View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Electronic Thesis and Dissertation Archive - Università di Pisa

UNIVERSITÀ DI PISA

**DIPARTIMENTO DI INGEGNERIA DELL'ENERGIA DEI SISTEMI,
DEL TERRITORIO E DELLE COSTRUZIONI**

**RELAZIONE PER IL CONSEGUIMENTO DELLA
LAUREA MAGISTRALE IN INGEGNERIA GESTIONALE**

# *Design of a Rule Based System to Assign Components to Drive Maps*

---

RELATORI

Prof. Ing. Gino Dini
*Dipartimento di Ingegneria Civile e Industriale*

Ing. Karl Doreth
*Department of Production, Structures and Flows*
*Leibniz Universität Hannover*

IL CANDIDATO

Alessandro Maggio
*Alessandro.maggio1@mail.com*

Sessione di Laurea del 24/09/2014
Anno Accademico 2013/2014
Consultazione Consentita

**Design of a Rule Based System to Assign Components to Drive Maps**
**Alessandro Maggio**


## Sommario

Questo lavoro di tesi s'inserisce all'interno del macroprogetto Leanergie portato avanti dall'università di Hannover nel campo delle macchine utensili, con lo scopo di prevedere già durante la fase di progettazione il consumo di energia della macchina a seconda dello scenario applicativo, in modo da poter pianificare la miglior combinazione dei diversi componenti in termini di consumo di energia. La previsione del consumo di energia, a differenza degli altri metodi basati su modelli matematici e simulativi, è basata su informazioni empiriche acquisite durante il funzionamento della macchina. Questo lavoro si è occupato in particolare di permettere una previsione del consumo di energia tutte le volte in cui non è possibile avere informazioni empiriche su un dato componente, ad esempio quando non è possibile estrarre dati operativi o quando sono presenti nuovi componenti che non hanno mai avuto un applicazione industriale. Per ottenere tali risultati, è stato sviluppato un concetto basato sulla Fuzzy Logic che assegni ad ogni componente privo di informazioni empiriche un diagramma che approssimi il suo funzionamento e il suo consumo di energia nella maniera più precisa possibile.

## Abstract

This thesis work is inserted in the Leanergie macro-project, carried on by the university of Hannover in machine tools field, with the scope of forecasting in the design phase the machine tool's energy consumption. The energy consumption's prevision, differently from other methods that are based on simulation and mathematical models, is based on empirical information acquired when the machine is operating. This work permits in particular to estimate the energy consumption in all those cases when there is no chance of getting empirical information of a component, e.g. when extracting operating data is not possible or when new developed components are present, that have never been in industrial use before. To achieve this results, a fuzzy logic based concept has been developed to assign to each component, without empirical information, a diagram describing its functioning and energy consumption in a manner as precise as possible.

University of Pisa

Faculty of Engineering

Prof. Dr.-Ing. G. Dini

In cooperation with

Institute of production engineering and machine tools

Prof. Dr.-Ing. B. Denkena

Master thesis

# „Design of a Rule Based System to Assign Components to Drive Maps"

Author:

cand. mach. Alessandro Maggio

Student ID: 414972 / 3005290

Supervisor:

Dipl. - Ing. Karl Doreth

## Abstract

As the electric energy cost is increasing, designing efficient operations in machine tools field is becoming fundamental. The choices that mainly affect the energy consumption of the cutting machines primarily occur within the design phase. Thus, forecasting the energy consumption of the different application scenario is extremely important to plan the best components combination in terms of energy consumption.

The Leanergy project wants to be different from all other existing approaches, that are based on complex simulation and mathematical models, using instead empirical information, acquired by measuring energy consumption of machine tools in operating production system.

The concept developed in the present work comes in help for every time that acquired these information and building a component related diagram is not possible. This can happen for many reasons e.g. when there is no chance of capturing operating data as it happens when the scope is to calculate the energy consumption of new developed components that have never been in industrial use before.

To achieve this result, an assignment to each component of the diagram that describes its behaviour better has been created basing on fuzzy logic. Diagrams that have to be used for the assignment are conserved in a database with all the data extracted within running production by means of internal and external sensor installed on the machine tool components.

