior*, Princeton University Press, 1944.

[17] DVORETZKY, A., KIEFER, J., WOLFOWITZ, J.: *On the optimal character of the (s; S) policy in inventory theory*, 1953, Econometrica 21: 586 - 596.


Production Systems and Information Engineering
Volume 5 (2009), pp. 109-138.

OpTol: Spatial Tolerance Analysis Application

KÁROLY NEHÉZ
University of Miskolc, Hungary
Department of Information Engineering
nehez@ait.iit.uni-miskolc.hu

TIBOR TÓTH
University of Miskolc, Hungary
Department of Information Engineering
toth@ait.iit.uni-miskolc.hu

[Received February 2009 and accepted April 2009]

Abstract. The analysis of manufacturing and assembly dimension chains is indispensable for performing up-to-date part manufacturing and assembly. It will both reduce the manufacturing and assembly costs and will result in a well-grounded body of knowledge and improved level of design. The paper deals with mathematical models for calculating spatial dimension chains and introduces the OpTol Tolerance Calculator software. This application is capable of calculating planar (2D) and spatial (3D) dimensional chains by using the classical worst-case and statistical methods, as well as applying the modern six-sigma tolerancing method. OpTol system also contains a CAD module in order to support engineers in analysing their existing assemblies.

Keywords: 3D tolerance analysis, tolerance calculation, dimension chains, direct linearization method

1. Introduction

For performing up-to-date part manufacturing and assembly, the analysis of dimension chains for manufacturing and assembly is essential. Such an analysis will reduce the manufacturing and assembly costs, on the one hand, and will result in a well-grounded body of knowledge and improved level of design, on the other hand. The build-up and analysis of dimension and tolerance chains play an important role in periods of design, production planning and execution of the manufacturing process. The designer provides information by part drawings giving dimensions and tolerances for planning tasks for technology and material processing.

Beyond determining the geometry of parts, the manufacturing dimension and tolerance chains give feasible manufacturing methods and the possible order of

manufacturing processes, as well as the costs of production of the part. The task of assembling dimension and tolerance chains is to determine the relative position of parts needing to be assembled to fulfil the requirements of operation (the function).

2. Fundamentals of dimension and tolerance chains

The dimension and tolerance chain – or simply dimension chain – consists of at least two toleranced dimensions connected together and the resultant dimensions derived from them. The chain used for tolerance calculation is always closed, i.e. comprises the open dimension chain in the dimension chain of the drawing and the resultant dimension. The dimension chain expresses: the chain of dimensions needed to define a part; the relation of a pair of toleranced dimensions; and the operational or assembly location produced by a series of toleranced dimensions. Dimensions occurring in dimension chains are called components**.** The closing or resultant dimension is the term that is worked out last. There can be only one resultant in each tolerance chain.

2.1. Chain types

Dimension chains can be: linear dimension chains, where all of the dimensions are parallel to each other; planar dimension chains, where the dimensions are partially or fully non-parallel but all of them lie in one or more parallel planes; spatial dimension chains, where the dimensions are partially or fully non-parallel and do not lie in one or more parallel planes; and angular dimension chains, where the dimensions are angular and the number of angle legs meet in one corner (see Figure 1).

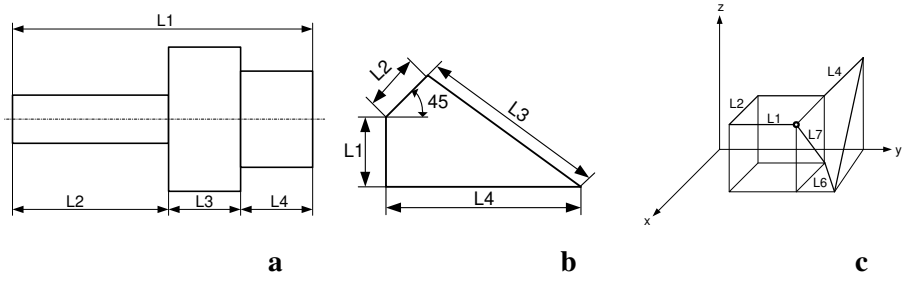


Figure 1. **a**: linear dimension chain, **b**: planar dimension chain, **c**: spatial dimension chain

In different assemblies, several different types of dimension chains can be found and these can connect to each other in different ways. The main characteristic of the serial type of connection is that if one link of the dimension chain changes, then the

basis of the next chain will be changed (see Figure 2a). It follows from this that a serial type of chains has a common basis.

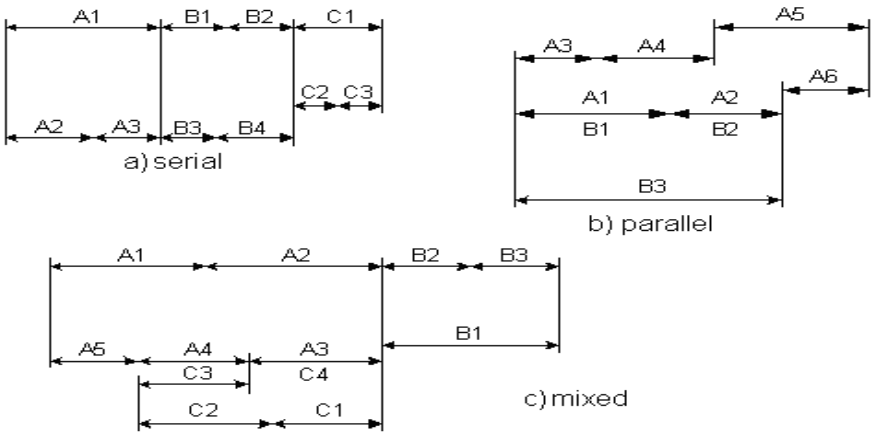


Figure 2. Connection types of dimension chains

2.2. Assembly dimension and tolerance chains

An assembly includes the joining of the connected components, the controlling of their allocation after fitting the corresponding basic surfaces and – if necessary – the correction of the allocation error. An assembly dimension chain is a sequence of dimensions which returns to itself in a determined order. This chain connects the surfaces of the components whose mutual positions are to be determined. The components of the dimension chain are characterized by their nominal values and permissible variations.

2.3. Tolerance analysis and allocation

In tolerance analysis all the component tolerances are known or prescribed and we have to calculate the resulting tolerance. In the case of tolerance allocation, construction requirements determine the assembly tolerance and the unknown component tolerances are to be calculated. We distribute the actual assembly tolerance corresponding to the appropriate components. The design application for tolerance analysis is based on analytical models, which take into account the stack up of tolerances in the assembled components.

3. Engineering calculation methods for assembly tolerances

If the process of part manufacturing is known, the tolerances can be chosen from tables of standard tolerances according to the process elements. In addition to this, the industrial standards often provide useful data for our calculations. We introduce briefly the two current models (see [6]).

3.1. Worst-case model

This model is often called the model of total changeability or calculation of maximum-minimum. The purpose of this method is to determine the assembly tolerance (T_Δ) by means of the summarization of the component tolerances. Each component is assumed to be at its greatest or least dimension; hereby we have the worst assembly limits.

In the case of a *one-dimensional* (linear) dimension chain we have:

$$T_\Delta = \sum_{i=1}^{n-1} T_i \,, \tag{1}$$

For a *multidimensional* (nonlinear) dimension chain:

$$T_\Delta = \sum_{i=1}^{n-1} \left| \frac{\partial f}{\partial X_i} \right| T_i \,, \tag{2}$$

where X_i means the nominal component dimension, $f(X_i)$ is the assembly function describing the resulting dimension of the given assembly and T_i denotes the width of the tolerance zone for the i-th dimension. The partial derivatives represent the sensitivity of the assembly tolerance regarding the changes in the independent component dimensions.

Equation (2) is not obvious at all. First observe that there is a well defined, analytical connection between the nominal component dimensions and the resulting dimension (closing component):

$$X_n = L_\Delta = f(X_1, X_2, ..., X_i ... X_{n-1}) \,. \tag{3}$$

Components $X_1, X_2, ..., X_i ... X_{n-1}$ are made with tolerances $T_1, T_2, ..., T_i ..., T_{n-1}$, so the resulting dimension L_Δ will have tolerance T_Δ:

$$L_\Delta + T_\Delta = f(X_1 + T_1, X_2 + T_2, ..., X_i + T_i ... X_{n-1} + T_{n-1}). \tag{4}$$

The *(n-1)* variables function L_Δ in Eq. (4) is assumed to have an expansion into the Taylor series, so it is differentiable at any time with respect to each independent variable in the neighbourhood of the point $X_1, X_2, ..., X_i ... X_{n-1}$:

$$\begin{aligned} L_\Delta + T_\Delta = f(X_1, X_2, ..., X_i, ..., X_{n-1}) + \frac{\partial f}{\partial X_1} T_1 + \frac{\partial f}{\partial X_2} T_2 + ... + \frac{\partial f}{\partial X_i} T_i + ... + \frac{\partial f}{\partial X_{n-1}} T_{n-1} + \\ + \frac{1}{2!} \frac{\partial^2 f}{\partial X_1^2} T_1^2 + \frac{1}{2!} \frac{\partial^2 f}{\partial X_2^2} T_2^2 + ... + \frac{1}{2!} \frac{\partial^2 f}{\partial X_i^2} T_i^2 + ... + \frac{1}{2!} \frac{\partial^2 f}{\partial X_{n-1}^2} T_{n-1}^2 + ... \end{aligned} \tag{5}$$

In Eq. (5) the members of second, third, etc. order in the Taylor series can be neglected, because the tolerances T_i are small, and their squares and higher powers are smaller. Subtracting both sides of Equation (5) from (3) we obtain:

$$L_\Delta - (L_\Delta + T_\Delta) = -T_\Delta = -\frac{\partial f}{\partial X_1} T_1 + \frac{\partial f}{\partial X_2} T_2 + ... + \frac{\partial f}{\partial X_i} T_i + ... + \frac{\partial f}{\partial X_{n-1}} T_{n-1} = -\sum_{i=1}^{n-1} \frac{\partial f}{\partial X_i} T_i . \tag{6}$$

Since tolerances T_i are the width of the tolerance zone for the component X_i and T_Δ is the width of the resulting tolerance zone, these numbers must be positive by definition. Multiplying both sides of Equation (6) by (-1) we have to use the modulus of partial derivatives because they can take negative numbers. In this way we have the equation:

$$T_\Delta = \sum_{i=1}^{n-1} \left| \frac{\partial f}{\partial X_i} \right| T_i$$

which is the same as Equation (2).

3.2. Statistical tolerance analysis

In this case the stack up of tolerances shows an analogy with random variations. The measured values X_i belonging to the function $y = f(X_i)$ (this function was described above) contain random errors δX_i. These errors have unknown signs and they vary their dimensions in given bounds. The linear addition of the greatest values of the errors δX_i would result in a too-high stack up. It is quite unlikely that the errors have the same signs and that they take their greatest value at the same time. Deviations can compensate for each other in the summation. Due to this observation we calculate the uncertainty factor δy in terms of Gauss summation law of random errors (instead of linear addition):

$$\delta y = \sqrt{\sum_{i=1}^{n-1} \left(\frac{\partial f}{\partial X_i} \delta X_i \right)^2} . \tag{7}$$

For application of the law errors must be independent, and within their bounds the partial derivatives $\frac{\partial f}{\partial X_i}$ can be considered to be constant values. The practical tolerance limit T_Δ assumes that the components of the dimension chain join each other with the value of the greatest probability within their tolerance area. If the extreme tolerance limits meet, the tolerance limit can be exceeded and we have a rejected assembly.

In the discrete processes of the machine industry the errors of measurements follow a typical discrete distribution. This is *binomial distribution* [2]. Adding numerous independent random variables where the variances of the components are negligible compared with the variance of the sum, we always get a variable of *normal distribution* independently of the distribution of the components.

From the viewpoint of the machine industry the most important distribution is the normal or Gaussian distribution. Although it is a continuous distribution, it is suitable for building a mathematical model of the *variation* of measurements (instead of using binomial distribution). The most typical example of normal distribution arises by measuring, in the case of random errors [2]. The general form of the density function of Gaussian normal distribution is the following:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-u)^2}{2\sigma^2} \right), \tag{8}$$

where x denotes expected value (mean of an infinite number of measured data) and σ is the standard deviation. The graph of the density function (8) is shown in Figure 3.

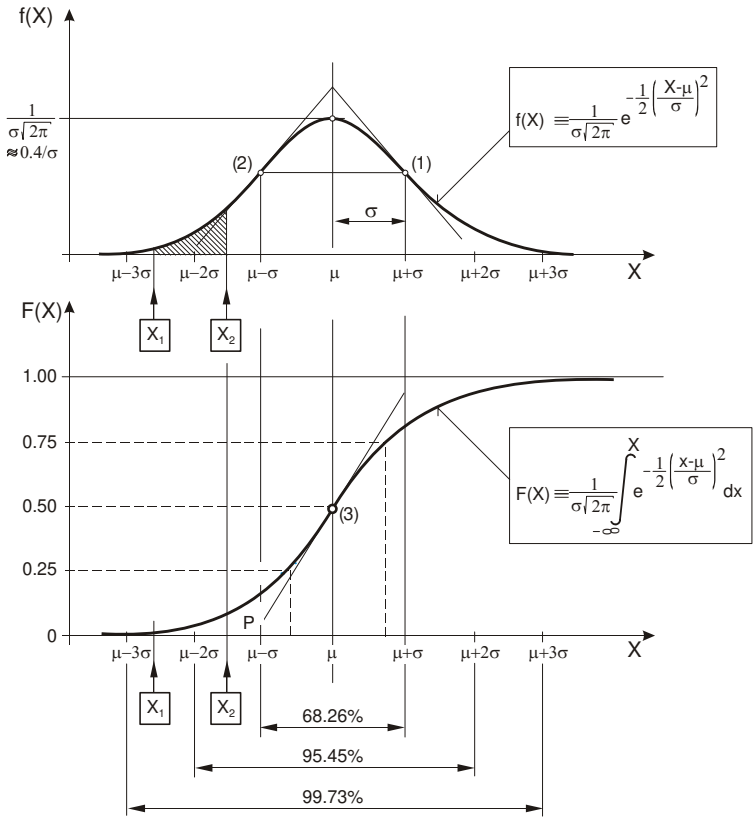


Figure 3. Density function *f(X)* and distribution function *F(X)* in the case of normal distribution. The points denoted by (1), (2) and (3) are inflections [9]

The domain of a normal distributed variable is the real line so an ideal Gauss-curve is situated above the interval [-∞, +∞]. In practice the normal distribution can be considered final and the outer part of the interval [μ-3σ, μ+3σ] is not significant so we can neglect it. It is shown in Figure 3 that 68.26 percent of all possible values of a

normal distribution variable lie in the interval [μ-σ, μ+σ], there is 95.45 percent between μ-2σ and μ+2σ and nearly the whole mass (99.73 percent) is settled in the interval [μ-3σ, μ+3σ]. The latter interval width is usually considered to be the 'technological 100 percent'. The normal distribution is completely determined by its two parameters: the expected value (μ) and the standard deviation (σ). We cannot calculate the exact expected value we use; instead we use the most possible value of the measured data given by the mean of measure sequences achieved in sufficiently large number. Similarly we consider empirical deviation instead of theoretical standard deviation.

Following statistical laws, component tolerances are accumulated in a square-root form. We allow for the lowest probability of the worst-case combinations, assuming that the variations of the components are normally distributed. In general, tolerances are supposed to suit a 6σ deviation of the normal distribution. Tolerance of the closing component in an assembly dimension chain is given by the following formulas.

In the one-dimensional case:

$$T_{\Delta} = \sqrt{\sum_{i=1}^{n-1} T_i^2} \ . \tag{9}$$

In the case of multi-dimensional chains:

$$T_{\Delta} = \sqrt{\sum_{i=1}^{n-1} \left(\frac{\partial f}{\partial X_i} \right)^2 T_i^2} \ . \tag{10}$$

In a more general case when tolerance distribution differs from ±3σ:

$$T_{\Delta} = C_f Z \sqrt{\sum_{i=1}^{n-1} \left(\frac{\partial f}{\partial X_i} \right)^2 \left(\frac{T_i}{Z_i} \right)^2} \ , \tag{11}$$

where Z is the required number of standard deviation according to the described assembly tolerance and Z_i denotes the expected deviations of the component tolerances. The correction factor C_f is often taken into account when circumstances differ from the ideal case. Typical values of C_f are 1.4 and 1.5.

3.3. Bounds of common assembly models

In statistical models we assume manufacturing variants which are normally distributed symmetrically about the centre of tolerance limits. These models do not consider possible asymmetry or deformation. Figure 4 illustrates the occurrence of unexpected rejects if we do not take asymmetry into account.

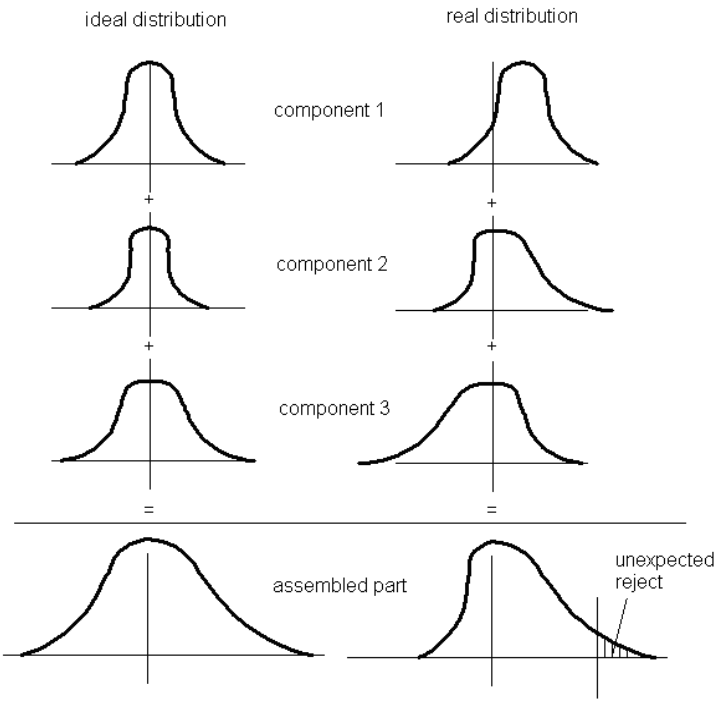


Figure 4. Ideal and real distribution for a three-component assembly

Asymmetric deformation is expressed in shift from the real measurement. It is very dangerous because the variations can stack in the given assembly resulting in an unexpectedly high percentage of rejects. All manufacturing processes show asymmetry, although some processes produce larger variations than others. Asymmetric deformation can come from setup errors, tool wear, etc. Asymmetric deformation occurs in a natural way in some processes, for instance by thermal contraction of the assembly parts cast in die. Deformation in an assembly model is just as critical as the capacity or variance of the process. Furthermore, statistical

approaches or genetic algorithms can be applied in case of non-ideal (real) probability distribution of component tolerances [15, 9].

3.4. Motorola 6 sigma model

The more we refine the process control, the more we have decreasing spread of operations and if the distribution of variations is symmetrical, fewer and fewer rejects will occur [12]. Figure 5 shows that if the lower limit (LL) and upper limit (UL) of the dimension are within the ±6σ limits, then we achieved the so-called 'six sigma quality'. If the UL and LL are set at the ±3σ limits, then we have 0.27% reject assemblies. This number does not seem very high, but it means that for one million products we can reject 2700 of them. Setting the UL and LL at ±4.5σ will yield 3.4 ppm rejects (products per million). In the case of ±6σ this ratio is nearly 100%, with only 2 rejected products per billion.

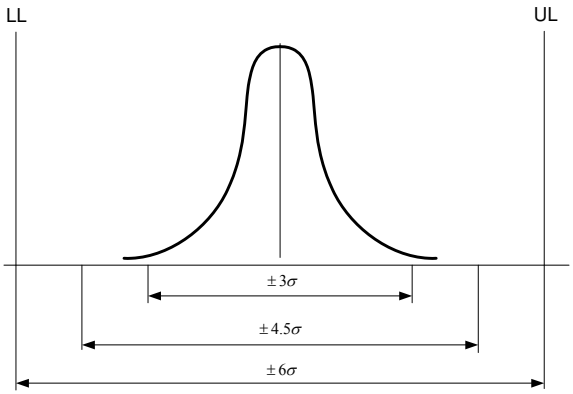


Figure 5. Density function of normal distribution in ±6σ model

It may sound surprising, but 'Six Sigma' is actually the target quality level of today's major manufacturing corporations. It seems easy to achieve this quality level by increasing UL and LL up to the ±6σ limits. But this solution cannot be successful because the UL and LL limits are not arbitrarily chosen; they must meet exacting requirements in the planning and working processes.

3.5. Estimated mean shift

Chase and Greenwood offer a new model for describing assembly tolerance stackup which contains the estimation of the expected asymmetric deformation (see [3]). We call this method 'Estimated Mean Shift Model', since the constructor has to estimate

the deformation of each component of the given assembly unit. This is done in the following way: surrounding symmetrically the centre of the tolerance area we define a zone (see Figure 5) which gives the possible position of some dimension of a typical component sequence.

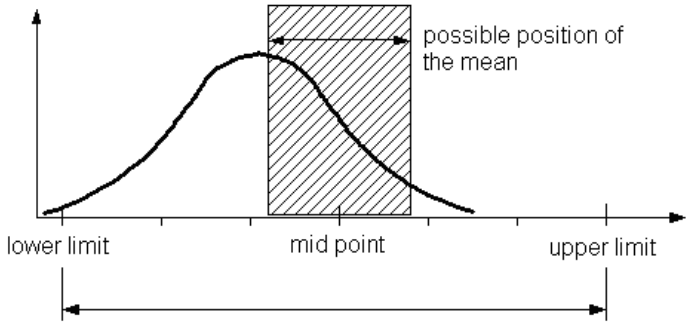


Figure 6. Position of the mean is not exactly known

The centred tolerance zone is given by a proportion of the tolerance area described for the actual component dimension. This number is between 0 and 1. In strictly controlled producing processes it is sufficient to choose a low mean shift factor, e.g. between 0.1 and 0.2. If the process is less known, e.g. in the case of a component supplied by a new business partner, we choose 0.7 or 0.8 in order to allow for some uncertainity.

After estimating the mean shift zone regarding each of the components we calculate the assembly tolerance in terms of the following mathematical model:

$$T_{\Delta} = \sum_{i=1}^{n-1} \left| m_i \frac{\partial f}{\partial X_i} T_i \right| + \sqrt{\sum_{i=1}^{n-1} ((1-m_i)^2 \left(\frac{\partial f}{\partial X_i} \right)^2 T_i^2)}, \tag{12}$$

where m_i denotes the mean shift factor for the i-th component. The assembly tolerance in Equation (2.12) consists of two parts. The first expression is the sum of the mean shifts which are given as the worst limits. The second part of the formula is the sum of the component tolerances calculated in a statistical way. So we obtain the contributions in the closing assembly tolerance according to the mean shift or deformation and component tolerance or deviation, respectively [3, 4].

Choosing every mean shift factor to be zero, Equation (12) can be reduced to the simple statistical model. In addition we obtain the Worst Case Model if all of the mean shifts are chosen to be 1.

We should mention further advantages of the Estimated Mean Shift Model. Mixed application of the factors assures proper flexibility in a given assembly. Some components may correspond to the worst limit while others may vary to a great extent in accordance with the statistical case. Because of a weakly controlled component we are not constrained to apply the worst case model for the whole assembly unit.

3.6. Effect of the mean shift

Figure 7 demonstrates clearly the effect of the mean shift. The values UL and LL were originally set at the ±6σ limits of the distribution. The mean shift of the arising dimension has been shifted 1.5σ to the right, leaving 4.5σ to remain. Since UL is 4.5σ from the mean, it will yield increasing rejects, i.e. 3.4/2 = 1.7 ppm. It is not a large number, but compared to it the ±6σ case (without mean shift the reject products are 2 per billion) we get nearly a factor of 1000 increase!

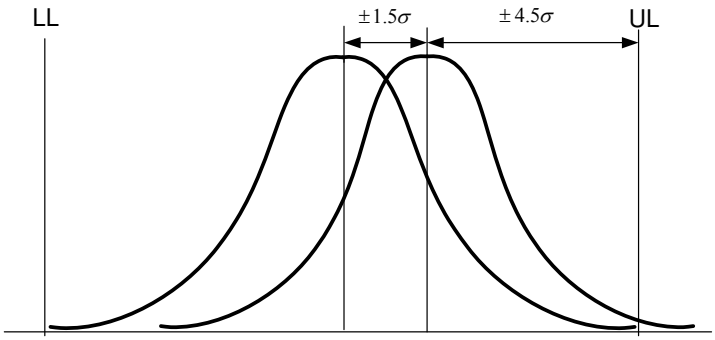


Figure 7. Effect of the mean shift

4. Other tolerance analysis methods

In certain cases other methods are applied in tolerance analysis, especially when the dimensions of the components are not normally distributed. We need to give full distribution as input in order to apply the assembly equation.

The Monte Carlo Simulation and the Method of Moments are useful tools for analysing tolerances if dimensions of the assemblies differ from the normal distribution [5]. The Monte Carlo Simulation generates pseudo-random numbers in order to describe a wide range of distribution curves. Each component receives an input dimension randomly for the assembly equation. After determining the value of the closing assembly variable, it is compared with the described assembly limit. This procedure is repeated again and again and the number of the assemblies proved rejects is divided by the number of the trials to estimate the proportion of the rejected assemblies [7], [8], [13].

The Method of Moments uses the empirical moments of the contributing distributions and the first and second derivatives of the assembly function to find the first four moments of the assembly distribution.

There is an alternative idea requiring a less sophisticated, quicker program, which can be regarded as a mixture of the methods mentioned. This hybrid method applies the Monte Carlo Simulation for generating assembly values in a smaller number. The sample size is usually between 1000 and 5000. The resultant assembly dimensions are used to compute the statistical moments of the assembly distribution and to estimate the percentage of rejected products. With the aid of this trick we avoid the greatest difficulties arising in the Method of Moments since we do not need numerical derivatives and summation of series in order to calculate assembly moments from the component moments. Considering that the sample is in the order of thousand the calculation is extremely simplified compared with the original Monte Carlo Simulation [11].

Constraint networks can also be applied for determining an optimum allocation of tolerances among components of an assembly and at the same time minimizing the total cost of manufacturing [20].

5. Classic methods of solving assembly dimension chains

The problems which can be solved on the basis of the theory of dimension chains are divided into three groups:

- calculation of closing tolerance based on the described tolerances of the components of the dimension chain;
- determination of the component tolerances using the described closing tolerance;
- determination of closing and component tolerances meeting general requirements.

These problems can be interpreted both for the component and the assembly dimension chains. The classical methods for assembly dimension chains are the following: (1) method of total changeability; (2) method of limited changeability; (3) method of selective coupling; (4) method of post fitting; (5) method of adjusting.

The first two methods have been already discussed to some extent previously. In this paper we give more details on total changeability.

5.1. Method of total changeability

In the case of total changeability, assembly can be carried out with randomly selected identical parts, and in this way the closing dimension will always be the prescribed value without adjusting the inserted parts.

If the dimension chain is solved with the method of total changeability, then it is not sufficient to calculate the tolerance values of all the parts, but each part has to be machined within these prescribed tolerance limits. Without this condition it is not possible to consider applying the method of total changeability.

Advantages of this method are:

- assembly is simple and economical because no adjustment or selecting of components is needed,
- the assembly process can be carried out with semi-skilled workers,
- by virtue of total changeability, assembly processes can be carried out in parallel factories,
- the assembly process can be carried out on an assembly line,
- greater ease in managing machining of spare parts: we can assemble each part onto the product without adjustment and fitting.

The main disadvantage of the method is that part machining requires high accuracy. The method of total changeability is the most economical if the dimensions of chain are very accurate, but the number of components is low [14]. It follows from this that the method of total changeability is used in mass production in the case of high precision and a low number of components.

5.2. Method of limited changeability

Increasing the precision of machining tolerances acts upon the production cost.

Therefore increasing the precision requirements would be limited by the costs and assembly precision.

When calculating the tolerances with the method of total changeability, the theoretical starting point is that particular components are machined with limit

dimensions. Components with limit dimensions of opposite direction can be assembled and will meet the precision requirements. During product manufacturing a very small percentage of components are prepared to their limit dimensions. Therefore considering the variations of machined dimensions the part tolerances are extensible and in this way production can be more economical, except that a certain number of products will exceed the prescribed tolerance limits (a higher rejection rate) [14].

Using the method of limited changeability, it is not possible to ensure the resultant dimensions between the prescribed limits. Applying the theory of the probability calculation we can increase the tolerance values of certain components, but we risk that rejected products will pass the tolerance limits. Increasing the tolerance values leads to more economical part production but marginally increases the reject rate.

Taking these factors into account, we can generally say that the method of limited changeability can be applied if the dimension chain consists of several components and a tight tolerance is prescribed only for the closing dimension. As a consequence of choosing tight closing dimension, it is possible to increase the tolerances of the individual dimension chain components, which reduces the machining cost.

In the case of the method of limited changeability, dimension chain solutions work on the principle that dimensional deviations of chain links as well as the summation of these deviation values have a random character, therefore the rules of probability calculation have to be applied. According to these rules, the boundary values of the closing dimension can be calculated with the summation of the regular and random errors of the links [14].

5.3. Indices of process capability

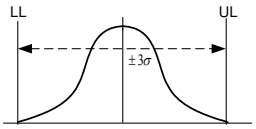
To measure process capability, there are two indices used in modern industrial practice:

C_p process capability index,

C_{pk} C_p adjusted for mean shift.

Process Capability Index:

$$C_p = \frac{UL - LL}{6\sigma}$$



Cp adjusted for k:

$$C_{pk} = C_p(1-k)$$

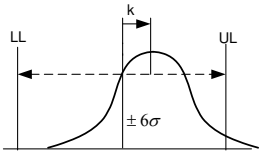


Figure 8. Indices of process capability

The value of capability index C_p is 1.0 only if the limits LL and UL are exactly on the 3σ boundaries of the standard deviation of dimensions. At this time, using the general assumptions of tolerance analysis, all the tolerances correspond exactly to 3σ. If LL and UL correspond to ±6σ, then C_p = 2.0, which matches the quality level 6σ. The previous explanation shows that C_p is a good indication of the quality level, but the mean shift is not taken into consideration.

C_{pk} adjusts the value of C_p, taking the mean shift into consideration. It can be seen from Figure 8 that C_{pk} is *(1-k)* times C_p, where *k=[0..1]*. If the mean shift is 25%, then *k=0.25,* i.e. the distance from the mean of UL and LL, thus the process capacity sinks to 75%.

C_p expresses how close the limits UL and LL are to the process capability ±3σ supposing symmetric distribution; while C_{pk} represents how close the nearest UL and LL limits are supposing non-symmetric distribution.

The model presented here is the 'Six sigma program' developed by Motorola Corporation. This model also takes the qualitative mean shift observed during the mass production of assemblies into account.

Instead of the relationship $T_i=3\sigma_i$, the resultant tolerance can be calculated as:

$$T_i = 3\, Cp_i\, \sigma_i\,, \tag{13}$$

which meets higher quality requirements. Taking the mean shift into account, the previous formula with the substitution of C_{pk} is as follows:

$$T_i = 3\, Cpk_i\, \sigma_i\,, \tag{14}$$

moreover:

$$\sigma_i = \frac{T_i}{3\,Cpk_i}\,. \tag{15}$$

Since *Cpk* is less than *Cp*, the estimated standard deviation σ_i will be greater.

In the case of mass production, the mean of the process can be shifted, e.g. as a consequence of tool wear or thermal expansion. In the long term, the aim of Motorola's Six Sigma principle is to achieve the quality level of 4.5σ. In order to realise it, the quality level of 6σ is to be aimed at in the short term:

$$\text{Short term: } T_i = 3\,Cp_i\,\sigma_i = 3\cdot 2.0\cdot\sigma_i = 6\sigma\,. \tag{16}$$

$$\text{Long term: } T_i = 3\,Cpk_i\,\sigma_i = 3\cdot 2.0\cdot(1-k)\,\sigma_i = 4.5\sigma\,. \tag{17}$$

If the mean shift is less than 0.25 ($k<0.25$), then in the long term, a quality level higher than 4.5σ can also be reached. If $k>0.25$, then the 4.5σ cannot be maintained.

6. Direct linearization method for analysing 3D mechanical assemblies

Kinematic tolerance analysis methods have had an extensive literature in the last five years. Kyung and Sack have successfully applied a nonlinear kinematic tolerance analysis algorithm for planar mechanical systems comprised of higher kinematic pairs [10], and additionally a combination of the direct linearization method and a kinematic error analysis was presented by Wittwer and Chase [19]. Joskowicz and Sacks introduced a new model of kinematic variation, called kinematic tolerance space, that generalizes the configuration space representation of nominal kinematic function [16]. Anselmetti et al. developed a new functional tolerancing method for analysing 3 dimensional variations of mechanical assemblies utilizing a solver implemented in Microsoft Excel [1].

In general, the kinematic constraints for a 3D mechanical assembly can be described by means of closed vector loops. The vector loop is traversed from the starting point to the end point of the mechanism, finally the cyclic translations and rotations will sum to zero. As a last step, the coordinate system of the end point has to be made congruent with the one at the beginning by means of a rotation. The method of vector loops derives from the 2D calculation method, as its spatial extension [5]. In the 3D case the equations of the system are much more complex. At this time it is highly practical to represent the rotation and translation constraints in matrix form. The closed vector chain can be expressed as a product of transformation matrices

representing the constraints. To describe transformation from point *i-1* to point *i* of the mechanism, a combination of three rotation matrices and one translation matrix is necessary in the most general case. The problem can be simplified if we carry out translations always along the local *x*-axis. For 3D rotational transformations, the following matrices can be used:

$$[R_x]=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0' & \cos\varphi_x & -\sin\varphi_x & 0 \\ 0' & \sin\varphi_x & \cos\varphi_x & 0 \\ 0' & 0 & 0 & 1 \end{bmatrix}, [R_y]=\begin{bmatrix} \cos\varphi_y & 0 & \sin\varphi_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\varphi_y & 0 & \cos\varphi_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, [R_z]= \begin{bmatrix} \cos\varphi_z & -\sin\varphi_z & 0 & 0 \\ \sin\varphi_z & \cos\varphi_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{18}$$

For translation it is assumed that the translational vector is always parallel to the local *x*-axis:

$$[T]=\begin{bmatrix} 1 & 0 & 0 & L \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{19}$$

With these matrices, the kinematic constraints of the assembly can be written in form of the following equation:

$$[R_1][T_1][R_2][T_2]...[R_i][T_i]...[R_n][T_n][R_f] = [I], \tag{20}$$

where $[R_i]$ is the product of rotation matrices at joint *i*; $[T_i]$ is the translation matrix at joint *i*; $[R_f]$ is the rotation matrix required to bring the loop to be congruent at the last joint; and I is the identity matrix. Equation (20) is a series of rotations and translations to transform the local coordinates from vector-to-vector to the end point via the joints representing the mechanism. At each joint, the rotation matrix $[R_i]$ is a product of rotation matrices, which aligns the local *x*-axis with the direction of the next vector. Then the transformation matrix $[T_i]$ contains only one translation value *L*

along the local *x*-axis, indicating the length of the current vector. Equation (20) can be decomposed into six independent non-linear equations. Since the nominal dimensions are much greater than their tolerances, the solution can be obtained through linearization. Six equations describe the loop variation in the global *x,y,z* and Θ_x, Θ_y, Θ_z directions, as follows:

$$\delta H_i = \sum_{j=1}^{n} \frac{\partial H_i}{\partial x_j} \delta x_j + \sum_{k=1}^{m} \frac{\partial H_i}{\partial x_k} \delta u_k \qquad \left(i = x, y, z, \Theta_x, \Theta_y, \Theta_z\right), \tag{21}$$

where δx_j are variations in the manufactured dimensions and angles (j = 1…n), δx_k are variations in the dependent assembly variables (k = 1…m) and δH_i is the resultant assembly variation in the corresponding global direction. For closed loops, δH_i is zero and δu_k means the kinematic adjustments bringing about closure. The applicable perturbation method can be found in [14], [6]. If derivation is needed with respect to translational and rotational variables then the actual variable has to be substituted into Equation (20) as L+ΔL in case of translational (L) and φ+Δφ in case of rotational (φ) variables. Due to the small perturbation the equation does not express a closed loop, but a small error vector will produced. The derivates can be expressed with numerical approximation. A detailed derivation of this method can be found in [6]. Based on this method, Equation (21) can be expressed as a linearized matrix form:

$$\{\delta H\} = [M]\{\delta X\} + [A]\{\delta U\} = \{\Theta\}. \tag{22}$$

where {δH} is vector of the clearance variations; {δX} is vector of the variations of the manufactured dimensions; {δU} is vector of the variations of the assembly dimensions; [M] is matrix of the first order partial derivatives of the manufactured variables, [A] is matrix of the first order partial derivatives of the assembly variables; and {Θ} is the zero vector.

Each element of [M] and [A] matrices can be determined with the perturbation method. The structure of both matrices will be as follows:

$$\left[A_i\right] = \left\{ \frac{\partial H_x}{\partial x_i}, \frac{\partial H_y}{\partial x_i}, \frac{\partial H_z}{\partial x_i}, \frac{\partial H_{\Theta x}}{\partial x_i}, \frac{\partial H_{\Theta y}}{\partial x_i}, \frac{\partial H_{\Theta z}}{\partial x_i} \right\}^T \tag{23}$$

where x_i is the i-th assembly variable. The structure of [M] matrix is the same, but u_i will be used instead of x_i. Equation (22) can be solved for [δU]:

$$\{\delta U\} = -[A]^{-1}\,[M]\{\delta X\}. \tag{24}$$

On the score of Equation (24), when [A] is a square matrix, {δU} vector can be determined. This matrix method is highly applicable in computerized implementations.

7. Introducing the Optol 3D tolerance calculation software

Developing a computerized algorithm and its integration into a CAD system is a difficult and complicated task. The research group has proposed a general 'CAD system' independent model. Our starting point is that CAD systems are able to export geometrical data of each design feature with an arbitrarily chosen coordinate system (in Pro/Engineer CAD system datum coordinate systems can be used for this purpose, in CATIA this export is also available). It is obvious that input data for our algorithm will be coordinates of vector end-points. Figure 9 depicts the functional diagram of our OpTol Software. The input data of the OpTol system is a special text-based Loop file that describes an assembly loop. The user is able to analyse an existing Pro/Engineer 2001 assembly by means of the OpTol Pro/Engineer module. In this case, the OpTol module creates the input Loop file for the OpTol System. This module was implemented by using Pro/JLink, which is an extension creator toolkit of the Pro/Engineer software. Additionally, the OpTol system can be used without Pro/Engineer, creating Loop files by using a simple text editor.

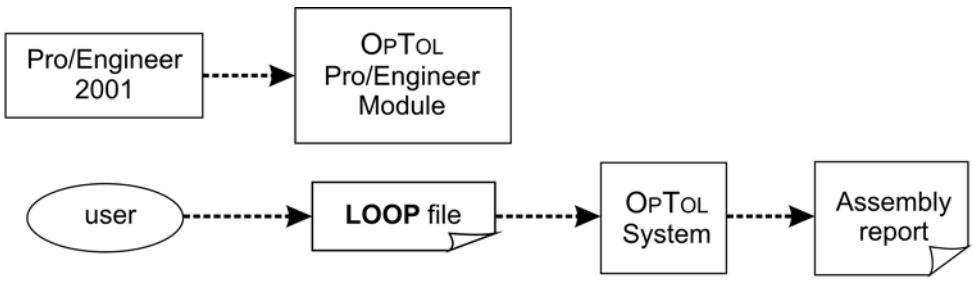


Figure 9. Functional diagram of the OpTol system

The OpTol system generates a detailed html assembly report as a result of tolerance calculations. The OpTol system also supports three dimensional tolerancing

calculation based on worst-case, statistical and six sigma methods. The OpTol system does not yet support geometrical tolerances.