Before starting with fuzzy logic assignment, application scenario analysis is required in order to figure out all characteristics that, for each component, have to be taken into account for the assignment. These characteristics have to be easily computable and responsible of the component's energy consumption. Through the analysis of the values range that each characteristic can assume, fuzzy membership functions are determined. After that, fuzzy inference rules are built basing on the influence of each feature on the component's energy consumption. Also an evaluation of the best defuzzification methods is presented and by means of an expert dialogue some reflections on the characteristics' choice are discussed. The work ends with a walkthrough of the concept pointing at explaining how using the concept and facing with different cases: it can be used not only to assign directly a diagram to each component but also to understand which diagrams should be mixed to increase the accuracy.

**Title:** Design of a rule based sysem to assign components to drive maps

**Author:** cand. Mach. Alessandro Maggio

Student ID 414972

**Release date:** 15-07-2014

**Supervisor:** Dipl. - Ing. Karl Doreth

# Contents

## List of figures

# List of tables

# List of abbreviations

| Sign | Description | Unit |
| --- | --- | --- |
| CBR | Case Based Reasoning | |
| COG | Center Of Gravity | |
| GE | General Episode | |

# 1 Introduction

Here below there is an introduction to the present work. It starts with the description of the needs that led to this work, it goes on with work's objectives and purposes; in the end the method and the procedure that will be adopted are described and explained.

## 1.1 Motivation

So long as electric energy cost is increasing and in a cutting machine tool it make up 20% of its lifecycle costs, trying to reduce the energy consumption is highly recommended. For this reason, designing efficient machine tools is required.

The design phase of a machine tool and its components becomes particularly relevant as it includes choices on which components using, how linking and combing them; these choices highly condition machine tool energy consumption. Thus, in this phase it is necessary to develop strategies and methods to find an efficient optimum between energy consumption, productivity acquisition costs and operating costs.

Differently from existing approaches whose aim is to predict the energy consumption, that are based on simulation models and thus are hard to parameterize, the research project Leanergie wants to create a concept that can forecast the energy consumption basing on empirical data and information captured within running production. This would permit the comparison of different configuration options and to forecast the energy consumption depending on the product that would be manufactured. For this purpose, a logger module will be developed, which continually captures the energy consumption by means of the machine integrated sensors. That information will be sent back to an energy navigator module, which processes that information in order to forecast the energy consumption of a new designed machine tool.

## 1.2 Objective and Purpose

This work will focus on developing a concept for a rule based assignment of different machine tool components to empirical diagrams. The paramount benefit and utility of this concept is to permit a forecast of consumed energy by a component also every time that operating data of that component are not available and thus, it is impossible creating a component-related diagram. Indeed, using this concept, it is possible to assign a component to the drive map, chosen from all captured diagrams within running production, that explains its behaviour better only basing on its structural and architectural characteristics, which are always known.

## 1.3 Method and procedure

The concept will be created starting from the analysis of the international state of the art of case based reasoning and fuzzy logic. After that, all components belonging to the application scenario of the pre-developed component groups will be studied in order to figure out which characteristics have to be taken into consideration to make each assignment. The concept will carry on with formulation of membership functions, inference rules and defuzzification methods.

The whole work will be evaluated by means of an expert dialog, aiming at characteristics determination, and a walk-through to explain how to face with different situations. In this part, specific significance should be given to the explanation of the possibility of combining more diagrams to increase the accuracy, especially for all cases that there is not a best diagram to describe component’s behaviour but fuzzy logic analysis makes clear that the best solution is to mix two or more diagrams to trace component’s trend.

# 2 State of the Art

In this chapter case based reasoning methods and fuzzy logic will be taken into account as possible methods to evaluate situations with a lack of data. Actual approaches on CBR will be discussed and there will be an explanation of how they work, their applicability and when they can be more useful.

In the second part of the chapter there is an overview of fuzzy logic. After an introduction to fuzzy logic to figure out what it is and why it is so useful to face some situations, its paramount definitions and an explanation of how a fuzzy process can be built are discussed. To help the comprehension of how fuzzy logic works, an example is presented.

In the end of the chapter there is an evaluation of CBR and fuzzy logic pros and cons.

## 2.1 Overview on the actual approaches on Case Based Reasoning Methods

This paragraph contains an overview of Case-Based Reasoning. The paramount goal is to analyze different CBR types and solvers and to have a balance between brevity and expressiveness. To do so, first it is explained how CBR works and how it has to be implemented by describing different CBR types and CBR cycle. Then we briefly review a representative set of systems, next we discuss the connections between CBR and learning. The main part of the chapter analyses the most important issues and problems of the CBR components, such as indexing/retrieval/selection, memory organization, adaptation/evaluation, forgetting, and integration with other techniques.