In further versions, our team is planning to extend the functionability of the system by supporting geometrical tolerances and tolerance allocation methods.

Our development strategy is based upon using only open-source software tools and components. All components of the OpTol software were developed in Java utilizing NetBeans IDE and Java Swing API. The OpTol system is basically platform independent but its installer and launcher application only runs under a Windows platform. In the following section a simple 2D example with multiple loops will be presented.

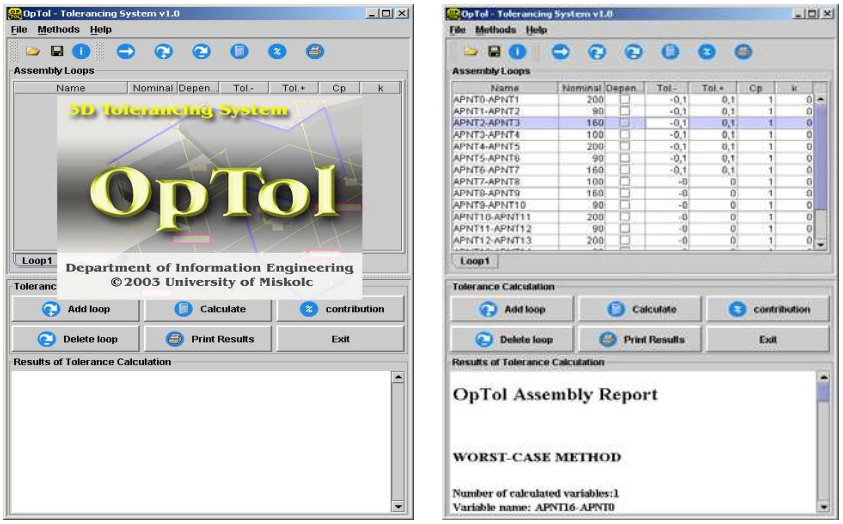
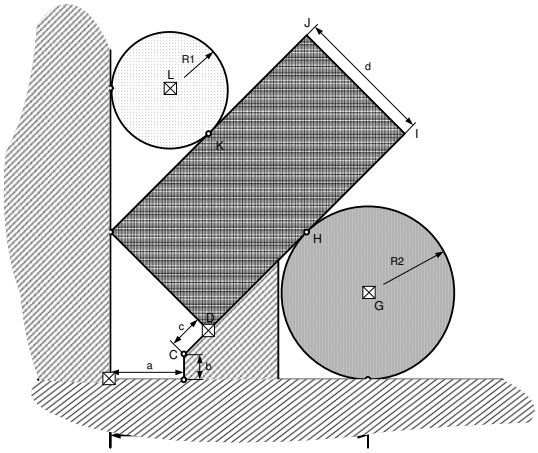


Figure 10. Screenshots of the OpTol Tolerancing System

7.1. 2D Tolerance calculation example (multiple-loops)

We tested the system on a relatively complicated industrial assembly, but publishing the results exceeds the limited length of this paper, at the same time the following 2D example demonstrates the fundamental functions of our application well.

Figure 11 represents a model assembly consisting of four parts: two cylinders, one block and a base. We are looking for the tolerance values of the dimensions X1, X2, X3. The following table shows the x and y coordinates of the points A, B, C… M (Point A is the origin). Assume that for the sake of convenience, all the tolerance values of each dimension (line segments e.g.: AB, BC, DE, … LM) are the same: ± 0.05 mm. A datum reference point must be defined for each part (⊠).



Point name	A	B	C	D	E	F	G	H	K	L	M
x-y coords	0, 0	3, 0	3, 1	4, 2	0, 6	10.5,0	10.5, 3.52	8, 6	4, 10	2.42, 11.8	0,11.8

Figure 11. 2D model example with four parts

The next step of the tolerance calculation is to determine the number of necessary assembly loops. The applicable relation is as follows: L = J – P + 1 where J is the number of joints, and P expresses the number of parts. For our example: J = 6, P = 4, thus L = 3.

7.2. Creating vector loops

A vector loop must fulfil some modelling rules when it passes through parts: [14]

- enter through a joint into a part,
- follow datum path to the datum reference point of the part,
- follow dimensions to another joint,
- leave part.

Figure 11shows this process.

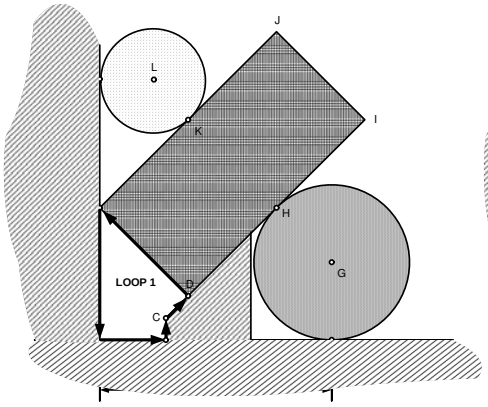


Figure 12a. Loop One

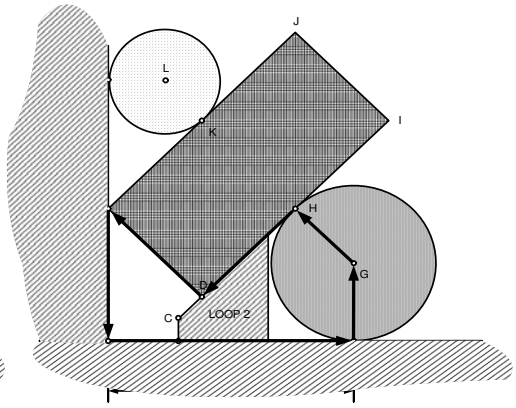


Figure 12b. Loop Two

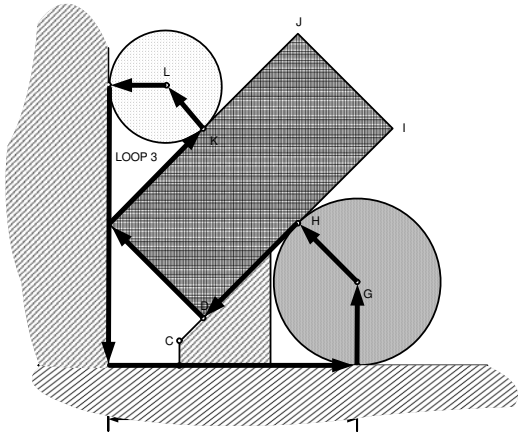


Figure 12c. Loop Three

The loops must go through every part and every joint in the assembly. The following table shows the Loop files, which are importable into the OpTol System (these source files can be found in the folder "*[Install dir]*/Tutorial/"):

Example2D_1.loop			Example2D_2.loop			Example2D_3.loop		
0.0, 0.0, 0.0, A			0.0, 0.0, 0.0, A			0.0, 0.0, 0.0, A		
3.0, 0.0, 0.0, B			10.5, 0.0, 0.0, F			10.5, 0.0, 0.0, F		
3.0, 1.0, 0.0, C			10.5, 3.52, 0.0, G			10.5, 3.52, 0.0, G		
4.0, 2.0, 0.0, D			8.0, 6.0 ,0.0, H			8.0, 6.0, 0.0, H		
0.0, 6.0, 0.0, E			4.0, 2.0, 0.0, D			4.0, 2.0, 0.0, D		
0.0, 0.0, 0.0, A			0.0, 6.0, 0.0, E			0.0, 6.0, 0.0, E		
			0.0, 0.0, 0.0, A			4.0, 10.0,0.0, K		
						2.42, 11.8,0.0, L		
						0.0, 11.8,0.0, M		
						0.0, 0.0, 0.0, A		
Name	Tol. -	Tol. +	Name	Tol.-	Tol. +	Name	Tol. -	Tol. +
A-B	-0.02	0.05	A-F	-0.02	0.02	A-F	-0.02	0.02
B-C	-0.02	0.01	F-G	-0.05	0	F-G	-0.05	0
C-D	-0.1	0.01	G-H	-0.01	0.01	G-H	-0.01	0.01
E-A	-0.03	0.04	H-D	-0.01	0.01	H-D	-0.01	0.01
			D-E	-0.05	0.05	D-E	-0.05	0.05
			E-A	-0.01	0.04	E-K	-0	0
						K-L	-0	0
						L-M	-0	0
						M-A	-0.02	0.05

Figure 13. Three loop files describing the example. The lower part of the table contains the sample tolerance values which have to be set in the application.

Restart OpTol system and import the entire three loops into the application:

- Import loop by pushing *"Ctrl+I"* and select Example2D_1.loop from the *[Install dir]*/Tutorial folder.

- Push the button 

to add a new loop to the table. Select the tab 'Loop2' and import the next loop by pushing *"Ctrl+I"* and select Example2D_2.loop.
- Push the button 

to add a new loop to the table. Select the tab 'Loop3' and import the next loop by pushing *"Ctrl+I"* and select Example2D_3.loop.

The next step is to edit the tolerance values. Figure 13 contains the tolerance values for each segment. In OpTol, you must set any parameter of a dimension only once. Having completed this procedure, check the checkbox 'dependent variable' for the following dimensions: EA, AF, MA (remember: these were X1, X2 and X3). Since we have not indicated the values *Cp* and *k*, we should turn off the six-sigma statistical method. After clicking on the 'Calculate' button, you will see the following result in the 'Results of Tolerance Calculation' tab.

OpTol Assembly Report

WORST-CASE METHOD

Number of calculated variables:3

Variable name: E-A

Tolerance	**+-0.08** [mm]
Calculated Upper limit	6.08 [mm]
Calculated Lower limit	5.92 [mm]

Variable name: A-F

Tolerance	**+-0.021** [mm]
Calculated Upper limit	10.521 [mm]
Calculated Lower limit	10.479 [mm]

Variable name: M-A

Tolerance	**+-0.046** [mm]
Calculated Upper limit	11.846 [mm]
Calculated Lower limit	11.754 [mm]

STATISTICAL (3-sigma) METHOD

Number of calculated variables:3
Variable name: E-A

Tolerance	**+-0.041** [mm]
Calculated Upper limit	6.041 [mm]
Calculated Lower limit	5.959 [mm]
Reject Calculation	
Design Upper limit	6.04 [mm]
Design Lower limit	5.99 [mm]
Predicted Rejects (UL)	1609.937 [ppm]
Predicted Rejects (LL)	230718.717 [ppm]
Predicted Rejects Total	232328.654 [ppm]

Variable name: A-F

Tolerance	**+-0.012** [mm]
Calculated Upper limit	10.512 [mm]
Calculated Lower limit	10.488 [mm]
Reject Calculation	
Design Upper limit	10.52 [mm]
Design Lower limit	10.48 [mm]
Predicted Rejects (UL)	0.498 [ppm]
Predicted Rejects (LL)	0.498 [ppm]
Predicted Rejects Total	0.996 [ppm]

Variable name: M-A

Tolerance	**+-0.028** [mm]
Calculated Upper limit	11.828 [mm]
Calculated Lower limit	11.772 [mm]
Reject Calculation	
Design Upper limit	11.85 [mm]
Design Lower limit	11.78 [mm]
Predicted Rejects (UL)	0.035 [ppm]
Predicted Rejects (LL)	15548.773 [ppm]
Predicted Rejects Total	15548.808 [ppm]

Figure 14. OpTol assembly report of the 2D example tolerance calculation

7.3. Calculating percentual contribution

This procedure is very useful if you want to see how each dimension contributes to a selected variable. If you press 

when the 'loop 1' tab is the selected tab on the pane, you will get the following.

According to Figure 15, BC is the principal contributor of the dimension EA, followed by the dimension CD. If the resultant tolerance is not desirable, we must change the dimension tolerance values. According to the percent contribution results we must reduce the tolerance values of the dimension BC.

Note: this percent contribution calculation is not a trivial task. We have three dimension loops and they affect the tolerance of the dimension EA simultaneously.

(Worst-case) Percent Contribution of variable: E-A

A-B	0.0%
B-C	37.68%
C-D	28.87%
D-E	8.88%
F-G	15.7%
G-H	4.42%
H-D	4.44%
E-K	0.0%
K-L	0.0%
L-M	0.0%

(Statistical) Percent Contribution of variable: E-A

A-B	0.0%
B-C	54.24%
C-D	31.83%
D-E	3.01%
F-G	9.42%
G-H	0.75%
H-D	0.75%
E-K	0.0%
K-L	0.0%
L-M	0.0%

Figure 15. Percent contribution of dimension 'EA'

8. Conclusion

The OpTol 3D tolerancing software and its mathematical models have been presented. The software is utilizing a direct linearization method to solve tolerance calculations up to 3 dimensional cases. The OpTol System can work as a stand alone system or consists of a CAD module in order to support engineers to analyse their existing assemblies. The OpTol system installation package can be downloaded from the website alpha.iit.uni-miskolc.hu/OpTol/setup_trial.exe. The package includes a detailed user's guide with examples and a fully functional trial license.

Continuing work will focus on handling geometrical tolerances and implementing tolerance allocation methods and cost optimizations.

9. Acknowledgements

The research and development summarized in this paper has been carried out by the Production Information Engineering and Research Team (PIERT) established at the Department of Information Engineering at the University of Miskolc and supported by the Hungarian Academy of Sciences. The financial support of the research by the aforementioned source is gratefully acknowledged.

References

[1] ANSELMETTI, B., MEJBRI, H. and MAWUSSI, K.: *Coupling experimental design – digital simulation of junctions for the development of complex tolerance chains*, Computers in Industry, Volume 50, Issue 3, pp. 277-292, 2003.

[2] BÁLINT, L. and GRIBOVSZKI, L.: *Fundamentals of Manufacturing Science and Technology.* Textbooks Publisher, Budapest, 1980. (in Hungarian).

[3] CHASE, K. W. and GREENWOOD, W. H.: *Design Issues in Mechanical Tolerance Analysis,* Manufacturing Review, Volume 11, No 1, pp. 50-59, 1988.

[4] CHASE, K. W.: *Tolerance Analysis of 2-D and 3-D Assemblies,* ADCATS Report No. 99-4, Department of Mechanical Engineering, Brigham Young University, Utah, 1999.

[5] DRAKE, P. J.: *Dimensioning and Tolerancing Handbook,* McGraw-Hill Professional ISBN: 0070181314, 1999.

[6] GAO, J. and CHASE, K. W.: *Generalized 3-D Tolerance Analysis of Mechanical Assemblies with Small Kinematic Adjustments,* IEEE Transactions, Volume 30, Number 4, pp. 367-377, 1998.

[7] GERTH, R. J. and HANCOCK, W. M.: *Computer aided tolerance analysis for improved process control,* Computers & Industrial Engineering, Volume 38, Issue 1, pp. 1-19, 2000.

[8] JOSKOWICZ, L., SACKS, E. and SRINIVASAN, V.: *Kinematic tolerance analysis,* Computer-Aided Design, Volume 29, Issue 2, pp. 147-157, 1997.

[9] KORN, A.G. and KORN, T.M.: *Mathematical Handbook for Scientists and Engineers.* Definitions, Theorems and Formulas for Reference and Review. Second, Enlarged and Revised Edition – McGraw-Hill Book Company, (in Hungarian: Technical Publisher, Budapest, 1975, p.567).

[10] KYUNGA, MIN-HO, SACKS, ELISHA: *Nonlinear kinematic tolerance analysis of planar mechanical systems,* Computer-Aided Design 35, pp. 901–911, 2003.

[11] LIN, CHIH-YOUNG, HUANG, WEI-HSIN, JENG, MING-CHANG, DOONG, JI-LIANG: *Study of an assembly tolerance allocation model based on Monte Carlo simulation,* Journal of Materials Processing Technology 70, pp. 9-16, 1997.

[12] *Motorola Six Sigma model*, http://www.isixsigma.com/me/six_sigma/, 2005.

[13] NIGAM, S. D. and TURNER, J. U.: *Review of statistical approaches to tolerance analysis,* Computer-Aided Design, Volume 27, Issue 1, pp. 6-15, 1995.

[14] ROBISON, R. H.: *A Practical Method for Three-Dimensional Tolerance Analysis Using a Solid Modeller*, M.S. Thesis, Mechanical Engineering Department, Brigham Young University, 1989.

[15] SHAN, A., ROTH, R. N. and WILSON, R. J.: *Genetic algorithms in statistical tolerancing,* Mathematical and Computer Modelling, Volume 38, Issues 11-13, pp. 1427-1436, 2003.

[16] SKOWRONSKI, V. J. and TURNER, J. U.**:** *Using Monte-Carlo variance reduction in statistical tolerance synthesis,* Computer-Aided Design, Volume 29, Issue 1, pp. 63-69, 1997.

[17] SOLTI, E.: *Tolerance computations of Economical Manufacturing.* Technical Publisher, Budapest, 1968. (in Hungarian).

[18] TÓTH, T.: *Interactive Programme System for Determining the Optimum Machining Tolerances Having Regard to Assembly Requirements,* Proceedings of the Twenty-Ninth International Matador Conference, Manchester, pp. 83-91, 1992.

[19] WITTWER, J. W., CHASE, K. W. and HOWELL, L. L.: *The direct linearization method applied to position error in kinematic linkages,* Mechanism and Machine Theory, Volume 39, Issue 7, pp. 681-693, 2004.

[20] YANG, C. C. and NAIKAN, A. V. N.: *Optimum design of component tolerances of assemblies using constraint networks,* International Journal of Production Economics, Volume 84, Issue 2, pp. 149-163, 2003.

Production Systems and Information Engineering
Volume 5 (2009), pp. 139-150.

QUALITY ASSESSMENT OF MOTION PICTURE TRANSMISSION OVER DIGITAL CHANNELS

ATTILA K. VARGA
University of Miskolc, Hungary
Department of Automation
varga.avarga@uni-miskolc.hu

DÉNES DALMI
University of Miskolc, Hungary
Department of Automation

dalmi2@mazsola.iit.uni-miskolc.hu

[Received January 2009 and accepted April 2009]

Abstract. Digital technology forms our environment day by day and we can enjoy its advantages in more and more fields everyday. As a matter of fact television and computer networking technology take over the leading role. The real error analysis of digital channels is the best way for modeling the behaviour of motion picture transmission over digital channels.

In this paper digital transmission has been focused on considering such factors as noises, disturbances and movement as well as the mathematical modeling of digital transfer channels has been shown. The paper first presents the most important standardized subjective quality assessment methods described in the ITU-R BT.500 recommendation. We briefly summarise why these subjective tests are so important. Finally, we discuss the implementation of the new subjective video quality measurement related to impaired digital quality television programs. Our aim is to improve these subjective picture quality assessment methods to get sophisticated results, which correlate better with objective picture quality test results.

Keywords: digital broadcasting, digital cable television, head-end, digital channels, transport stream, subjective test, video quality analysis

1. Aims and Scope of the Paper

Several articles and studies have investigated the quality of telecommunication transfer [1]. ITU-T Recommendation P.800 describes methods and procedures for conducting subjective evaluations of telecommunication transmission quality. ITU-T recommendation [2] gives an objective method for determining voice quality, described in P.862 Recommendation, which is known as "Perceptual Evaluation of Speech Quality" (PESQ).

Although we cannot find a recommendation or standard for the objective quality measurment of video transfer, but subjective measuring algorithm exist [3]. The development of picture quality analysis algorithms available today started with still image models which were later enhanced to also cover motion pictures. The measurement paradigm is to assess degradations of a decoded video sequence output from the network in comparsion to the original reference picture.

The main objective of this paper is to present what kind of assessment tests have been used for examining the quality of digital channels and to describe the standards and subjective methods we used for determining the sources of the errors in the transfer channels.