Case-based reasoning (CBR) is a major paradigm in automated reasoning and machine learning. In case-based reasoning, a reasoner solves a new problem by noticing its similarity with one or several previously solved problems and by adapting their known solutions instead of working out a solution from scratch. In many aspects, case-based reasoning is a problem solving method different from other AI approaches. In particular, instead of using only general domain dependent heuristic knowledge like in the case of expert systems, it is able to use the specific knowledge of concrete, previously experienced, problem situations. Another important characteristic is that CBR implies incremental learning since a new experience is memorized and available for future problem solving each time a problem is solved. Case-based reasoning is a powerful and frequently used way of human problem solving. Case-based reasoning can provide an alternative to rule-based expert systems, and is especially appropriate when the number of rules needed to capture an expert's knowledge is unmanageable or when the domain theory is too weak or incomplete. CBR can work in problem domains where the underlying models used for solutions are not well understood.

### 2.1.1 CBR Types and CBR Cycle

If the way people around us solve problems is analyzed, it is possible to observe case-based reasoning in use all around us. Attorneys are taught to use cases as precedents for constructing and justifying arguments in new cases. Mediators and arbitrators are taught to do the same. Other professionals are not taught to use case-based reasoning, but often find that it provides a way to solve problems efficiently.

Case-based reasoning can mean different things depending on the intended use of the reasoning: adapt and combine old solutions to solve a new problem, explain new situations according to previously experienced similar situations, critique new solutions based on old cases, reasoning from precedents to understand a new situation, warn of possible failures or build a consensus solution based on previous cases. However, these different aspects can be classified into two major types: interpretive (or classification) CBR, and problem solving CBR [LOPE01]. In problem solving CBR, the goal is to build a solution to a new case, based on the adaptation of solutions to past cases. Old solutions can provide almost right solutions to new problems and they can provide warnings of potential mistakes or failures. Problem solving case-based reasoning is useful for a wide variety of problem solving tasks, including planning, diagnosis, and design. In each of these, cases are useful in suggesting solutions and in warning of possible problems that might arise. Generating a solution from scratch is a time-consuming task. In most of domains, however, there is sufficient regularity for a case-based approach to solution generation to provide efficiency. Of course, the problem solver cannot assume that a case-based suggestion is the answer. The case-based suggestion must be validated. Often, however, validation is much easier than generation. In those kinds of domains, case-based reasoning can provide big wins.

The problem solving style is characterized by heavy use of adaptation processes to generate solutions and interpretive processes to judge derived solutions [KOLO92].

In interpretive CBR the key aspect is arguing whether or not a new situation should be treated like previous ones based on similarities and differences among them. The interpretive style uses cases to provide justifications for solutions, allowing evaluation of solutions when no clear-cut methods are available and interpretation of situations when definitions of the situation's boundaries are open-ended or fuzzy.

Interpretive case-based reasoning takes a situation or solution as input, and its output is a classification of the situation, an argument supporting the classification or solution, and/or justifications supporting the argument or solution. It is useful for situation classification, evaluation of a solution, argumentation, justification of a solution, interpretation, or plan, and projection of effects of a decision or plan.

Both styles of case-based reasoning depend heavily on a case retrieval mechanism that can recall useful cases at appropriate times, and in both, storage of new situations back into memory allows learning from experience.

This division, though it is useful to present the field, it is not always clear in practice because many problems have components of both types of CBR and certainly the most effective case-based learners will use a combination of both methods.

Furthermore, many systems use interpretive CBR to evaluate the solutions reached since evaluation is one of the basic operations in any case-based reasoned [LOPE01]. In general, given a case to solve, case-based reasoning involves the following steps :

1. retrieving relevant cases from the case memory (this requires to index the cases by appropriate features);

2. selecting a set of best cases;

3. deriving a solution;

4. evaluating the solution (in order to make sure that poor solutions are not repeated);

5. storing the newly solved case in the case memory.

According to these steps, Case Based Reasoner could be described as a cyclic process comprising "the 4 R’s": Retrieve, Reuse, Revise and Retain, that is:

1. RETRIEVE the most similar previously experienced case or cases

2. REUSE the information and knowledge in the retrieved case(s) to solve the new problem

3. REVISE the solution

4. RETAIN the parts of this experience that are likely to be useful in the future by

incorporating it into the case base.

Figure 1 in the ext page shows the CBR cycle.