2. Introduction

In 2004, the Department of Automation (University of Miskolc, Hungary) won a three-year project (GVOP) in the field of Digital Broadcasting the aim of which is to develop software and hardware modules for Digital Cable Television (CATV) head-ends in cooperation with one of the most famous and world-wide known Hungarian Digital CATV components manufacturing company called CableWorld Ltd. A team formed by students and a group of the staff of the Department carried out the developments at the University, mainly the software developments and the installation of the head-end were performed in the laboratory of the Department of Automation.

For the past few years we have dealt with subjective and objective picture quality measurements of digital television streams in the Digital Television Laboratory of the Department of Automation. After we had analysed the results of our subjective tests and drawn the conclusions, we started new subjective quality measurements, which focus on the video quality of digital television streams, so-called transport streams having different bit-rates.

Compression methods for digital television use different compression algorithms. Quality measurements are used to find the best compression method. There are two main categories of comparison methods: the objective video quality evaluation method based on mathematical calculations and the subjective video quality evaluation methods based on tests performed by the audience.

Digital television streams are compressed according to the MPEG-2 or MPEG-4 standards. Nowadays digital television broadcasting systems often use statistical multiplexers. In statistical multiplexing, the communication channel is divided into an appropriate number of variable bit-rate digital channels or data streams. Our goal is to determine the lowest bit-rate, which has still acceptable quality. This bit-

rate would be used in statistical multiplexers as the minimum bit-rate. Consequently, we use these quality measurements in order to find the compression parameters, which still result in acceptable video quality.

3. Subjective Motion Picture Quality Assessment Methods

In this section we would like to introduce the most common subjective quality assessment methods of the digital television picture [2].

International recommendations for subjective quality assessment of television picture consist of specifications of how to perform many different types of subjective tests. Subjective assessment methods are used to establish the performance of television systems. Measurements are therefore applied, which more directly anticipate the reactions of those who might view the tested systems. In this regard, it is understood that it may not be possible to fully characterize the system performance by objective means. Consequently, it is necessary to supplement objective measurements with subjective measurements.

In the course of a typical subjective quality test, a number of non-expert observers are selected, tested for their visual capabilities, shown a series of test scenes for about 10 to 30 minutes in a controlled environment and asked to score the quality of the scenes in one of a variety of manners.

In general, there are two types of subjective assessments. First, there are assessments that bring about the performance of systems under optimum conditions. These are usually called quality assessments. Second, there are assessments that create the ability of systems to retain quality under non-optimum conditions associated with the transmission or emission called impairment assessments. Some of these test methods are double-stimulus where viewers rate the quality or the change in quality between two video streams (reference and impaired). Others are single-stimulus where viewers rate the quality of just one video stream (the impaired one). These methods will be later described.

In a modern television system, however, the picture quality is not constant over time due to the compression streams. In the case of statistical multiplexing, the picture quality is a function of the complexity of the program material and the continuous operation of the transmission system. The selection of the assessment method is affected by a number of procedural elements. These are the viewing conditions, the choice of observers, the scaling method to score the opinions, the reference conditions, the signal sources for the test scenes, the timing of the presentation of the various test scenes, the selection of a range of test scenes and the analysis of the resulting scores.

A description of the various subjective measurement methods provides some insight in the following sections.

3.1. Double-stimulus Impairment Scale Method

The double-stimulus Impairment Scale (DSIS) is a subjective assessment method when observers are shown multiple reference scenes and degraded scene pairs. The reference scene is always shown first. Scoring is on an overall impression scale of impairment.

Table 1. Five-grade scale recommended by ITU

Five-grade scale	
Quality	**Impairment**
5 Excellent	5 Imperceptible
4 Good	4 Perceptible, but not annoying
3 Fair	3 Slightly annoying
2 Poor	2 Annoying
1 Bad	1 Very annoying

This scale is commonly known as the 5-point scale, where 5 equals the imperceptible level of impairment and 1 shows the very annoying level as shown in Table 1.

3.2 Double-stimulus Continuous Quality-scale Method

In case of the Double-stimulus Continuous Quality-scale (DSCQS) method, observers are shown multiple sequence pairs with the reference and degraded sequences randomly first. Scoring is on a continuous quality scale from excellent to bad where each sequence of the pair is separately rated but in reference to the other sequence in the pair. Analysis is based on the difference in rating for each pair rather than the absolute values.

3.3. Single-stimulus Methods

Multiple separate scenes are shown in the Single-stimulus methods. There are two approaches: SS with no repetition of test scenes and SSMR where the test scenes are repeated multiple times. Three different scoring methods are used. The adjectival scoring method has a 5-grade impairment scale, and half-grades may be

allowed. The numerical scoring method has an 11-grade numerical scale, useful if a reference is not available. And finally there is Non-categorical scoring, where assessors can score in a continuous scale with no numbers or a large range.

3.4. Stimulus-comparison Method

The stimulus-comparison method is usually implemented with two well-matched monitors but may be done with one. The differences between sequence pairs are scored in two different ways: the adjectival scale is 7-grade, +3 to -3 scale labelled: much better, better, slightly better, the same, slightly worse, worse, and much worse, while the Non-categorical is a continuous scale with no numbers or a relation number either in absolute terms or related to a standard pair.

3.5. Single Stimulus Continuous Quality Evaluation

The Single Stimulus Continuous Quality Evaluation (SSCQE) is performed with a program, as opposed to separate test scenes, which is continuously evaluated over a long period of 10 to 20 minutes. Data are taken from a continuous scale every few seconds. Scoring is a distribution of the amount of time a particular score is given. This method relates well to the time variant qualities of new compressed systems. However, it tends to have a significant content of program quality in addition to the picture quality [4].

4. Statistical Multiplexing

The flexibility of the MPEG-2 coding system provides the opportunity to broadcast digital television streams, which have more or less bit-rates. Everybody knows that the picture contains more information and has better quality when the rate of the stream, which transmits the compressed picture, is higher. In case of still or slowly moving picture sequences, which do not contain fine details, there is a limit, above which there is no use increasing the data rate, the picture, which has good quality, cannot be better at the receiver side. The change of the picture content and the moving of picture elements increase the amount of information to be transferred. Consequently, to observe the video quality, the data rate must be raised.

The creation of data rate depending on the picture content only makes sense when we can utilize the unused data rate range. In different transmission networks, where more TV programmes can be simultaneously transmitted, in the spaces, which become vacant, one or more TV programmes can be delivered if we can control the resulting data rate.

Statistical multiplexing means that at the transmitter site we compress the data stream with content-dependent data rate; however, we should meet the requirements that the resulting data rate cannot be higher than a predefined value. It is also important to determine a predefined order with which we ensure how much data rate will be allocated to the given programme in case of a large bit-rate demand at the same time [3].

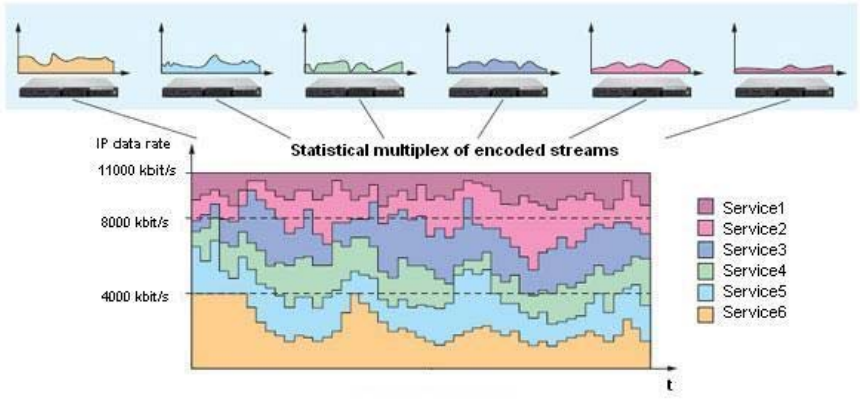


Figure 1. Statistical multiplexing

Figure 1. shows how the statistical multiplex works, so the digital television streams which are coming from different locations (e.g. studios) with variable bit-rates are added in one statistical multiplex stream.

With subjective quality measurements of digital TV streams, the minimum level of bit-rate and other coding parameters, such as GOP (Group of Pictures) size and structure, as well as video picture parameters like brightness, contrast, saturation, can be determined. Nowadays there is a significant demand for these subjective results.

5. Subjective Video Quality Measurements

Measuring the quality of digital transfer channels can be carried out by using subjective methods described above. The main idea of measuring subjective video quality is the same as in the Mean Opinion Score (MOS) for audio. Many parameters of the viewing conditions can influence the results, such as room illumination, display type, brightness, contrast, resolution, viewing distance, and the age and educational level of experts. There are an enormous number of ways of

showing video sequences to experts and to record their opinion. A few of them have been standardized.

These methods can be used for several different purposes including, but not limited to, selection of algorithms, ranking of audiovisual system performance and evaluation of the quality level during an audiovisual connection. Source might be a TV0 type signal given in ITU-R Recommendation BT.601-5 [5]. We can test audio channel (without video), or video channel (without voice) and audiovisual channel (voice and video).

In this section we would like to describe our previous subjective picture quality measurements, and then we would like to go into details about our new measurements.

5.1 Short Presentation of Previous Quality Tests

We have previously executed three different types of subjective picture quality tests of digital television pictures coming from different digital television channels. We used a wide screen LCD television for the experiment, whose screen could be separated into two parts. We chose three different digital television channels: satellite, cable and terrestrial. We selected three different programs: m2, Duna and Autonómia, which can be freely received in Hungary. The observers were undergraduates and one test session consisted of 5-15 of them. In the first test, observers rated the still pictures one after the other. In the second one, picture sequences were displayed on the two separate screens, so students had to evaluate the picture quality simultaneously. Finally, in the last test, observers assessed the quality of short motion picture sequences.

The evaluation was created by taking into account three aspects: sharpness, naturalness and subjective order. Therefore, observers had to determine an order between pictures A and B. They could note the results in an evaluation form. Test sessions took about 20-30 minutes. One test session comprised 8-12 pairs of 10-second pictures, covered the possible combination of different sources, such as satellite vs. cable. Between pictures there was a 10-second interval for the evaluation. Before the test pictures there was a mid-grey picture as mentioned in the ITU standard. We evaluated the test results by counting the scores of the observers in the different categories. In the serial subjective test of still pictures, we collected 216 scores, according to which the cable system was given most of the scores in each category. In the serial test of motion pictures, we obtained a varied result, from the 243 scores gathered, the terrestrial system dominated in the sharpness category, while the satellite system was given most of the votes in the naturalness and the subjective order categories [5].

Drawing the conclusions, we can make some important remarks. First of all, we should create some teaching methods for video assessment, so that the non-expert observers could prepare for assessing the quality. It is very important to teach the observers what they should pay attention to before the real test, because it greatly influences the test results. The experiment leader should explain and demonstrate the evaluation categories (naturalness, sharpness, saturation, hue, etc.), the typical errors, which can occur in the digital video streams, and naturally the essential information about the subjective quality assessment (number of test sequences, the duration of the scoring period, the scoring scale, etc.). In our opinion, by using a well-implemented teaching method, the fidelity of the subjective quality assessment can be improved.

Another important point is to select and record the test material in an appropriate way. In our previous subjective quality measurements it was a serious problem that the test sequences were recorded after the error correction on the receiver side and not at the end of the transmission channel before the error correction. In the new subjective quality assessment, it was also a difficult task how to record test samples with various bit-rates. We provide the related information in the following section.

We should also consider the laboratory circumstances (the distance between the screen and the observers, the resolution and other parameters of the television set, etc.). The ITU recommendation has good criteria to establish the appropriate laboratory environment; however, it has financial implication.

Finally, we should find a better way to record the scores of the observers, because so far they have filled a voting form. We had to evaluate thousands of scoring papers, which resulted in mistakes. Consequently, a subjective quality assessment application is developed in order to help our work.

5.2. Subjective Quality Measurement

As mentioned previously, our purpose is to conduct some subjective video quality tests of digital television streams, which have various bit-rates.

Subjective Quality Assessment Supporter Application

For these measurements we have developed an application in Java environment, which provides a graphical interface in order to assess the digital television video easily.

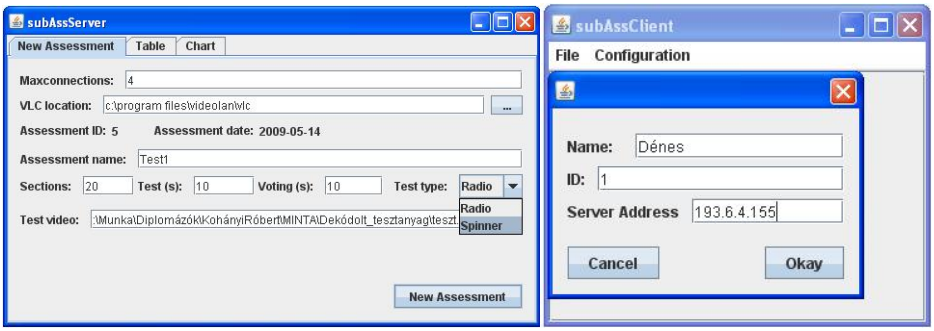


Figure 2. Subjective quality assessment software

The program has two parts: the server and the client, which can be seen in Figure 2. The experiment leader, who conducts the measurement, can configurate or customize the subjective quality test on the *New Assessment* tab in the server software. First, the *Maxconnections* field has to be set, which determines the number of observers. Then, the experiment leader should give the path of the VLC location. If it is well configurated, then after the start of the new assessment, the VLC media player will display the test sequences. The assessment name and date are automatically set by the program. In the following steps the experiment leader should give the name of the assessment, set the number of sections in the test session, configurate the duration of one test sequence and the scoring period in seconds and select the type of the test scale, which can be a 5-grage scale recommended by ITU as it is shown in *Table 1.* or a spinner, which is a 100-grade continuous scale. Finally, the path of the test material has to be set.

The observers should run the client program and set some parameters, such as the name, the unique ID and the IP of the computer on which the server application runs.

When the experiment leader starts the measurement, which can be automatic or manual, the voting screen will automatically appear on the client screen and the observers will have a defined amount of time to score the quality. The client software sends the scores to the server application, which stores them in its database. When the subjective measurement is finished, the experiment leader can evaluate the results in a table or in a chart. The table contains the assessment ID, the assessment name and date, the assessor ID and name, the section number and the quality score. With SQL commands, the experiment leader can create some queries in order to filter the huge amount of data. In the chart, the results of a given assessment can be seen, where the two axes are the number of sections and the mean value of the scores given by the observers.

Recording the Test Material

Our first task was to record digital television video samples, which have different bit-rates. *Fig.3.* presents the environment how we recorded the test material.

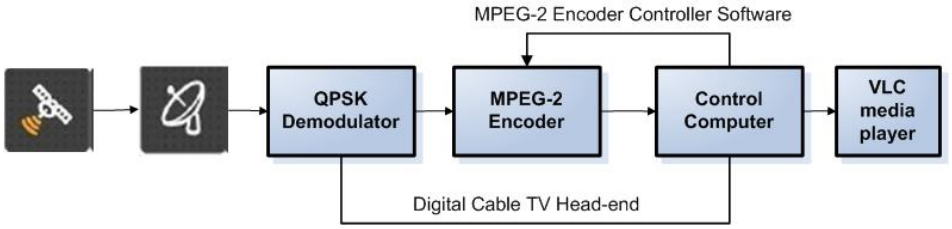


Figure 3. Environment for Recording the Test Material

In the Digital Television Laboratory we used the Digital Cable TV Head-end, which contains special hardware devices developed by CableWorld Ltd. The QPSK demodulator is used to receive the digital transport streams broadcasted via satellite channel. The demodulated transport stream is then sent to the MPEG-2 Encoder. With the MPEG-2 Encoder Controller application running on the Control Computer, the coding parameters and the bit-rates of the transport stream could be configurated. In the final step, this encoded transport stream was displayed with the VLC media player. We used this media player to record video samples.

The problem was that we could not record test samples with various bit-rates continuously; it was the fault of the VLC media player. Therefore, we recorded 10-second video samples and concatenated them into one test video sequence, which could be later used for the subjective quality measurements. However, we have not found appropriate MPEG-2 editor software yet, with which we can concatenate the split sections without re-encoding them. So it is a problem, which needs to be solved in the future.

6. Evaluating the Subjective Quality Assessment

We established a quality assessment environment in our laboratory. We created a computer network with 9-12 personal and one server computers. Observers used the personal computers to run the client application. On the server machine the experiment leader ran the server application and conducted the subjective quality test. One test session took about 10-20 minutes, because the observers needed to concentrate hard during the quality assessment.

Table 2. Five-grade scale recommended by ITU

Seq. N.	Bit-rate (kbps)	1. Measurement (0-5)	2. Measurement (0-100)
1.	8000	2.75	39.75
2.	992	1.25	4.75
3.	1504	3.75	51.50
4.	4000	4.50	73.25
5.	1104	1.50	8.25
6.	1600	2.50	39.25
7.	2608	5.00	87.75
8.	3504	4.25	79.75
9.	3008	3.75	69
10.	2800	4.50	67.75
11.	1904	3.50	33.50
12.	1200	2.25	21
13.	6000	3.50	76
14.	1312	1.00	7.75
15.	4512	4.00	67.75
16.	1408	2.25	22.25
17.	2400	3.25	65
18.	5008	4.00	73
19.	2000	4.25	75.50

So far we have only a few number of test results as described in *Table 2.*We used test material including 19 sections with different bit-rates. In the first and the second measurements the mean of the quality scores can be seen. The difference between the two measurements is the scoring scale, which was used for the test. It can be seen that the video sequence, which has a higher bit-rate, was given better quality scores, but there are discrepancies in the test results. It is important to mention that this result is not representative because the number of assessors who were involved in our assessment is less than 10.

To give a significant result we need to repeat this measurement with a large number of observers. According to our assumption, the lowest bit-rate which has still acceptable quality is about 1500 Kbit/s. However, it will be our future work to verify it.

7. Conclusions

An important issue in choosing a test method is the basic difference between the methods that use explicit references and methods that do not use any explicit reference. If we want to determine the quality of an audio-visual transfer channel, then we can do it using a subjective measuring algorithm.
The accuracy of perceptual objective test methods can be verified by comparison with subjective video quality tests. However, subjective testing can be both time-consuming and costly. In order to achieve statistically relevant results a huge test population must be evaluated.
The possible number of subjects in a viewing and listening test (as well as in usability tests on terminals or services) is between 6 and 40. Four is the absolute minimum for statistical reasons, while there is rarely any point in going beyond 40. The actual number in a specific test should really depend on the required validity and the need to generalize from a sample to a larger population.
In general, at least 15 subjects should participate in the experiment. They should not be directly involved either in picture or audio quality evaluation as part of their work and should not be experienced assessors.
In case of 1500-4000 scores the result might be inconsistent. If the number of observers less than 15, e.g. 6 then we expect binomial distribution of voting, and if the number of observers more than 15, then the distribution is normal.

REFERENCES

[1] ITU-R Recommendation BT.500-11: *Methodology for the subjective assessment of the quality of television pictures,* International Telecommunication Union, Geneva, Switzerland, pp. 2-24. 2002.
[2] BEERENDS J. G.: *Audio Quality Determination Based on Perceptual Measurement Techniques, Applications of Digital Signal Processing to Audio and Acoustics*, Ed. M. Kahrs and K. Brandenburg, Kluwer Academic Publishers, 1998.
[3] ITU-T REC. P.862: *Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs*, ITU Recommendation, 2001.
[4] ITU-T REC. P.910: *Audiovisual quality in multimedia services, Subjective video quality assessment methods for multimedia applications*, ITU Recommendaton, 1999.
[5] ITU-R REC. BT.601-5: *Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios*, ITU Recommendation, 1995.
[6] ITU-T REC. G-114.: *One-way transmission time*, ITU Recommendation, 2003.


Production Systems and Information Engineering
Volume 5 (2009), pp. 151-166.