Thus we can consider that the quality of a case-based reasoner's solutions depends on four things:

• the experiences it's had

• its ability to understand new situations in terms of those old experiences

• its adeptness at adaptation

• its adeptness at evaluation

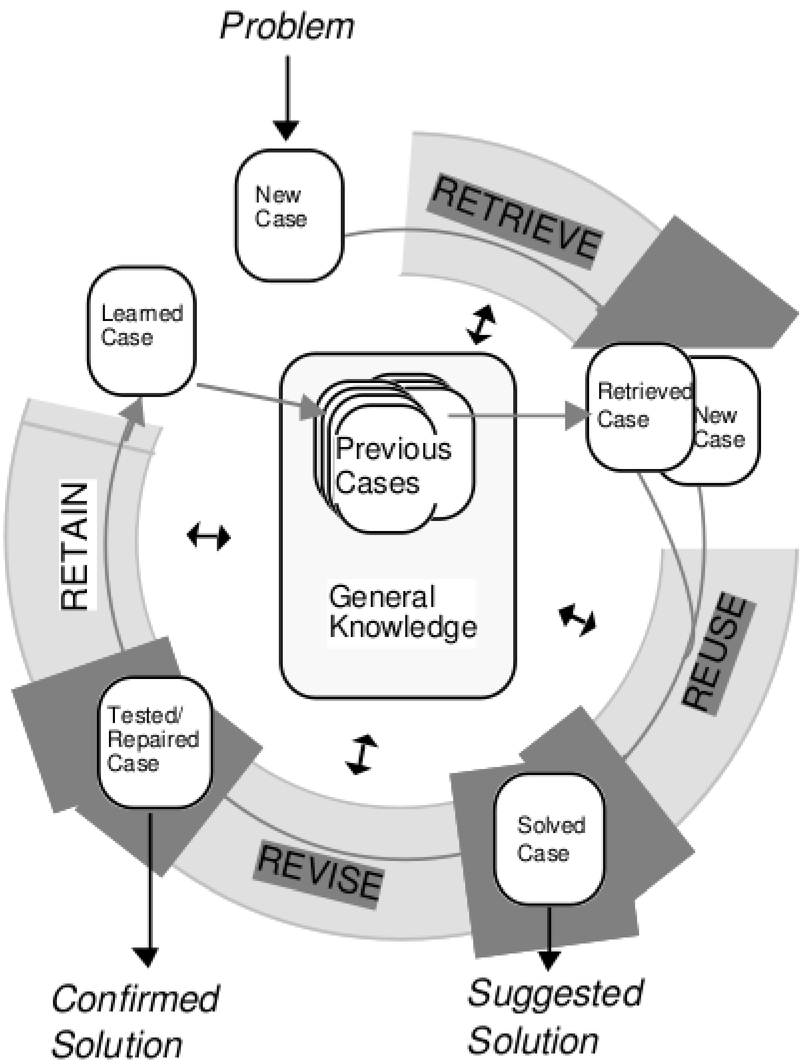



**Figure 1. The CBR Cycle**

Figure 1. CBR Cycle [SCRE13]

The less experienced reasoner will always have fewer experiences to work with than the more experienced one [KOLO92]. But the answers given by a less experienced reasoner won't necessarily be worse than those given by the experienced one if he is creative in his understanding and adaptation. Any programs we write to automatically do case-based reasoning will need to be seeded with a representative store of experiences. Those experiences (cases) should cover the goals and subgoals that arise in reasoning and should include both successful and failed attempts at achieving those goals. Successful attempts will be used to propose solutions to new problems. Failed attempts will be used to warn of the potential for failure.

The second skill, that of understanding a new problem in terms of old experiences has two parts: recalling old experiences and interpreting the new situation in terms of the recalled experiences. The first it is called the indexing problem. In broad terms, it means finding in memory the experience closest to a new situation. In narrower terms, we often think of it as the problem of assigning indexes to experiences stored in memory so that they can be recalled under appropriate circumstances. Recalling cases appropriately is at the core of case-based reasoning.

Interpretation is the process of comparing the new situation to recalled experiences. When problem situations are interpreted, they are compared and contrasted to old problem situations. The result is an interpretation of the new situation, the addition of inferred knowledge about the new situation, or a classification of the situation. When new solutions to problems are compared to old solutions, the reasoner gains an understanding of the pros and cons of doing something in a particular way. One generally see interpretation processes used when problems are not well understood and when there is a need to criticize a solution. When a problem is well understood, there is little need for interpretive processes.

The third, adaptation, is the process of fixing up an old solution to meet the demands of the new situation. Eight methods for adaptation have been identified. They can be used to insert something new into an old solution, to delete something, or to make a substitution. Applying adaptation strategies straightforwardly results in competent but often unexciting answers. Creative answers result from applying adaptation strategies in novel ways.

One of the hallmarks of a case-based reasoner is its ability to learn from its experiences. In order to learn from experience, a reasoner requires feedback so that it can interpret what was right and wrong with its solutions. Without feedback, the reasoner might get faster at solving problems but would repeat its mistakes and never increase its capabilities. Thus, evaluation and consequent repair are important contributors to the expertise of a case-based reasoner. Evaluation can be done in the context of the outcomes of other similar cases, can be based on feedback or can be based on simulation.

### 2.1.2 Case Based Reasoning as a Learning Paradigm

Learning in AI usually mean generalizing through induction or explanation. Learning is in fact inherent to any case-based reasoner not only because it induces also generalizations based on the detected similarities between cases but mostly because it accumulates and indexes cases in a case memory for later use [LOPE01]. The main difference of the CBR approach to inductive learning methods is that CBR emphasizes the semantics of a given domain through similarity based retrieval and case adaptation knowledge. Case-based reasoning as a learning paradigm has several technical advantages:

1. Since each new solved case is stored in memory for later use, instead of deriving new solutions from scratch, a CBR system remembers and adapts old solutions. If such solutions have been adapted in a different novel way or combined in a different way then, when solving another similar case, these circumstances will be remembered rather than re-derived.

2. A case-based reasoner becomes more competent over time, can avoid previously made mistakes, and can focus on the most important parts of a problem first.

3. CBR enables prototype systems to operate with a small initial set of cases and to increase its coverage by storing new cases incrementally.

Since many efforts in CBR address the problem of finding techniques to analyze and select cases, perhaps some of these techniques could be used by the rest of the machine learning community to help in the selection of training instances.

Perhaps the most important advantage of the case-based approach to learning is its affinity to human learning: people take into account and use past experiences to take future decisions.

Instance-based learning (IBL) is a particular case of case-based learning. IBL algorithms store previously categorized examples and use them to classify new inputs by assigning the same classification that was assigned to the most similar previous example. Although they may involve complex indexing, they use a limited representation (feature-value) and do not address case-adaptation.

Case-based learning algorithms have been applied to a large variety of tasks, among them we can point to the following ones: predicting power load levels for the Niagara Mohawk

Power Co.; speech recognition, evaluating oil prospecting sites in the North Sea, knowledge acquisition and refinement, robotic control, molecular biology, architectural design, and medicine.

### 2.1.3 Indexing/Retrieval/Selection (components, issues, problems)

The most basic problems in CBR are the retrieval and selection of cases since the remaining operations of adaptation and evaluation will succeed only if the past cases are the relevant ones.

The retrieval of relevant cases depends on the good indexing of the cases. One way to do it is to fix the indices a priori for a given domain but the problem is the loss in generality. Among the techniques being explored to solve this problem we can mention: Inductive Learning Methods to identify predictive features which will then be used as indices, Instance-Based Learning to learn feature importance, Introspective Reasoning to learn features for indexing and Explanation-Based Techniques to identify relevant features. Explanation-Based Techniques are used to justify the actions of a case with respect to those features known when the case was originally executed. Demonstrably relevant features are generalized to form primary indices. Inconsistencies between the domain theory and the actual case are used to determine irrelevant features. The remaining features are marked as secondary indices that are subject to refinement using Similarity-Based Inductive Techniques. In learning feature importance, each feature is associated with a weight that is adjusted after each prediction attempt during the training process. The adjustment is done by comparing the current case with its most similar stored case. The introspective approach provides the CBR system with an introspective reasoning capability to detect poor retrievals, identify features which would retrieve more adaptable cases and refine the indexing criteria to avoid future failures [LOPE01].

Heuristic search techniques and Qualitative Models are also very promising approaches to the indexing/retrieval problem. Heuristic search techniques are used in a graph containing cases and domain knowledge to look for support for legal arguments. The idea is to narrow the gap between the available indexing scheme and the requirements of arguments through the use of best-first search guided by evaluation functions. A qualitative model of a physical system has been used to derive minimal sets of control parameters relevant to each of the desired inputs in a two stage sewage treatment plant. This approach reduces the number of features used for indexing the cases in a CBR system, which suggests the settings of the control parameters based on past experience in controlling the plant.