BEHAVIOUR BASED CONTROL WITH FUZZY AUTOMATON IN VEHICLE NAVIGATION

DÁVID VINCZE
University of Miskolc, Hungary
Department of Information Technology
david.vincze@iit.uni-miskolc.hu

SZILVESZTER KOVÁCS
University of Miskolc, Hungary
Department of Information Technology,
Technical University of Kosice, Slovakia
Department of Cybernetics and AI
szkovacs@iit.uni-miskolc.hu

[Received January 2009 and accepted April 2009]

Abstract. From the viewpoint of Behaviour based Control many control tasks can be divided into separate behaviour components. By defining the relevant behaviour components, the actual control action can be constructed based on the individual control actions of the component behaviours. In this case the control action is either related to an individual behaviour component or to a fusion of behaviour components based on their relevance to the actual situation. This paper adapts the concept of fuzzy automaton for achieving the decision related to the relevance of the behaviour components in the task of the navigation of an autonomous vehicle. In the structure applied, the relevance of the behaviour components is approximated by a fuzzy rule interpolation (FRI, namely the FIVE method) based fuzzy automaton. The main reason for the FRI application is the state-transition rule-base simplification of the fuzzy automaton. In case of FRI, sparse rule bases (incomplete rule bases) are acceptable, because derivable rules can be omitted intentionally, saving construction time and reducing the complexity of the state-transition rule-base. The paper also provides a brief overview of Behaviour based Control and fuzzy rule interpolation (FRI). For demonstration purposes the paper gives a simple example of state-transition rule-base construction in case of the vehicle navigation task mentioned.

Keywords: behaviour based control, fuzzy automaton, fuzzy rule interpolation, FIVE, vehicle navigation control

1. Introduction

The main building blocks of Behaviour based Control (BBC, a comprehensive overview can be found in [14]) are the behaviour components themselves. The behaviour components can be copies of typical human or animal behaviors, or can be artificially created behaviours. The actual behaviour response of the system can be formed as one of the existing behaviour components, which gives the best match for the actual situation, or a fusion of the behaviour components based on their suitability for the actual situation. Encoding the behaviour components can be realized with simple reflexive agents, which assign an output response to each input situation.

In the case when more than one behaviour components are simultaneously competing for the same actuator an aggregation or selection of the behaviour components is necessary. Handling multiple behaviour components in a BBC system can be done in two ways. The first is the competitive way, when the behaviour components are assigned priorities, and the behaviour component with the highest priority takes precedence, while the behaviours with lower priorities are simply ignored. The second is the cooperative way when the outputs are fused based on various criteria.

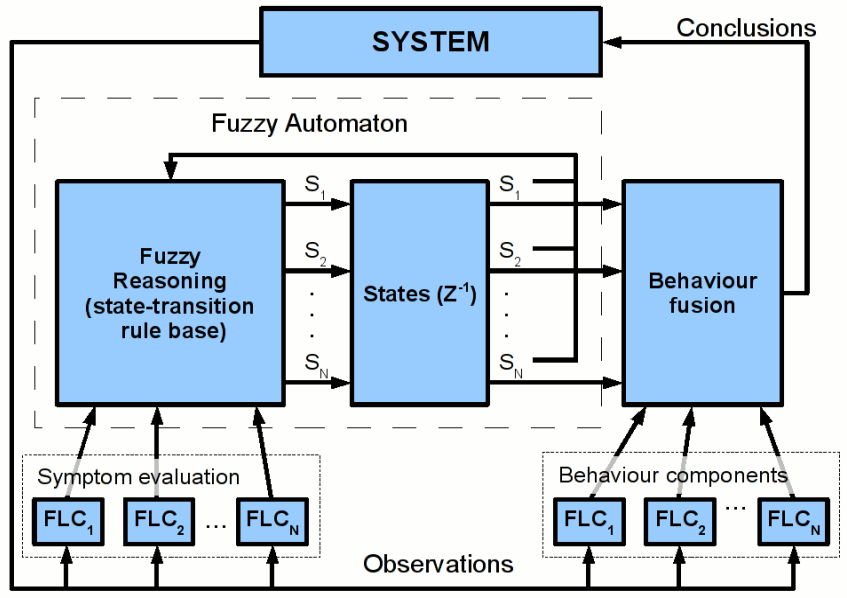


Figure 1. Diagram of the fuzzy automaton

For achieving the decision related to the relevance of the behaviour components this paper adapts the concept of fuzzy automaton. (See the diagram of the suggested fuzzy automaton based system in Fig. 1). The system consists of not only the automaton but the behaviour fusion component and various component behaviours implemented as fuzzy logic controllers (FLC). The state variables characterize the relevance of the component behaviours. The state-transition rule base of the automaton applies fuzzy rule interpolation (namely the FIVE method) for state-transition evaluation. The previous states are fed back to the automaton and the conclusion given by the automaton is used as a weight in the behaviour fusion component for determining the final conclusion of the BBC. The conclusion of the fuzzy automaton will be the new system state for the next step of the behaviour fusion. The behaviour fusion component can also be implemented by fuzzy reasoning (e.g. using fuzzy rule interpolation), or simply as a weighted sum. The symptom evaluation components provide a kind of preprocessing for the automaton based on the observations gathered. These components can also employ FRI techniques.

Embedding fuzzy rule interpolation the model always gives an usable conclusion even if there are no rules defined for the actual observations. Hence the application of sparse rule bases (not complete) can be beneficial, because derivable rules can be omitted intentionally, radically simplifying the rule base creation, saving time-consuming work. The example application of the paper is also based on sparse (not a complete) rule bases. The main reason of applying sparse rule bases and FRI in this case is the simple adaptation of expert knowledge to the system. The existing knowledge is naturally sparse as the experts concentrate on giving the main state-action rules only. On the other hand, having sparse rule bases also helps the final parameter optimization process, as it has usually fewer tunable parameters than complete rule bases.

In the next section, the FRI method FIVE will be introduced in more detail, as it is a quick and simple FRI method. It has the speed benefit against other FRI methods in the price of handling crisp observations and crisp conclusions only. (This makes no real drawback in the example.) Next a simple application example will be presented, which applies the proposed FRI based BBC structure for autonomous vehicle navigation. The vehicle follows pre-defined waypoints while avoiding collision with obstacles and walls. The vehicle navigation example includes competitive and cooperative behaviour components as well.

2. Fuzzy Rule Interpolation

2.1. FRI Introduction

Traditional fuzzy reasoning methods (e.g. the Zadeh-Mamdani compositional rule

of inference (CRI) and the Takagi-Sugeno reasoning method) are demanding complete rule bases, and hence the construction of a classical rule base requires extensive work to define all the required rules. In contrary, the application of fuzzy rule interpolation (FRI) methods, where the derivable rules are missing on purpose (as FRI methods are capable of providing reasonable (interpolated) conclusions even if none of the defined rules fire under the current observation) allows avoiding a considerable amount of unnecessary work in the construction of the rule bases, because the rule base of an FRI controller can contain the most significant fuzzy rules only. On the other hand, most of the FRI methods are sharing the burden of high computational demand, e.g. the task of searching for the two closest surrounding rules to the observation, and calculating the conclusion at least in some characteristic α-cuts. Additionally, in some methods interpreting the fuzzy conclusion gained is not straightforward [8] even if there has been a great deal of effort to rectify the interpretability of the interpolated fuzzy conclusion [16]. In [1] Baranyi *et al.* give a comprehensive overview of recent existing FRI methods. Moreover, some of the FRI methods need special extension for the multidimensional case (e.g. [2]-[3]) because they are originally defined for one dimensional input space. In [19] Wong et al. gave a comparative overview of the multidimensional input space capable FRI methods and in [2] Jenei introduced a way for axiomatic treatment of the FRI methods. In [6] Johanyák *et al.* introduce an automatic way for direct sparse fuzzy rule base generation based on given input-output data. Many of these methods are hardly suitable for real-time applications due to the high computational demand (notably the search for the two closest surrounding rules to an arbitrary observation in the multi-dimensional antecedent space). Some FRI methods, e.g. LESFRI [7] or the method introduced by Jenei *et al.* in [3], eliminate the search for the two closest surrounding rules by taking all the rules into consideration, and therefore speed up the reasoning process. An application oriented aspect of the FRI emerges in the concept of FIVE (Fuzzy Interpolation based on Vague Environment), where the fuzziness of the antecedent and consequent fuzzy partitions is replaced by the concept of vague environment. This makes a speed benefit against other FRI methods, but it has the price of handling crisp observations and crisp conclusions only. It is a real disadvantage of FIVE, but in many direct FRI control applications, like the example in this paper, where the fuzzy conclusion is not required, it has no effect. In the followings the method FIVE will be introduced briefly.

2.2. The FRI "FIVE"

The FIVE method was originally introduced in [9], [10] and [11] and it was developed to fulfill the speed requirements of direct fuzzy control. In this case the conclusions of the fuzzy controller are applied directly as control actions in a real-time system, so the concept of the FIVE method is an application oriented aspect of the FRI techniques. Most of the control applications serve crisp observations and

require crisp conclusions from the controller. Adopting the idea of the vague environment (VE) [4], FIVE can handle the antecedent and consequent fuzzy partitions of the fuzzy rule base by scaling functions [4], therefore it can turn the task of fuzzy interpolation to crisp interpolation. The idea of a vague environment is based on the similarity or in other words the indistinguishability of elements. In a vague environment the fuzzy membership function $\mu_A(x)$ indicates the level of similarity of x to a specific element a which is a representative or prototypical element of the fuzzy set $\mu_A(x)$, or it can be interpreted as the degree to which x is indistinguishable from a [4]. Two values in a vague environment are ε-distinguishable if their distance is greater than ε, where the distances are weighted distances. The weighting factor or function is called scaling function [4]. The scaling function serves the purpose of describing the shapes of the fuzzy sets in the partition. After determining the vague environment of both the antecedent and consequent part universes (the scaling function or at least the approximate scaling function [9], [11]), every member set of the fuzzy partition can be characterized by points in that vague environment (e.g. the approximated scaling function s shown in Fig. 3).

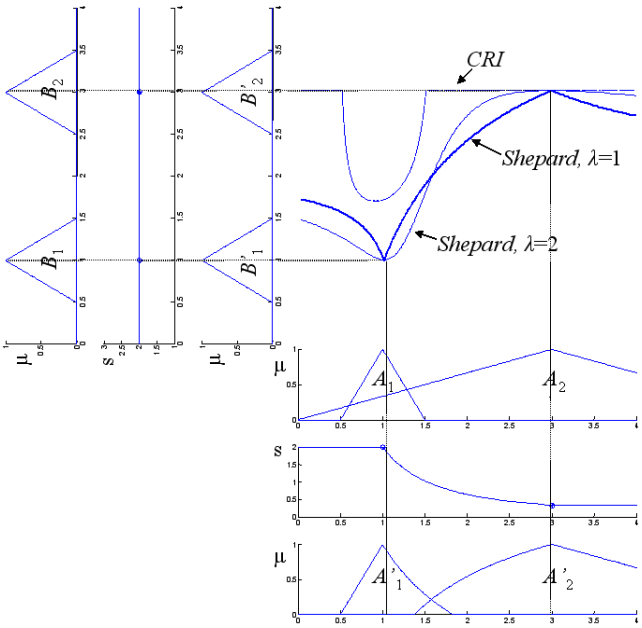


Figure 2. Interpolation of two fuzzy rules (R_i: $A_i \rightarrow B_i$), by the Shepard operator based *FIVE*, and for comparison the min-max *CRI* with COG defuzzification. λ is a parameter of the Shepard operator

The consequent and antecedent sides of the vague environment and scaling functions can be precalculated and cached, which provides the fastness of the method. Fig. 2 presents an example of a one-dimensional antecedent and consequent system with two fuzzy rules. Therefore if the observation is a singleton, any crisp interpolation, extrapolation, or regression method can be adapted very simply for FRI [9], [11]. In method FIVE, because of its simple multi-dimensional applicability, the Shepard operator based interpolation (first introduced in [15]) was adapted (see e.g. in Fig. 2). The Shepard operator based interpolation also appeared in other FRI methods like the stabilized KH interpolator which is proved to hold the universal approximation property in [17] and [18]. Beside its simplicity and therefore high reasoning speed, the original FIVE method has obvious drawbacks: the lack of the fuzziness on the observation side and on the conclusion side. The explanation is that this deficiency is inherited from the nature of the vague environment applied, which describes the indistinguishability of two points and therefore the similarity of a fuzzy set and a singleton only. The lack of fuzziness on the conclusion side has a little influence on common applications where the next step after the fuzzy reasoning is the defuzzification. On the other hand, the lack of fuzziness on the observation side can restrict applicability of the method. Furthermore, an extension of the original FIVE method was suggested in [12], where the question of fuzzy observation is handled by merging vague environments of the antecedent universes and the fuzzy observation. An implementation of FRI FIVE as a component of the FRI Matlab Toolbox [5] can be downloaded from [20] and [21].

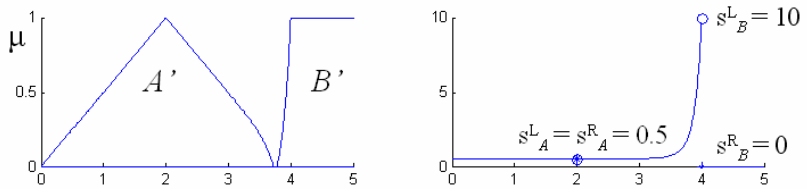


Figure 3. Approximate scaling function *s* generated by non-linear interpolation, and the partition as described by the approximate scaling function (A', B')

3. Vehicle Navigation Example

The example application of the paper is an autonomous vehicle navigation simulation which demonstrates the benefits of the proposed FRI based BBC structures. The goal of the application is to navigate the vehicle around given waypoints in a pre-defined order, while the vehicle should avoid collision with

obstacles and the walls of the room. The vehicle can detect whether some obstacle is standing in its way, and hence whether the planned path of the vehicle seems to be blocked. In this case the vehicle can turn back and head in the opposite direction by reversing the sequence of the waypoints. The example waypoint configuration has four members which correspond to the four corners of the room.

3.1. Circular Waypoint Navigation and Collision Avoidance

For the navigation control the previously proposed BBC structure is adapted (see Fig. 1). The actual states, observations and symptom evaluation and behaviour components of the example are shown in Fig. 4. The suggested BBC has homogeneous FRI knowledge representation. In the following the rule bases for all the BBC components, the symptom evaluation, the state-transition and the behaviour component rule bases will be described and explained in more detail.

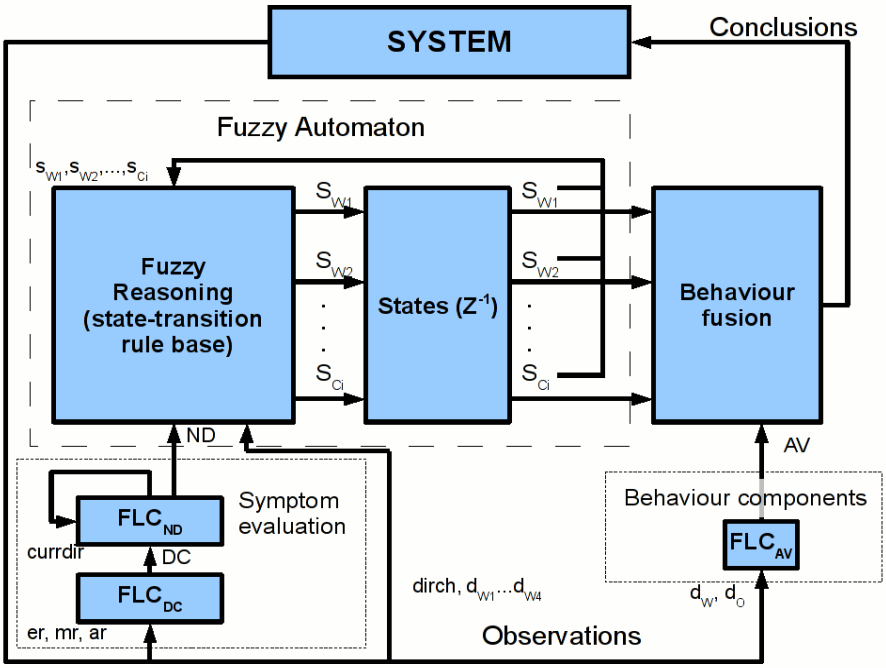


Figure 4. Diagram of the actual fuzzy automaton for the demonstration example

The navigation control is built of the following components: *waypoint approach (one for each waypoint)*, *wall avoidance, obstacle avoidance*, and the *heading direction change*.

The *waypoint approach component* (which is a component of the 'Fuzzy Automaton' labeled block in Fig. 4) partly determines the current state vector, the selection weights of the waypoints. (Then these weights will be applied as selection strengths of the corresponding waypoint directions.) The approximation of the waypoint selection weights is based on the following input parameters: the current position of the vehicle (described by the distances from the four waypoints; denoted respectively: dw_1, dw_2, dw_3, dw_4), the previous selection weights of the four waypoints (namely sw_1, sw_2, sw_3, sw_4 – from the previous state of the automaton), the need for direction changing and the current direction of the vehicle. The need for direction changing component calculates a weight, and if this weight is beyond the value of an adjustable parameter, then a direction change is necessary. The state-transition rule base is very simple, it assigns the highest waypoint selection weights in a predefined sequence which follows the nearest waypoint to the vehicle. A high level of a waypoint selection weight means that the vehicle is mainly heading towards the corresponding waypoint. For expressing the distance from an arbitrary waypoint in the fuzzy rule base, the linguistic terms for the antecedent universes are given as the following: zerus (*Z*), large (*L*). For the state variables related to the waypoint selection weights (*WW* in Table 4-7), there are only two linguistic terms defined: true (*T*) and false (*F*) for the antecedent partitions and zerus (*Z*) and large (*L*) for the consequents. Each element of the waypoint selection weight (partly the state) vector has a separate state-transition rule base, and a similar structure. In the example case it means four state-transition rule bases (equal to the number of the pre-defined waypoints). Each rule base needs to be evaluated with the same measured distances and previous state variables. The conclusion is partly the new state, the normalized weight of the behaviour components heading for the corresponding waypoints, which is used to scale a vector pointing towards the corresponding waypoint.

The collision avoidance strategy consists of as many behaviour components as the number of the walls and obstacles and hence the same number of state values (the weight of the corresponding collision avoidance component) in the state vector. By definition walls are the borders of the room and obstacles are objects which can move freely inside the room. The *wall avoidance components* are very simple. There are as many normalised movement vectors as the number of the walls having a perpendicular direction to the corresponding wall. The state variables are the corresponding repulsion rates, one for each wall avoidance component. The state variables (repulsion rates, $S_{Ci\ i}$ in Fig. 4) are calculated based on the distance from the corresponding wall. The structures of the rule bases are similar and introduced in Table 1. Obstacle avoidance is solved in the same manner as wall avoidance. It has as many component behaviours as the number of the obstacles. They are normalised movement vectors having a direction opposite to the resultant waypoint movement vector. Similarly to wall avoidance the corresponding state variables are their weights calculated based on the distance between the vehicle and the obstacle

in the same way as the states of wall avoidance (see again Table 1). Observations of the *wall and obstacle avoidance components* are the measured distances from each of the walls (denoted: d_w), and the measured distances from each of the objects inside the room (denoted: d_o). The linguistic terms of the antecedent universes are: zerus (*Z*), small (*S*), medium (*M*), large (*L*), and for the consequent universes (*AV*): zerus (*Z*), small (*S*), large (*L*).

The *wall and obstacle avoidance components* use the same rule base structure (see Table 1) for all the required conclusions; only the input distances differ within every evaluation. The conclusions are the state variables (component weights) related to the wall and obstacle avoidance components and applied in the behaviour fusion component in the same manner as for the waypoint direction components.

The structure of the wall and obstacle avoidance state rules are defined in the following form:

RColl$_i$: **If** d_w = A$_i$ **Then** AV = B$_i$

Table 1. Wall and obstacle avoidance weight rule base

RColl	d_w, d_o	AV
Rule 1	Z	L
Rule 2	S	S
Rule 3	M	Z
Rule 4	L	Z

The behaviour fusion part of the example is a simple convex combination of the component behaviours with the corresponding weights (state variables).