Selecting the best case requires being able to find matching cases together. Nearest Neighbor techniques provide a measure of how similar a previous case is to a given problem. In general the match is not perfect because on one hand, the values of the features of the new case and previous cases are not exactly the same and on the other hand there are usually missing values for some or many of the features, therefore the usual approach is to define some similarity metric.

An additional difficulty in the matching problem is that the similar metrics must take into account the different importance of the features. In some situations a weighted similarity measure can be used to account for these differences, but often this is not possible because the importance of some features is context dependent. Usually, the context is represented by the cases already in memory and therefore they can be used to determine which features of the new case are the most important ones. Up to now, practically all the existing similarity measures assume that cases are represented just by collections of feature-value pairs, however we have started to see the need for more structured representations in complex domains and therefore for new approaches to similarity such as graph similarity measures or using domain knowledge to describe declarative biases to guide the retrieval process [GOLD96].

Finally, it is important to mention a very interesting approach that allows to incrementally learn better similarity metrics by interpreting the behavior of the analogical problem solver PRODIGY replaying retrieved cases. To do so, the problem solver provides information about the utility of the candidate cases suggested as similar. This information is used to refine the case library organization and the similarity metric. This process starts with a simple metric that is refined by analyzing the derivational trace produced by PRODIGY.

### 2.1.4 Memory Organization

In this step, the new case is stored appropriately in the case memory for future use. A case is comprised of the problem, its solution, plus any underlying facts and supporting reasoning that the system knows how to make use of, and its outcome. The most important process that happens at this time is choosing the ways to 'index' the new case in memory. Indexes must be chosen such that the new case can be recalled during later reasoning at times when it can be most helpful. It should not be over-indexed, since we would not want it recalled indiscriminately. This means that the reasoner must be able to anticipate the importance of the case to later reasoning. Memory's indexing structure and organization are also adjusted in this step.

This problem shares all of its issues with the first: we must choose appropriate indexes for the new case using the right vocabulary, and we must at the same time make sure that all other items remain accessible as we add to the case library's store [SHIH11].

Good indexing is not enough when the case memory is large. Good organization of the memory is necessary because a simple linear organization, like a list, is very inefficient for retrieval purposes. A much better approach is to have dynamic model with a hierarchical structure, where internal nodes are generalizations of individual cases. The basic idea is to organize specific cases which share similar properties under a more general structure called ‘generalized episode’ (GE) [KOLO92]. A GE contains norms, cases and indices. Norms are features common to all cases, indexed under a GE, and indices are features which discriminate between cases of a GE. An index is composed of an index name and an index value. The entire case memory is in fact a discrimination network where a node is either a generalized episode, an index, or a case. When a new case description input is given and the search for the best matching starts, when one or more features of the input case match one or more features of a GE, the case is further discriminated based on its remaining features. A case is retrieved by finding the GE with most norms in common with the problem description, and the indices under that GE are then traversed in order to find the case which contains most of the remaining problem features. In case storing, when a feature of the case matches a feature of an existing case, a GE is created. The two cases can be discriminated by indexing them under different indices below the GE. If two cases or two GEs end up under the same index, a new GE is automatically created. Hence, the memory structure is dynamic in the sense that similar parts of the two cases are dynamically generalized into a GE.

Even assuming that we have solved the basic problems of retrieval and indexing there is still an additional, somehow unexpected, problem resulting from an uncontrolled growth of the case memory which may result in the degradation of the performance of the system, as a direct consequence of the increased cost in accessing memory.

Existing approaches to this problem include: storing new cases selectively (for example only when the existing cases in memory lead to a classification error) and deleting cases occasionally; and incorporating a restricted expressiveness policy into the indexing scheme, by placing an upper bound on the size of a case that can be matched.

### 2.1.5 Adaptation/Evaluation

A good adaptation of old cases to fit the new case can reduce significantly the amount of work needed to solve it. Since new situations rarely match old ones exactly, however, old solutions must be fixed to fit new situations. In this step, the ballpark solution is adapted to fit the new situation. There are two major steps involved in adaptation: figuring out what needs to be adapted and doing the adaptation [LOPE01].

More recently, the interest in adaptation has increased. For example, quite a few papers have addressed this problem in the recent workshops and conferences on CBR. One adaptation technique uses generalization and refinement heuristics. An example is the plausible design adaptation for design tasks. This adaptation is a process that takes a source concept, a set of constraints and constraint violations, and a set of adaptation transformations and returns a new concept that satisfies the constraints. Many techniques such as interpolation, inductive learning, introspective reasoning are used to adapt old cases. For each type of adaptation strategy, we must also designate the knowledge necessary for its application. Also important in adaptation is methodologies for noticing inconsistencies between old solutions and new needs and, based on that, choosing what should be adapted.

The relations between case adaptation and case retrieval are also being studied. Some of the major issues involved include strategies for evaluating cases and the assignment of blame or credit to old cases.

In this step, the results of reasoning are tried out in the real world. Feedback about the real things that happened during or as a result of executing the solution are obtained and analyzed. Evaluation gives feedback to the case-based reasoner about whether or not the new case was solved adequately. If results were as expected, further analysis is not necessary in this step, but if they were different than expected, explanation of the anomalous results is necessary. This requires figuring out what caused the anomaly and what could have been done to prevent it. Explanation can sometimes be done by case-based reasoning.

This step is one of the most important for a case-based reasoner. It gives it a way of evaluating its decisions in the real world, allowing it to collect feedback that enables it to learn. Feedback allows it to notice the consequences of its reasoning. This in turn facilitates analysis of its reasoning and explanation of things that didn't go exactly as planned. This analysis, in turn, allows a reasoned both to anticipate and avoid mistakes it has been able to explain sufficiently and to notice previously unforeseen opportunities that it might have a chance to reuse. If there is an already-known instance of a similar situation failing, the reasoner must consider whether or not the new situation is subject to the same problems.

Evaluation is the process of judging the goodness of a proposed solution [KOLO92]. Sometimes evaluation is done in the context of previous cases, sometimes it is based on feedback from the world, sometimes it is based on a mental or real simulation. Evaluation includes explaining differences (e.g., between what is expected and what actually happens), justifying differences (e.g., between a proposed solution and one used in the past), projecting outcomes, and comparing and ranking of alternative possibilities. The result of evaluation can be additional adaptation, or repair, of the proposed solution.

### 2.1.6 Range of Applicability and usefulness of Case-Based Reasoning

Case-based reasoning provides a way for people to generate solutions easily, and at the same time it also provides a way for a computer program to propose solutions efficiently when previous similar situations have been encountered. This doesn't mean that causal reasoning is without merit. On the contrary, it must play the role that the medical doctor's logic plays after a solution is proposed.

The causal-model-based system needs to work along with the case-based system to identify changes that must be made in an old solution, to ensure valid adaptations, and to verify proposed solutions.

So case-based reasoning is useful to people and machines that know a lot about a task and domain because it gives them a way of reusing hard reasoning they have done in the past [KOLO92]. It is equally useful, however, to those who know little about a task or domain. Consider, for example, a person who has never done any entertaining yet has to plan a meal with different guest types. His own entertaining experience won't help him. But if he has been to dinner parties, he has a place to start. If he remembered meals he had been served under circumstances similar to those he has to deal with, he could use one of those to get started. For example, if he could generate a list of large dinner parties he has attended, he could, for each one, figure out whether it was easy to make and inexpensive, and when he remembered one, adapt it to fit.

Case-based reasoning is also useful when knowledge is incomplete and/or evidence is sparse. Logical systems have trouble dealing with either of these situations because they want to base their answers on what is well-known and sound. More traditional AI systems use certainty factors and other methods of inexact reasoning to counter these problems, all of which require considerable effort on the part of the computer and none of which seem intuitively very plausible. Case-based reasoning provides another method for dealing with incomplete knowledge [GOLD96]. A case-based reasoner makes assumptions to fill in incomplete or missing knowledge based on what his experience tells him, and goes on from there. Solutions generated this way won't always be optimal, or even right, but if the reasoner is careful about evaluating proposed answers, the case-based methodology gives him a way to generate answers easily.

While the advantages of case-based reasoning are easily evident when an old solution is fairly close to that needed in the new case, case-based reasoning also can provide advantage even if the old solution is far from what is needed. There are two possibilities. Those features of the remembered case that must be ruled out in the new situation can be added to its description and a new case recalled, or the recalled case can be used as a starting point for coming up with a new solution. When there is considerable interaction between the parts of a solution, then even if large amounts of adaptation are required to derive an acceptable solution, it may still be easier than generating a solution from scratch. And the case provides something concrete to base reasoning on. For many people, this is a preferred reasoning style.