3.2. Heading Direction Change Extension

As already mentioned earlier in the case when the way of the vehicle seems to be blocked in the current direction, the vehicle can change its heading, by assigning the waypoints in the reverse order. This direction change decision is made by the *heading direction change* symptom evaluation (see Fig. 4) component. The observations needed for this component (see Fig. 5) are the sum of movement rates of the vehicle and the collision avoidance vector (denoted: *mr*), the summarized rate of the length of the wall and obstacle avoidance vectors (denoted: *ar*). In a hierarchical navigation control, the vehicle could do some other types of movements beyond navigation among the waypoints ('exploration'), hence an 'exploration rate' observation could be also added (*er* in Table 2) to control the level of our example navigation strategy as a component behaviour itself in a more complex system.

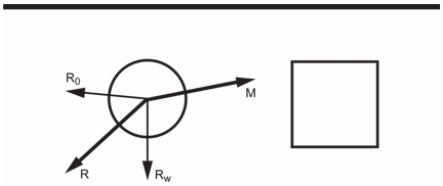


Figure 5. *M* is the movement vector of the vehicle towards the next waypoint, R_o is the repulsion vector of the obstacle, R_w for the wall and *R* is their sum

The linguistic terms of the two antecedent universes of the *heading direction change component* are: zerus (*Z*) and large (*L*). For the conclusion universe (*DC*), which tells whether to change the direction of the vehicle or not, the linguistic terms are: false (*F*) and true (*T*). The rule base consists only of three rules, which can be seen in Table 2. The rules are defined in the following form:

RDirCh$_i$: **If** *er* = $A_{1,i}$ **and** *mr* = $A_{2,i}$ **and** *ar* = $A_{3,i}$ **Then** *DC* = B_i

Table 2. Direction changing behaviour component decision rule base

RDirCh	*er*	*mr*	*ar*	*DC*
Rule 1	Z			F
Rule 2	L	Z	L	T
Rule 3	L	L		F

One more rule base is used to determine the new heading direction for the vehicle. Two observations are required for this subcomponent, which is also a fuzzy automaton in the symptom evaluation component, with one state variable (see in Fig. 4): a value which tells whether a direction heading change is necessary (denoted: *dirchg*) (this is the conclusion above, see Table 2) and the current heading direction state (denoted: *currdir*). The linguistic terms for the antecedent universes are the following: for expressing the need of direction changing: true (*T*), false (*F*), for expressing the current direction and also for the consequent universe, which gives the new direction: clockwise (*C*), counter-clockwise (*CC*). The conclusion of the symptom evaluation will be the new state of the fuzzy automaton (direction).

The state-transition rule base of the fuzzy automaton embedded into the symptom evaluation can be seen in Table 3, and the rules can be interpreted according to the following form:

RNewDir$_i$: **If** *dirchg* = $A_{1,i}$ **and** *currdir* = $A_{2,i}$ **Then** *ND* = B_i

Table 3. Selection of current direction decision rule base

RNewDir	*dirchg*	*currdir*	*ND*
Rule 1	F	C	C
Rule 2	F	CC	CC
Rule 3	T	C	CC
Rule 4	T	CC	C

Having the rule bases for direction changing decision the state-transition rule base of the waypoint selection weights can also be extended with direction changing. Some new observations should be added to the waypoint selection weights state-transition rule base which were introduced earlier: the current heading direction (denoted: *dir*) and a parameter expressing whether the heading direction was changed (denoted: *dirchg*). The newly added antecedent linguistic terms for the necessity of reversing the direction are: true (*T*), false (*F*). For the current direction: clockwise (*C*), counter-clockwise (*CC*).

As mentioned, four rule bases are required in this particular case. E.g. having four waypoints in case the direction is clockwise: first to calculate the weight needed to take the vehicle towards the 2nd waypoint, second to direct the vehicle to the 3rd waypoint, third to take the vehicle to the 4th waypoint, and a fourth rule base to navigate the vehicle back to the 1st waypoint. E.g. in case of the first waypoint the waypoint selection weights state-transition rule base has the following meaning (see Table 4): the first rule means that when the corresponding waypoint (1st) is reached by the vehicle then that waypoint (1st) should be abandoned, hence the weight of the waypoint (1st) will be zerus (*Z*). The second rule keeps the vehicle coming to the waypoint (1st) if it has been selected earlier. The third rule stops the vehicle when a direction change is necessary. The fourth rule changes the direction if needed and if the previous heading was towards the next waypoint in the defined sequence (2nd in this particular case). The fifth rule is similar to the fourth one, it changes the direction if required and if the previous heading was the previous waypoint in order (in this case the 4th). The sixth rule serves the purpose of keeping down the weight when the vehicle is going to the next (2nd) waypoint, so do the seventh and eighth rules, but for the remaining two waypoints (4th and 3rd respectively). The ninth means that when the vehicle reaches the previous waypoint in the sequence (4th), it should head for the current waypoint (1st). The meaning of the last rule is very similar to the previous one, but for the opposite heading direction. The rule bases for the 2nd, 3rd and 4th waypoints contain similar rules, the differences are only the rotational numbering of the corresponding next and previous waypoints numbers.

The extended waypoint selection weights state-transition rules are defined in the following form:

RWX$_i$:

If $dw_1 = A_{1,i}$ **and** $dw_2 = A_{2,i}$ **and** $dw_3 = A_{3,i}$ **and** $dw_4 = A_{4,i}$
and $sw_1 = A_{5,i}$ **and** $sw_2 = A_{6,i}$ **and** $sw_3 = A_{7,i}$ **and** $sw_4 = A_{8,i}$
and $dir = A_{9,i}$ **and** $dirch = A_{10,i}$

Then $WW = B_i$

With the rule bases above described the vehicle can cycle around the given waypoints, with direction change in blocked situations, while still avoiding obstacles and walls.

Table 4. First waypoint selection weight with direction changing rule base

RW1	dw_1	dw_2	dw_3	dw_4	sw_1	sw_2	sw_3	sw_4	dir	$dirch$	WW
Rule 1	Z										Z
Rule 2	L				T					F	L
Rule 3	L				T					T	Z
Rule 4			L	L		T			CC	T	L
Rule 5		L	L					T	C	T	L
Rule 6		L				T				F	Z
Rule 7				L				T		F	Z
Rule 8			L				T				Z
Rule 9				Z					C	F	L
Rule 10		Z							CC	F	L

Table 5. Second waypoint selection weight with direction changing rule base

RW2	dw_1	dw_2	dw_3	dw_4	sw_1	sw_2	sw_3	sw_4	dir	$dirch$	WW
Rule 1		Z									Z
Rule 2		L				T				F	L
Rule 3		L				T				T	Z
Rule 4	L			L			T		CC	T	L
Rule 5			L	L	T				C	T	L
Rule 6			L				T			F	Z
Rule 7	L				T					F	Z
Rule 8				L				T			Z
Rule 9	Z								C	F	L
Rule 10			Z						CC	F	L

Table 6. Third waypoint selection weight with direction changing rule base

RW3	dw_1	dw_2	dw_3	dw_4	sw_1	sw_2	sw_3	sw_4	*dir*	*dirch*	*WW*
Rule 1			Z								Z
Rule 2			L				T			F	L
Rule 3			L				T			T	Z
Rule 4	L	L						T	CC	T	L
Rule 5	L			L		T			C	T	L
Rule 6				L				T		F	Z
Rule 7		L				T				F	Z
Rule 8	L				T						Z
Rule 9		Z							C	F	L
Rule 10				Z					CC	F	L

Table 7. Fourth waypoint selection weight with direction changing rule base

RW4	dw_1	dw_2	dw_3	dw_4	sw_1	sw_2	sw_3	sw_4	*dir*	*dirch*	*WW*
Rule 1				Z							Z
Rule 2				L				T		F	L
Rule 3				L				T		T	Z
Rule 4		L	L		T				CC	T	L
Rule 5	L	L					T		C	T	L
Rule 6	L				T					F	Z
Rule 7			L				T			F	Z
Rule 8		L				T					Z
Rule 9			Z						C	F	L
Rule 10	Z								CC	F	L

3.3. Implementation Remarks

It is recommended to arrange the evaluation of these rule bases and observation calculations in a loop. First the waypoint selection conclusions should be calculated, the result vector should be added to the current position of the vehicle. With this new position the distances from the walls and obstacles should be computed, then the wall and obstacle avoidance fuzzy rule bases should be evaluated, these results should be summarized with the current position. This will be the next valid position of the vehicle. Finally we have all the required data to get the conclusion for direction changing. If the direction has to be changed, the direction state variable should be inverted and in the next iteration it should take effect. Following this procedure gives a working FRI model of vehicle navigation

and collision avoidance.

With a simple algorithm the waypoint selection rule bases can be generated based on the number of the defined waypoints. This was implemented as a standalone script written in Python programming language, which can be found at [22]. Using this script, dynamic waypoint insertion/deletion could be achieved with regeneration of the waypoint selection rule base every time the count of waypoints has been modified. A drawback is that this feature implies modifications not only in the rule bases (and the number of rule bases also), but in the rule base evaluation procedures. Achieving the latter requires further research.

For non-commercial purposes the Matlab source of the example of the paper can be accessed free of charge at [22].

Conclusion

Some details of an autonomous surveillance vehicle implementation based on fuzzy automaton and behaviour-based control navigating were examined in the paper. The knowledge representation of the behaviour components, and the state-transition rule-base of the system state approximation were implemented as sparse fuzzy rule bases of the "FIVE" fuzzy rule interpolation method. Beyond the successful application, a notable conclusion of the paper is that by using FRI methods the rule base sizes can be considerably reduced to a fraction of the original sizes. Building a complete fuzzy rule base for the behaviour components introduced in the paper with the same strategies, but with complete rule-base could require approximately a thousand fuzzy rules. On the other hand applying sparse fuzzy rule bases (and fuzzy rule interpolation, in case of 4 waypoints, 4 walls and 2 obstacles) only 71 rules are sufficient. This rule base size is easily implementable even in an embedded FRI fuzzy logic controller. The implementation also proves the real-time suitability of the FIVE fuzzy rule interpolation method itself (in application areas where as a restriction of the method crisp observation and crisp conclusion are sufficient). For non-commercial purposes an implementation of FRI FIVE as a component of the FRI Matlab Toolbox [5] can be downloaded from [20] and [21].

Acknowledgements

This research was partly supported by the Hungarian National Scientific Research Fund grant no: OTKA K77809.

REFERENCES

[1] P. BARANYI, L. T. KÓCZY, AND GEDEON, T. D.: *A Generalized Concept for Fuzzy Rule Interpolation.* IEEE Trans. on Fuzzy Systems, vol. 12, No. 6, 2004, pp 820-837.

[2] S. JENEI: *Interpolating and extrapolating fuzzy quantities revisited – an axiomatic approach*. Soft Comput., vol. 5., 2001, 179-193.

[3] S. JENEI, E. P. KLEMENT AND R. KONZEL: *Interpolation and extrapolation of fuzzy quantities – The multiple-dimensional case*. Soft Comput., vol. 6., 2002, 258-270.

[4] F. KLAWONN: *Fuzzy Sets and Vague Environments*. Fuzzy Sets and Systems, 66, 1994, pp. 207-221.

[5] ZS. CS. JOHANYÁK, D. TIKK, SZ. KOVÁCS, K. W. WONG: *Fuzzy Rule Interpolation Matlab Toolbox – FRI Toolbox*. Proc. of the IEEE World Congress on Computational Intelligence (WCCI'06), 15th Int. Conf. on Fuzzy Systems (FUZZ-IEEE'06), July 16--21, Vancouver, BC, Canada, Omnipress. ISBN 0-7803-9489-5, 2006, pp. 1427-1433.

[6] ZS. CS. JOHANYÁK, R. PARTHIBAN, AND G. SEKARAN: *Fuzzy Modeling for an Anaerobic Tapered Fluidized Bed Reactor*. SCIENTIFIC BULLETIN of "Politehnica" University of Timisoara, ROMANIA, Transactions on AUTOMATIC CONTROL and COMPUTER SCIENCE, ISSN 1224-600X, Vol:52(66) No:2, 2007, pp. 67-72.

[7] ZS. CS. JOHANYÁK, SZ. KOVÁCS: *Fuzzy Rule Interpolation by the Least Squares Method*. 7th International Symposium of Hungarian Researchers on Computational Intelligence (HUCI 2006), November 24-25, 2006 Budapest, pp. 495-506.

[8] L. T. KÓCZY AND SZ. KOVÁCS: *On the preservation of the convexity and piecewise linearity in linear fuzzy rule interpolation*. Tokyo Inst. Technol., Yokohama, Japan, Tech. Rep. TR 93-94/402, LIFE Chair Fuzzy Theory, 1993.

[9] SZ. KOVÁCS: *New Aspects of Interpolative Reasoning*. Proceedings of the 6th. International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Granada, Spain, 1996, pp. 477-482.

[10] SZ. KOVÁCS, AND L.T. KÓCZY: *Approximate Fuzzy Reasoning Based on Interpolation in the Vague Environment of the Fuzzy Rule base as a Practical Alternative of the Classical CRI*. Proceedings of the 7th International Fuzzy Systems Association World Congress, Prague, Czech Republic, 1997, 144-149.

[11] SZ. KOVÁCS, AND L.T. KÓCZY: *The use of the concept of vague environment in approximate fuzzy reasoning*. Fuzzy Set Theory and Applications, Tatra Mountains Mathematical Publications, Mathematical Institute Slovak Academy of Sciences, Bratislava, Slovak Republic, vol.12, 1997, pp. 169-181.

[12] KOVÁCS, SZ.: *Extending the Fuzzy Rule Interpolation "FIVE" by Fuzzy Observation*. Theory and Applications, Springer Berlin Heidelberg, 2006, pp. 485-497.

[13] SZ. KOVÁCS AND L. T. KÓCZY: *Application of an approximate fuzzy logic controller in an AGV steering system, path tracking and collision avoidance strategy*. Fuzzy Set Theory and Applications, In Tatra Mountains Mathematical Publications, Mathematical Institute Slovak Academy of Sciences, vol.16, Bratislava, Slovakia, 1999, pp. 456-467.

[14] PIRJANIAN, P.: *Behavior Coordination Mechanisms - State-of-the-art, Tech-report* IRIS-99-375, Institute for Robotics and Intelligent Systems, School of Engineering, University of Southern California, October (1999)

[15] D. SHEPARD: *A two dimensional interpolation function for irregularly spaced data.* Proc. 23rd ACM Internat. Conf., 1968, pp. 517-524.

[16] D. TIKK AND P. BARANYI: *Comprehensive analysis of a new fuzzy rule interpolation method*. IEEE Trans. Fuzzy Systems, vol. 8, No. 3, June, 2000, pp. 281-296.

[17] D. TIKK, I. JOÓ, L. T. KÓCZY, P. VÁRLAKI, B. MOSER, AND T. D. GEDEON: *Stability of interpolative fuzzy KH-controllers*. Fuzzy Sets and Systems, (125) 1, 2002, 105-119.

[18] D. TIKK: *Notes on the approximation rate of fuzzy KH interpolator.* Fuzzy Sets and Systems, (138) 2, 2003, pp. 441-453.

[19] K. W. WONG, D. TIKK, T. D. GEDEON, AND L. T. KÓCZY: *Fuzzy Rule Interpolation for Multidimensional Input Spaces With Applications*. IEEE Transactions on Fuzzy Systems, ISSN 1063-6706, Vol. 13, No. 6, December, 2005, pp. 809-819.

[20] The FRI Toolbox is available at: http://fri.gamf.hu/

[21] Some FRI applications are available at: http://www.iit.uni-miskolc.hu/~szkovacs/

[22] The example application can be found at: http://www.iit.uni-miskolc.hu/~vinczed/vehnav/


Production Systems and Information Engineering
Volume 5 (2009), pp. 167-175.

LOOK UP TABLE EDITOR FOR SMALL ANIMAL PET INSTRUMENT

Ákos Szlávecz
Budapest University of Technology and Economics, Hungary
Department of Control Engineering and Information Technology
szlavecz@iit.bme.hu

Gábor Hesz
Budapest University of Technology and Economics, Hungary
Department of Control Engineering and Information Technology
hesz@iit.bme.hu

Péter Major
Mediso Ltd., Hungary
major.peter@mediso.hu

Balázs Benyó
Budapest University of Technology and Economics, Hungary
Department of Control Engineering and Information Technology
bbenyo@iit.bme.hu

[Received April 2009 and accepted June 2009]

Abstract. Small animal PET is a widely used instrument for functional examinations in pharmaceutical industry research; its spatial resolution has reached the physically possible limit by now. One important issue of the imaging process is to determine the position of γ photons impacting the surface of the detectors, and to filter them based on their energy. In this paper we introduce the methods developed to complete these tasks for the small animal PET instrument of Mediso Ltd. All the introduced methods have been implemented in an application called LUT-QT. The efficiency of the methods has been evaluated in real experiments.

Keywords: Positron Emission Tomography (PET), position discrimination, energy discrimination

1. Introduction

Positron Emission Tomography (PET) [1], [2] has become a fundamental instrument of functional examinations over the past two decades. Even though its spatial resolution fails to keep level with anatomical examinations, e.g.

X-ray tomography (abbreviated CT), PET allows monitoring the track of a selected molecule, traced with radioactive isotopes, within the organism examined, which may lead to conclusions about the functioning or eventual abnormalities of certain organs. Clinical (i.e. human) PET instruments nowadays operate with a spatial resolution of about 3-5 mm considering the rational cost limits as the consequence of the large size, whereas a spatial resolution of pre-clinical (or small animal) PET instruments has reached the theoretic limit of 1-2 mm attainable with positron annihilation. That quality is necessary because in pharmaceutical industry research the operation of the internal organs of experimental mice needs to be monitored. Therefore, in addition to a very good spatial resolution, small animal instruments have to meet another important requirement: because of the statistical nature of impact mechanism tests of drugs a great number of mice must be examined under identical circumstances to obtain a reliable result, so the examination should be very fast (a couple of minutes). In order to understand the problems of data processing we have addressed, let us review the imaging process in brief.

2. Main steps of PET imaging process

Injecting some positron (β^+) emitting marker into the investigated organism, a positron is produced through decay of the marker. The positron hits an electron within the positron range [3] and both get annihilated. In a medium with a density close to that of water, it is about 1 mm. In consequence of conservation of impulse and energy, two γ photons with 511 keV energy are produced leaving in approximately 180^o, neglecting the non-collinearity effect [4] as can be seen in Figure 1. If we can detect these photons, we can determine the line (line of response, LOR) along which the annihilation has happened. Having the data of several million LORs, we can estimate the density of the marker in each small space unit (the so-called voxel) because the number of lines (LORs) going through a space unit is proportional to the number of local decays. The condition of coincidence ensures that the detected photons originate from the same event: the data gathering device keeps only events arriving within a certain small time window (typically 5ns), otherwise we would get false LORs connecting the positions of two impacts from two independent annihilations. This means that we need to know exactly the LORs and therefore the impact positions of the γ photons in order to have a proper spatial resolution. Generally, detecting photons with such high energy is feasible with scintillation crystal, [5] a material which absorbes the incoming photons with high probability, producing a visible light flash. If we put scintillation crystal needles in a matrix made of reflecting material arranged as septa, the subsequent flash will remain localised in two directions,

which helps the determination of the impact position. Therefore the task is to determine the flashing crystal needle for each event. One of the issues we have addressed was that localisation based on the detector signals.

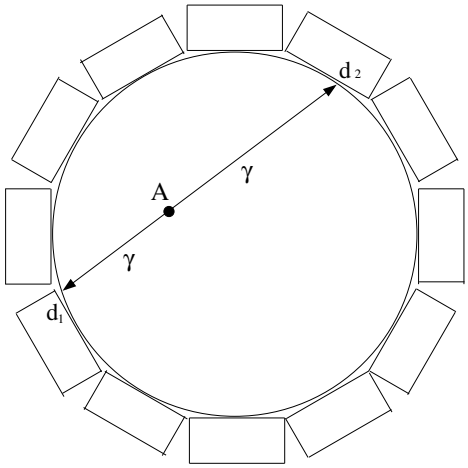


Figure 1. Antiparallel 511 keV γ photons produced from positron annihilation at point 'A' and the line (line of response, LOR) connecting the positions of detection (d_1, d_2)

3. Determining the position of detected γ photons

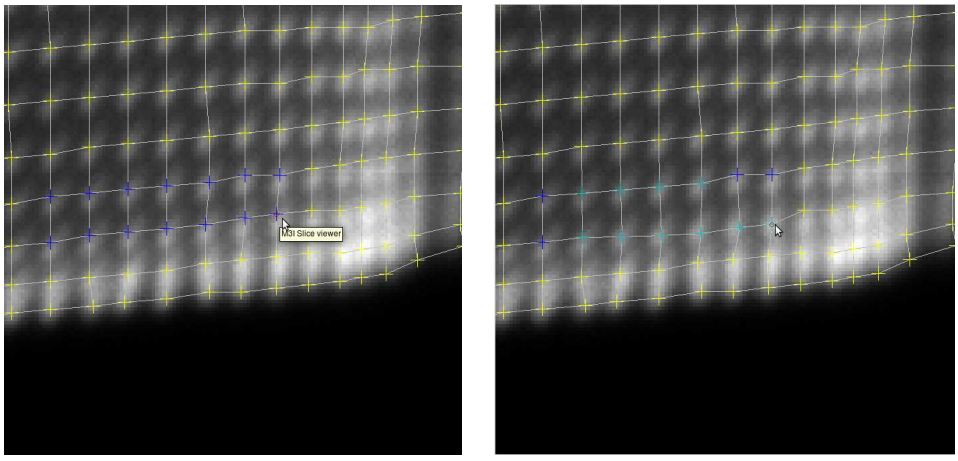
In the case of our device, a Hamamatsu H-9500 position sensitive detector with 16x16 sensors (anodes) converts the flashes of the crystal matrix made of 35x35 pcs, 1.27 mm x 1.27 mm LYSO needles, and the created electrical signals can be used to determine the needle that produced the flash. The setup of the detector module is basically the same as described in [6] although the needles area thinner and the electronics has been modified. In the course of the fast examinations mentioned in the introduction we must handle several hundred thousand events per second, hence the digitalization and processing of all the 256 detector signals would be impossible at this data processing speed. Therefore the usual process is to add up the anode signals with a resistor network, wherein the anode currents are weighted with resistors in such a way that the difference of the resultant currents in the corners of the network is more or less proportional to the coordinate of the anode. This method produces a distorted, but two-dimensional image called Anger image [7]. Thus the task is to prepare a table (Look Up Table, abbreviated LUT) for each detector, which assigns a crystal needle index to arbitrary detector signals (x and y Anger-coordinates). In order to determine a certain detector

signal position map, we start from the image of the homogenously irradiated detector, the so-called Flood-field image, which is actually the impact number as the function of the two-dimensional (2D) position. Each crystal needle has its own light cloud in the Flood-field image, we need to find the centre of the light clouds of the crystal needles and label them with the proper needle indices. As soon as we have found the centre of the light cloud of every crystal needle, we should be able to determine in case of any (x,y) points the light cloud of which needle is the closest to it, and we should assign it to that needle. Because of the speed of data gathering it is worth calculating a two-dimensional map preliminarily, i.e. we determine the closest needle to each point of the image space covering the certain distorted quadratic lattice (the centers of the needle clouds) sized 35x35. In other words, we store the ranges of points closest to the centres (Wigner-Seitz cell or Voronoi-cell) in a matrix, in which a matrix cell, assigned to an incoming event's discrete (x,y) coordinate as an index, contains the proper needle index. As a consequence, based on the coordinate (x,y) produced as a result of measurement, if we have this matrix we immediately know the index of the flashing needle without any further calculation.

During the search for the light clouds of the needles we must first of all filter the image because of the Poisson statistics of the light produced in scintillation and the noise of the electronic signals. We have selected the matrix of the convolution filter used to have isotropic smoothing in the x-y plane, since the light clouds have a circular shape. In case of the 5x5 unit matrix for example, the characteristic distance of smoothing is $\sqrt{2}$ times larger in the direction of the diagonal, so we have chosen such weights as elements of the convolution matrix that a circle with r=2.5 units would cover from a given pixel of the quadratic lattice of 5x5 (a.k.a. disk filter). Then an automatic algorithm finds the valleys between the clouds and then finds the centre of the light clouds in the areas defined by the intersections of the valleys. It seemed to be uncertain to search for a maximum based on the derivative because of the accidental nature of impact numbers and the noises, plus exaggerated smoothing will shift culminations, thus we are searching for the centre of gravity in a given area, where the 'weight' is the impact number. The process is iterative; it defines an environment around the initial point in a given iteration, determines its centre of gravity, defines its environment, etc.

Finding every light cloud automatically is especially difficult in case of the crystal needles at the edges of the detector because of the natural distortions of the Anger image; the spots get closer to each other and they run into one another, furthermore, the signal to noise ratio is the worst in that area. On

the other hand, different optical faults are also possible, e.g. poor optical connection between the detector and the crystal matrix, the lack of optical grease leads to intensity decrease, or an air bubble in the optical grease produces lens impact. Sometimes a deficient crystal needle is misplaced or it has lower intensity or disappears completely. To ensure a good performance, we have decided to integrate the options of check-up and manual correction into LUT-QT after the automatic algorithm. (See Figure 2.)



a., **b.,**

Figure 2. Figure **a.,** shows part of a Flood-field image with mistakenly detected light clouds (crosses mark the spots detected by LUT-QT). Figure **b.,** shows image during manual correcting.

Projecting the measured Flood-field image and the centres of the crystal needles we assume to have found on the graphical user surface of LUT-QT, the grid of the latter is modifiable according to either points or range. In order to enable modification of ranges, we have searched for a 2D transformation (distortion), which can act as displacement, enlargement, rotation, shearing or any required combination of those. It has 8 parameters and can transform a rectangle not only into a parallelogram but into a general trapezoid as well. The input data are the displacements of the four corners of the range to be distorted (4 2D vectors). We get the displacements of the interior points of the initial quadrilateral by weighting the displacements of the corners, just like in case of the coordinates of the point of division. That transformation corresponds to the characteristics listed. The range to be modified and the new corners are displayed in Figure 3 with an interior point P of the range. Using the notation in Figure 3, displacement of point P in direction x is:

$$\mathbf{e}_P(x) = \left(\frac{d_f}{c_f + d_f} \cdot \frac{b_b}{a_b + b_b} \cdot \mathbf{e}_1(x) + \frac{c_f}{c_f + d_f} \cdot \frac{b_j}{a_j + b_j} \cdot \mathbf{e}_2(x) + \right.$$
$$\left. + \frac{d_a}{c_a + d_a} \cdot \frac{a_b}{a_b + b_b} \cdot \mathbf{e}_3(x) + \frac{c_a}{c_a + d_a} \cdot \frac{a_j}{a_j + b_j} \cdot \mathbf{e}_4(x) \right) \quad (3.1)$$

and similarly the y component of the displacement of point P is:

$$\mathbf{e}_P(y) = \left(\frac{d_f}{c_f + d_f} \cdot \frac{b_b}{a_b + b_b} \cdot \mathbf{e}_1(y) + \frac{c_f}{c_f + d_f} \cdot \frac{b_j}{a_j + b_j} \cdot \mathbf{e}_2(y) + \right.$$
$$\left. + \frac{d_a}{c_a + d_a} \cdot \frac{a_b}{a_b + b_b} \cdot \mathbf{e}_3(y) + \frac{c_a}{c_a + d_a} \cdot \frac{a_j}{a_j + b_j} \cdot \mathbf{e}_4(y) \right) \quad (3.2)$$

After the eventual manual modification we can reiterate the centre of gravity search around the new corners. Figure 5/a shows the distorted Anger image and the light spot centres determined with LUT-QT of a detector with several faults.

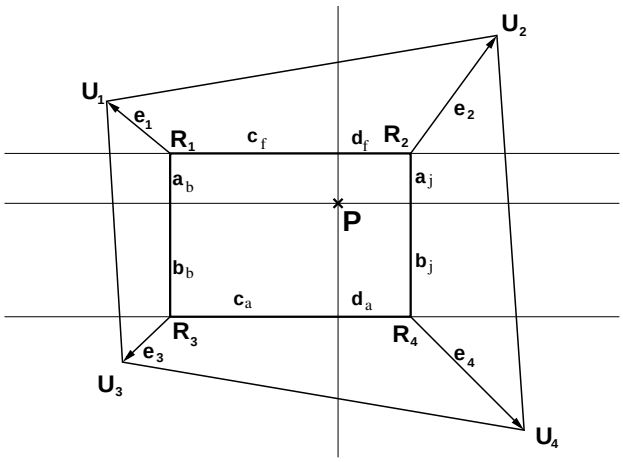


Figure 3. 'Old' (R_1,R_2,R_3,R_4) corner points, their new (U_1,U_2,U_3,U_4) positions, (e_1,e_2,e_3,e_4) displacement vectors and point P of the selected range

4. Energy of the detected γ photons

As a consequence of detection with scintillation crystal, certain signals have to be filtered, because one part of the γ photons participates in scattering where only a part of their energy is absorbed and the direction of their speed changes, sometimes they can induce further scintillations in other positions,

making their localisation theoretically impossible. However, that makes local correction of the energy of events necessary, as the sensitivity of the detectors is not homogenous. The magnitude of the detector signals is proportional to the transmitted energy of the impacted γ photon. As soon as we have resolved the localisation of the detected γ photons, there is nothing to prevent us from preparing the incoming event's detector signal histogram for each crystal needle separately. That will give us the energy-histogram, if we manage to find the proportion coefficient (as the function of position) between detector signal and energy. Thus creating a local energy scale adapted to the inhomogeneous amplification of the detector will become possible, as well as consequent local energy filtering.

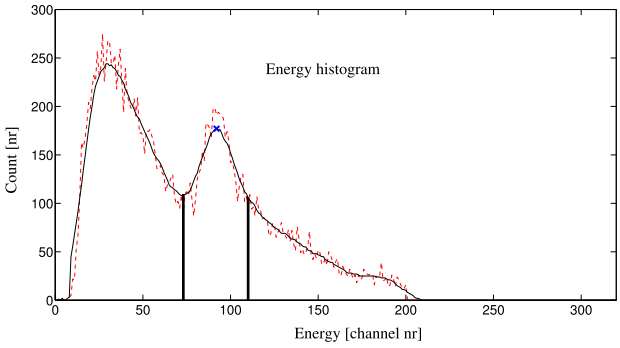
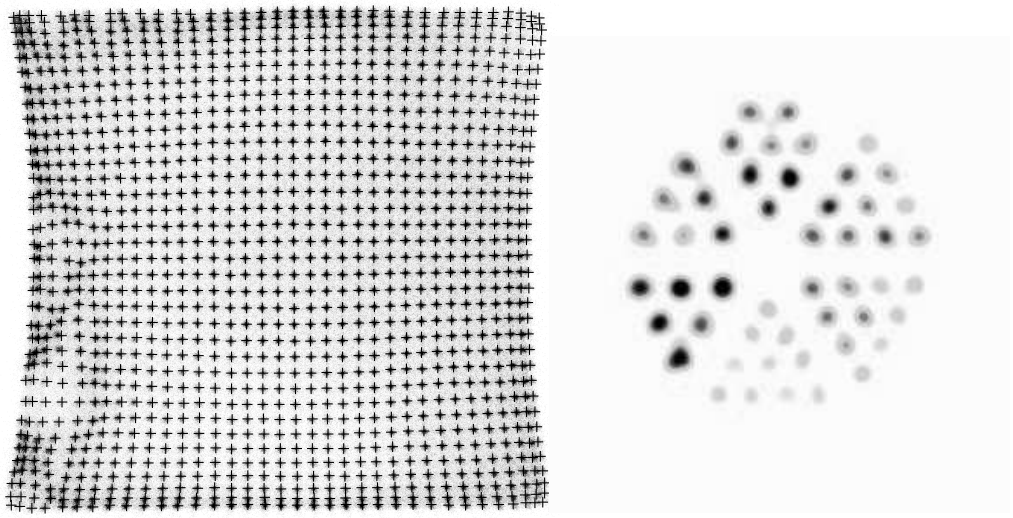


Figure 4. Energy histogram of events detected by a selected (No. 137) crystal needle. The raw curve is shown by a broken line, the curve smoothed with convolution filter by a continuous line; columns show the energy gates and the sign $\times$ shows the photopeak.

Since we allocate the events collected during measurement among 1225 crystal needles, the energy spectrum statistics for a single needle is not very good; they are rather noisy. However, the task is fairly simple, as we search for the right hand peak in the histogram smoothed with the convolution filter, which is actually the photopeak. In case of a properly selected threshold, where the two intervals of the events with impact number higher than the threshold would merge, the limits of the right hand interval will be the events with appropriate energy. Local scaling and filtering can be performed based on the fact that the position of the peak corresponds to 511 keV on the energy scale, and the noise level of the Anger image will subsequently have a significant decrease. Figure 4 shows the energy spectrum of a selected crystal needle, the photopeak and local energy gates.

5. Results

Figure 5/b shows a reconstructed image made by a small animal PET instrument using energy filter and position maps prepared with LUT-QT. In order to examine spatial resolution we have positioned a plastic cylinder (called Derenzo phantom) with holes of different diameters filled with isotopes in the scope of the device, the diameters of the cylinders being 1.6-2.1 mm. Since the images of each cylinder are shown isolated, spatial resolution is smaller than 1.6 mm.



a., **b.,**

Figure 5. Figure **a.,** shows the Flood-field image of a faulty detector module, crosses mark the spots detected by LUT-QT. Figure **b.,** shows the reconstructed image of a Derenzo phantom positioned in the field of view of the small animal PET instrument (activity: ≈ 4 MBq FDG, acquisition time: 30 minutes).

6. Summary

Problem specific methods have been developed to create detector signal position maps used in Mediso's small animal PET instrument. The methods have been implemented in the LUT-QT application. The detector signal – crystal needle index matrix – created from Flood-field images of the single

detectors, determining the centres of light clouds on the adequately filtered images makes quick and accurate determination of the position of incoming γ photons possible. The operation of the code has been successfully tested on real measurement data. Our further aim is to reduce the computation time of the LUT-QT application by optimizing the code and to stabilize the automatic search for light spots, and we would like to develop the possibility of the automatic correction of eventual slow and continuous changes (i.e. because of the outside temperature) in the amplification of the detector signals.

7. Acknowledgements

This work was supported by Mediso Ltd., by the National Office for Research and Technology (NKTH) Grants No. NKFP-A1-2006-0017 (PETCT) and TECH-08-A2-TeraTomo, and by the Hungarian National Research Found (OTKA) Grant No. T69055. We express our thanks for the help and the inspiring and forward-looking debates with our colleagues, László Árvai, László Balkay, Tamás Bükki, Balázs Domonkos, Gábor Jakab, Gábor Németh, Gergely Németh, Gergely Patay and Sándor Török.

REFERENCES

[1] Muehllehner, G. and Karp, J. S.: Positron Emission Tomography. *Phys. Med. Biol.*, **51**, (2006), 117–137.

[2] Lewellen, T. K.: Recent developments in PET detector technology. *Phys. Med. Biol.*, **53**, (2008), 287–317.

[3] Levin, C. S. and Hoffman, E. J.: Calculation of positron range and its effect on the fundamental limit of positron emission tomography system spatial resolution. *Phys. Med. Biol.*, **44**, (1999), 781–799.

[4] Shibuya, K., Yoshida, E., Nishikido, F., Suzuki, T., Inadama, N., Yamaya, T., and Murayama, H.: A healthy volunteer FDG-PET study on annihilation radiation non-collinearity. In *IEEE Nuclear Science Symposium Conference Record*, vol. 3, 2006, pp. 1889–1892.

[5] Doshi, N. K., Williams, C. W., Schmand, M., Andreaco, M., Aykac, M., Loope, M. D., Eniksson, L. A., Melcher, C. L., and Nutt, R.: Comparison of typical scintillators for PET. In *IEEE Nuclear Science Symposium Conference Record*, vol. 3, 2002, pp. 1420–1423.

[6] Imrek, J., Hegyesi, G., Kalinka, G., Molnar, J., Novak, D., Valastyan, I., Vegh, J., Balkay, L., Emri, M., Kis, S., Tron, L., Bükki, T., Szabo, Z., and Kerek, A.: Development of an improved detector module for miniPET-II. In *IEEE Nuclear Science Symposium Conference Record*, vol. 5, 2006, pp. 3037–3040.

[7] Anger, H. O.: Survey of radioisotope cameras. *J. Nucl. Med.*, **5**, (1966), 311–334.


Production Systems and Information Engineering
Volume 5 (2009), pp. 177-188.

GRADIENT-BASED CONSEQUENT OPTIMIZATION OF A FRI RULE BASE

Zoltán Krizsán
University of Miskolc, Hungary
Department of Information Technology
krizsan@iit.uni-miskolc.hu

Szilveszter Kovács
University of Miskolc, Hungary
Department of Information Technology,
Technical University of Kosice, Slovakia
Department of Cybernetics and AI
szkovacs@iit.uni-miskolc.hu

[Received January 2009 and accepted June 2009]

Abstract. The main contribution of this paper is the extension of an existing Fuzzy Rule Interpolation (FRI) method by gradient-based consequent optimization. The targeted FRI method is an application oriented approach, called *FIVE* (Fuzzy Rule Interpolation based on the Vague Environment of the Fuzzy Rule Base [1]). The goal of the consequent optimization is the rule base parameter optimization based on input-output sample data of the modelled system.

Keywords: Fuzzy Rule Interpolation, FRI, FIVE, gradient-based rule optimization

1. Introduction

There are more and more practical applications of Fuzzy Rule Interpolation (FRI) methods appearing in recent literature. Their popularity is based on their ability to handle incomplete fuzzy knowledge representation i.e. 'sparse' fuzzy rule bases. A 'sparse' rule base in this case means a fuzzy rule base, which does not have rules for all the possible observations, in other words, at least one observation may exist which does not lead to an interpretable conclusion applying classical fuzzy reasoning methods (like Zadeh, Mamdani, Larsen, or Takagi-Sugeno). Numerous FRI methods can be found in the literature, and every method has its own advantage. Some of them are very precise, some of them are less precise but their computational complexity is better. In the last

few years FRI based systems have been applied successfully for several fuzzy modeling and control tasks.

For example Johanyák, Parhiban and Sekaran [2] developed fuzzy models for an anaerobic tapered fluidized bed reactor, Johanyák and Szabó [3] used a FRI based fuzzy model for tool life prediction depending on cutting parameters in the case of machining operations.

An application oriented aspect of FRI emerges in the concept of FIVE. The fuzzy reasoning method FIVE (originally introduced in [4] and described in [5], [6] and [1]) was developed to fit the speed requirements of direct fuzzy control, where the conclusions of the fuzzy controller are applied directly as control actions in a real-time system (see e.g. a downloadable and executable code of a real-time vehicle path tracking and collision avoidance control at [7]).

Automatic rule base generation is a hard task and most of the available methods are based on a gradient-free approach. The main problem of these methods is the slow convergence compared to gradient-based methods. On the other hand, for gradient-based parameter optimization there has to be a performance measure which is at least partially derivable with the optimizable (tunable) parameters.

In the following, first the FRI method FIVE will be introduced in more detail with the applied Shepard interpolation, then the paper will suggest a gradient-based optimization method (steepest descent) for the automatic consequent optimization of the FIVE rule base. Finally, we illustrate the method with two sample data sets:

1. the training sample is a simple input-output set of randomly selected data from $y = sin(x)/x$,
2. the training data set contains measured porosity values in a real environment corresponding to specified input values (gamma ray, deep induction resistance and sonic travel time).

2. The Concept of FIVE

The FIVE FRI method is based on the concept of vague environment [8]. Applying the idea of vague environment, the linguistic terms of fuzzy partitions can be described by scaling functions [8] and the fuzzy reasoning itself can be replaced by classical interpolation. The concept of vague environment is based on the similarity or indistinguishability of the elements considered. Two values in a vague environment are ϵ-distinguishable if their distance is greater than ϵ. The distances in a vague environment are weighted distances. The weighting

factor or function is called scaling function (factor) [8]. Two values in the vague environment X are ϵ-indistinguishable if

$$\epsilon \geq \delta_s(x_1, x_2) = \left| \int_{x_2}^{x_1} s(x)dx \right|, \tag{2.1}$$

where $\delta_s(x_1, x_2)$ is the scaled distance of the values x_1, x_2 and $s(x)$ is the scaling function on X.

For finding connections between fuzzy sets and a vague environment the membership function $\mu_A(x)$ can be introduced as an indicating level of similarity of x to a specific element a that is a representative or prototypical element of the fuzzy set $\mu_A(x)$, or equivalently, as the degree to which x is indistinguishable from a (2.2) [8]. The α-cuts of the fuzzy set A are the sets that contain the elements that are $(1-\alpha)$-indistinguishable from a (see Fig. 1):

$$\begin{aligned} 1 - \alpha \geq \delta_s(a, b), \mu_A(x) &= 1 - min\{\delta_s(a, b), 1\} \\ &= 1 - min\left\{ \left| \int_b^a s(x)dx \right|, 1 \right\}. \end{aligned} \tag{2.2}$$

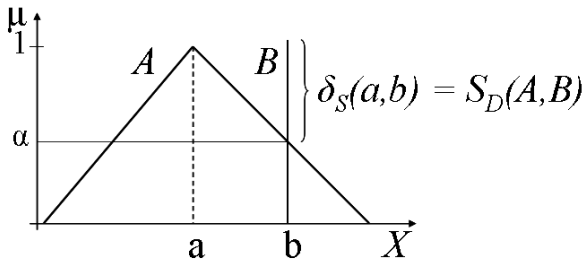


Figure 1. The α-cuts of (x) contain the elements that are $(1-\alpha)$ indistinguishable from a

In this case (see Fig. 1), the scaled distance of points a and b ($\delta_s(a, b)$) is the *Disconsistency Measure (SD)* (mentioned and studied among other distance measures in [9] by Turksen et al.) of fuzzy sets A and B (where B is a singleton).

$$S_D(A, B) = 1 - \sup_{x \in X} \mu_{A \cap B}(x) = \delta_s(a, b) \; if \; \delta_s(a, b) \in [0, 1], \tag{2.3}$$

where $A \cap B$ denotes the min t-norm: $\mu_{A\cap B}(x) = min[\mu_A(x), \mu_B(x)] \forall x \in X$.

Taking into account the most common way of building a traditional fuzzy logic controller where the first step is defining the fuzzy partitions on the antecedent and consequent universes by setting up the linguistic terms and then based on these terms building up the fuzzy rule base, the concept of vague environment [8] is straightforward. The goal of the fuzzy partitions is to define indistinguishability, or vagueness in the different regions of the input-output universes. The vague environment is characterized by its scaling function. For generating a vague environment of a fuzzy partition an appropriate scaling function is needed, which describes the shapes of all the terms in the fuzzy partition. Generally, a fuzzy partition cannot be characterized by a single scaling factor, so the question is how to describe all fuzzy sets of the fuzzy partition with one universal scaling function. For this task, the concept of an approximate scaling function is proposed in [4], [5], [6] as an approximation of the scaling functions describing the terms of the fuzzy partition separately.

3. Shepard Interpolation for FIVE

The main idea of the FRI method FIVE can be summarized in the followings:

1. If the vague environment of a fuzzy partition (the scaling function or at least the approximate scaling function) exists, the member sets of the fuzzy partition can be characterized by points in that vague environment. (These points indicate the positions of the fuzzy terms, while the membership functions are described by the scaling function itself.)
2. If all the vague environments of the antecedent and consequent universes of the fuzzy rule base exist, all the primary fuzzy sets (linguistic terms) compounding the fuzzy rule base can be characterised by points in their vague environment. Therefore the fuzzy rules (built-up from the primary fuzzy sets) can also be characterized by points in the vague environment of the fuzzy rule base. In this case, approximate fuzzy reasoning can be handled as a classical interpolation task.
3. Applying the concept of vague environments (the distances of points are weighted distances), any crisp interpolation, extrapolation, or regression method can be adapted very simply for approximate fuzzy reasoning [4], [5] and [6].

Owning to its simple multidimensional applicability, this paper suggests the adaptation of the Shepard operator based interpolation (first introduced in [10]) for interpolation based fuzzy reasoning. The Shepard interpolation method

for arbitrarily placed bivariate data was introduced as follows [10]:

$$f = g(x,y) = \begin{cases} f_k & if\ (x,y) = (x_k, y_k) \\ & \text{for some } k \\ \left(\sum_{k=0}^{n} f(x_k,y_k)/d_k^{\lambda}\right) / \left(\sum_{k=0}^{n} 1/d_k^{\lambda}\right) & \text{otherwise,} \end{cases} \tag{3.1}$$

where the measurement points x_k, y_k ($k \in [0,n]$) are irregularly spaced in the domain of $f \in R^2 \to R, \lambda > 0$, and $d_k = \left[(x-x_k)^2 + (y-y_k)^2\right]^{1/2}$. This function can be used typically when a surface model is required to interpolate scattered spatial measurements.

The adaptation of the Shepard interpolation method for interpolation based fuzzy reasoning in the vague environment of the fuzzy rule base is straightforward by substituting the Euclidean distances d_k by the scaled distances $\delta_{s,k}$:

$$\delta_{s,k} = \delta_s(a_k, x) = \left[\sum_{i=1}^{m} \left(\int_{a_{k,i}}^{x_i} S_{X_i}(X_i)dX_i\right)^2\right]^{1/2}, \tag{3.2}$$

where S_{X_i} is the i^{th} scaling function of the m dimensional antecedent universe, x is the m dimensional crisp observation and a_k is the abscissa of the prototype point of the k^{th} fuzzy set in the i^{th} antecedent dimension.

Thus, in the case of singleton rule consequents (c_k) the fuzzy rule R_k has the following form:

$$If\ x_1 = A_{k,1}\ and\ x_2 = A_{k,2}\ and \ldots and\ x_m = A_{k,m}\ then\ y = c_k. \tag{3.3}$$

By substituting (3.2) into (3.1) the conclusion of interpolative fuzzy reasoning can be obtained as:

$$y_{(x)} = \begin{cases} c_k & \text{if } x = a_k \text{ for some } k \\ \left(\sum_{k=1}^{r} c_k/d_{s,k}^{\lambda}\right) / \left(\sum_{k=1}^{r} 1/d_{s,k}^{\lambda}\right) & \text{otherwise.} \end{cases} \tag{3.4}$$

4. Gradient-Based Consequent Optimization

The main contribution of this paper is the suggestion of a gradient-based optimization method (steepest descent) for the consequent optimization of the FIVE rule base.
If the performance function is derivable, we can apply the gradient method. Consequent Optimization is based on a set of sample (training) data. The goal of the optimization method is to minimize the squared error E of the fuzzy model.

$$E = \sum_{k=1}^{N} (y_d(x_k) - y(x_k))^2, \tag{4.1}$$

where $y_d(x_k)$ is the desired output of the k^{th} training data and $y(x_k)$ is the output of the fuzzy model applying FIVE (as interference technique), N is the number of the training data points.

The applied steepest descent parameter optimization method modifies the rule consequents based on their partial derivatives to the squared error function E (4.1) in the following manner:

$$g(c_k) = \frac{\partial E(c_k)}{\partial c_k} = \frac{\partial E(c_k)}{\partial y(x)} \frac{\partial y(x)}{\partial c_k} \tag{4.2}$$

$$c_{k_{next}} = c_k - \tau g(c_k), \tag{4.3}$$

where τ is the step size of the iteration and $c_{k_{next}}$ is the next iteration of the k^{th} conclusion c_k.

According to (4.1), (4.2) can be rewritten in the following form:

$$g(c_k) = -2(y_d(x_k) - y(x_k)) \frac{\partial y(x)}{\partial c_k}. \tag{4.4}$$

Applying the Shepard interpolation formula of FIVE (3.4), for the partial derivatives we get the following formulas:

$$\frac{\partial y(x)}{\partial c_k} = \begin{cases} 1 & if\ x = a_k \text{ for some } k \\ \left(1/d_{s,k}^{\lambda}\right) / \left(\sum_{k=1}^{r} 1/d_{s,k}^{\lambda}\right) & \text{otherwise.} \end{cases} \tag{4.5}$$

According to (4.3), (4.4) and (4.5) the next iteration of the k^{th} conclusion c_k can be calculated.

5. Test and Benchmark

5.1. Application Example

The training data of the application example are a simple input-output set of randomly selected data from the y=sin(x)/x function in the domain of [-20, 20]. For demonstration purposes this domain is covered by 13 single input, single output fuzzy rules in the following form (for the k^{th} rule of the rule base):

$$If\ x = A_k\ then\ y = c_k. \tag{5.1}$$

For the initial state of the experiment all the consequents of the fuzzy rules are set to 1 ($c_k = 1, k \in [1, 13]$).

The antecedents (A_k) of the fuzzy rules are fixed and more or less evenly distributed in the domain according to 5.1.

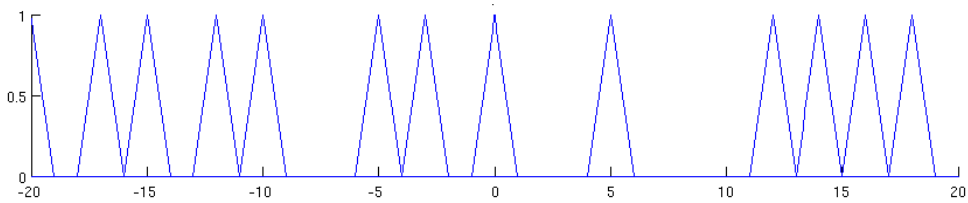


Figure 2. Fixed antecedents (A_k) of the 13 fuzzy rules

Fig. 3 introduces the values of the training data, the conclusions of the initial, and the parameter-optimized fuzzy models. The change of the squared error of the training data and the fuzzy model (4.1) in the function of the iteration steps is illustrated in Fig. 4.

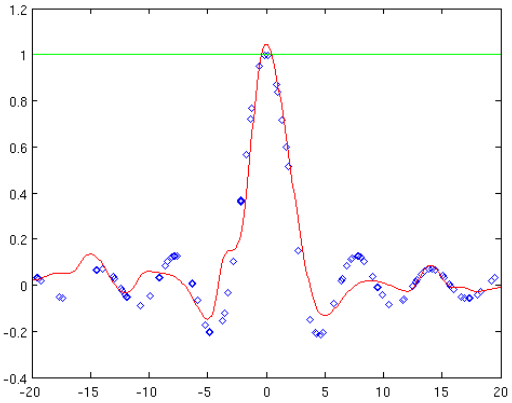


Figure 3. Training data (circles), conclusions of the initial (horizontal line), and the parameter-optimized fuzzy model (curve)

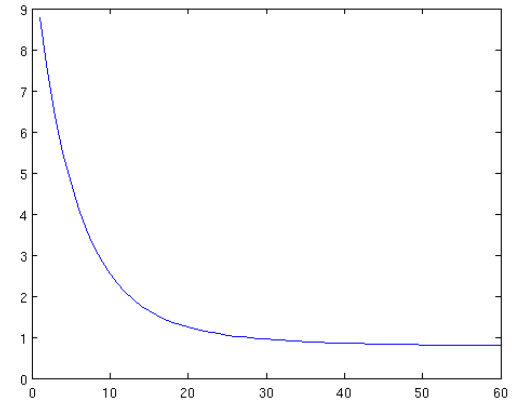


Figure 4. Change of the squared error (4.1) against the iteration steps

5.2. Petrophysical Properties Benchmark

In order to prove the practical applicability of our technique we compared the performance of a fuzzy model generated with the method presented above to some previously published results obtained using other methods for a real world problem taken from the field of petrophysical properties analysis.

The main goal of this example is to compare the optimized FIVE system to the system generated by the RBE-DSS (introduced by Johanyák in [11]) method.

One of the key tasks in the course of the analysis of petroleum oil well data is the prediction of petrophysical properties corresponding to specific input data, i.e. depth values that are different from the original ones used by the experiments. Such properties are porosity, permeability and the volume of clay [12]. The expensive and time-consuming character of data collection from boreholes increases the significance of the prediction. The predicted values help making decisions on the rentability of the exploration of a specific region. The research task of Johanyák was to create a fuzzy model with low complexity that is applicable for the prediction of porosity (PHI) based on well log data described by three input variables. These are the gamma ray (GR), deep induction resistance (ILD), and sonic travel time (DT).

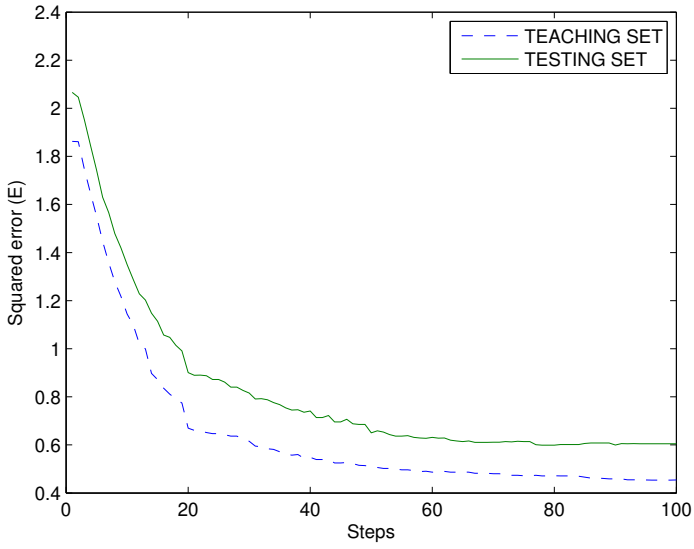


Figure 5. Change of the squared error (4.1) against the iteration steps

The reference system used in the course of the benchmark was developed by Johanyák [11] using RBE-DSS as the model identification technique and LESFRI as the FRI method.

The initial rule base was generated by the RBE-DSS method which is sub-optimal in the case of method FIVE, hence the beginning correlation is not a notable value. With the help of gradient-based optimization (introduced in Section 4) after 100 steps the correlation became good as can be seen in Tab. 1 and Fig. 6, and the performance function value (squared error) decreased as Fig. 5 shows. The optimization results are worse than the results

obtained with RBE-DSS and LESFRI because our method modifies only the consequent, and disregards the antecedent part of the rules.

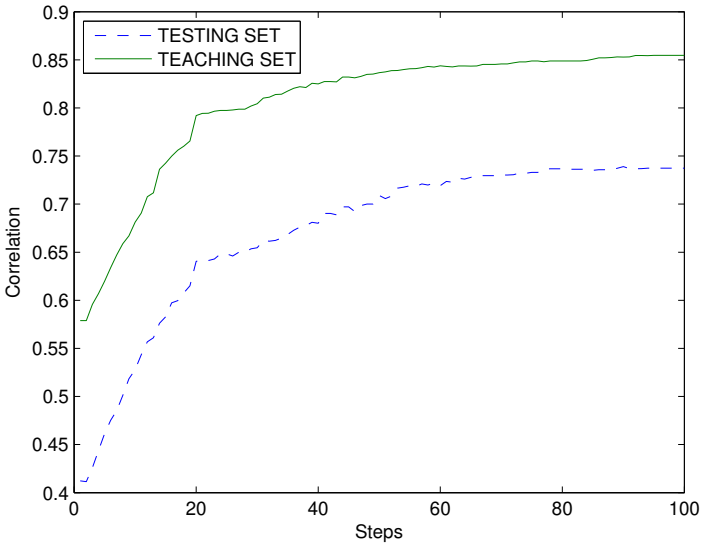


Figure 6. Change of the correlation against the iteration steps

Table 1. Changing of the correlation coefficient against the iteration steps

	beginning	step 50	step 100	RBE-DSS
training	0.5885	0.8343	0.8562	0.934
test	0.4056	0.6935	0.7281	0.890

6. Conclusion

As a first step of automatic rule base generation for FRI methods, this paper suggests a gradient-based optimization method (steepest descent) for the consequent optimization of the FIVE rule base. Consequent optimization is based on a set of sample (training) data. The goal of the optimization method is to minimize the squared error of the training data and the FRI fuzzy model.

Based on the numerical example (introduced in Section 5.2) the correlation is increasing and the squared error is decreasing quickly and it is close to the result obtained with RBE-DSS and LESFRI. The optimization results are worse than the results of RBE-DSS because this method modifies only the consequent, and does not alter the antecedent part of the rules. The chosen

FRI method (FIVE) is rather simple but efficient, serving as a good basis for further improvement of fully automatic FRI FIVE rule base generation which optimizes the antecedent part as well.

Acknowledgements

This research was partly supported by the Hungarian National Scientific Research Fund grant number OTKA K77809 and the Intelligent Integrated Systems Japanese-Hungarian Laboratory.

REFERENCES

[1] Kovács, S.: Extending the fuzzy rule interpolation "five" by fuzzy observation. In *Advances in Soft Computing, Computational Intelligence, Theory and Applications*, Springer, Germany, ISBN 3-540-34780-1, 2006, pp. 445–497.

[2] Johanyák, Z., Parthiban, R., and Sekaran, G.: Fuzzy modeling for an anaerobic tapered fluidized bed reactor. In *Advances in Soft Computing, Computational Intelligence, Theory and Applications*, Timisoara, Romania, ISBN 3-540-34780-1, 2006, pp. 67–72.

[3] Johanyák, Z. and Szabó, A.: Tool life modelling using RBE-DSS method and LESFRI inference mechanism. *A GAMF Közleményei*, pp. 18–27.

[4] Kovács, S.: New aspects of interpolative reasoning. In *Proceedings of the 6th. International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Granada, Spain, 1996, pp. 477–482.

[5] Kovács, S. and Kóczy, L. T.: Approximate fuzzy reasoning based on interpolation in the vague environment of the fuzzy rule base as a practical alternative of the classical CRI. In *Proceedings of the 7th International Fuzzy Systems Association World Congress*, Prague, Czech Republic, 1997, pp. 144–149.

[6] Kovács, S. and Kóczy, L. T.: The use of the concept of vague environment in approximate fuzzy reasoning. In *Fuzzy Set Theory and Applications, Tatra Mountains Mathematical Publications, Mathematical Institute Slovak Academy of Sciences*, vol. 12, Bratislava, Slovak Republic, 1997, pp. 169–181.

[7] Kovács, S.: `http://www.iit.uni-miskolc.hu/~szkovacs`.

[8] Frank, K.: Fuzzy sets and vague environments. *Fuzzy Sets and Systems*, **66**, (1994), 207–221.

[9] Turksen, I. and Zhong, Z.: An approximate analogical reasoning scheme based on similarity measures and interval valued fuzzy sets. *Fuzzy Sets and Systems*, **34**, (1990), 323–346.

[10] Shepard, D.: A two dimensional interpolation function for irregularly spaced data. In *Proc. 23rd ACM Internat.*, 1968, pp. 517–524.

[11] JOHANYÁK, Z.: *Fuzzy szabály-interpolációs módszerek és mintaadatok alapján történő automatikus rendszergenerálás*. Ph.D. thesis, Hatvany József Informatikai Tudományok Doktori Iskola, 2007.

[12] WONG, K. W. and GEDEON, T. D.: Petrophysical properties prediction using self-generating fuzzy rules inference system with modified alpha-cut based fuzzy interpolation. In M. Nikravesh, J. Kacprzyk, and L. A. Zadeh (eds.), *Proceedings of The Seventh International Conference of Neural Information Processing ICONIP*, 2000, pp. 1088–1092.