## 2.2 Introduction to Fuzzy Logic

Everyday people usually have to confront with concepts that are subjective, difficult to quantify and be classified with certainty. For example, is it possible determine with certainty whether a person is high? Anyone claim that a person of the stature of two meters belongs to the category of high, but a person of 178 cm high? And a 175? According to traditional mathematical logic, a precise limit above that persons can be considered high should be defined: the people who are at least 178 cm are high, the other they are not. But a definition of this type is not representative of the human way of thinking. It is much more natural to think about the whole set of tall people such a set that degrades in a more or less regular manner, from the people who are unequivocally high to reach those who certainly are not. In this case, anyone who is between the two extremes is high, but only partially: someone more, someone else less.

It is therefore clear that the membership of a person to the tall set does not follow the rules of traditional logic, it cannot be expressed easily with a yes or no [CHIN98]. This membership is instead described much better by defining for each person a certain degree of membership, expressing how much the person belongs to the tall set. The same reasoning could be repeated for concepts such as high speed, cheap price, cold climate and so on. The theory of fuzzy logic is based on the definition of "fuzzy" sets, in order to obtain a more realistic representation of variables and concepts which their nature are gradual, not dichotomous.

The fuzzy variables are not numerical, but linguistic, and take their values as high, low, cold, hot. How it will be later shown this is a powerful characteristic of fuzzy logic.

The second fundamental characteristic of fuzzy logic's approach to human way of thinking is its way of representing the reasoning. Usually controls use mathematical formulas and numerical methods to establish the correspondence between input and output variables. The human reasoning is instead characterized by the use of empirical rules, sometimes approximate, due to good sense or experience, but hardly translatable in analytical terms. Also in this case fuzzy theory refers to human decision criteria, using linguistic rules, instead mathematical, to define the way in which the variables influence themselves. For example in driving a car it is needed to perform actions based on the following reasoning type: if the speed is high and close to the obstacle, press hard on the brake pedal if the speed is moderate and the obstacle is located at average distance, press lightly on the brake pedal.

Any driver performs spontaneously and instantaneously reasoning of this kind, while it is much more difficult to quantify precisely the strength in Newton that has to be applied to the brake pedal in correspondence of a certain speed in kilometers per hour and a distance obstacle in meters. The fuzzy linguistic rules are analogous to the empirical descriptive ones here expressed, and do not require the use of formulas or complex analytical models.

Thanks to this way of “thinking”, fuzzy systems behave effectively in those situations in which a person would handle easily, but those are most difficult to manage with analytical methods, as the example of the braking just described [CHIN98].

Furthermore fuzzy logic system based are particularly suitable to work in conditions of uncertainty and disturbances in data acquisition. They adapt well to variable in time or highly non-linear processes, and which are difficult to be represented by mathematical models. Feature of fuzzy logic is the remarkable easy use and understanding, due to its affinity with human reasoning.

### 2.2.1 Fuzzy Logic Use

Managed systems with fuzzy logic are rapidly expanding in many fields. The areas of main use are two, the control systems and expert or decision support systems. Examples of first type applications are the adjustment of humidifiers and air conditioners, the elimination of vibration and focusing for cameras and video cameras, the management of security systems in transportation (such as ABS, intelligent suspension, automatic maintenance of the security distance), the definition of strategies for washing machine according to the load's characteristics. Among second type applications can be mentioned trading decision and risk assessment systems for weather forecasting and geophysical characters and images recognition.

In many fuzzy systems, the input variables are expressed with numerical values (eg . temperature read by a sensor, or a particular decision's cost), and a numerical value for the answers that the system has to provide is required (the power to be delivered to an air conditioner, the magnitude of an investment). In such situations, the need to create an interface between the fuzzy reasoning and the world of numbers arises. At this purpose using fuzzification and defuzzification operations, which convert a numeric (usually called crisp value) to a fuzzy value and vice versa. Between these two phases, the process of fuzzy inference takes place, which equates to inputs an appropriate outputs.

Returning to the fuzzy decision support systems, they offer a conceptual advantage respect to decision-making systems based on operations research or other analytical methods [WYGR13]. When you have to make a choice based on use of analytical methods, we find ourselves faced with a decision space, finite or infinite, containing possible alternatives; then we try to find the alternative that maximizes a certain objective function, respecting at the same time a number of constraints. The objective function allows to order alternatives according to a preference degree, while constraints restrict the space of the alternatives.

Therefore the objective function choice, which must be formulated analytically, and constraints definition result extremely important in determining the process' outcome. In cases in which you wish to achieve more objectives, particularly if they are conflicting, we are bound by this approach limitations. On the contrary, in fuzzy decision-making philosophy, objectives and constraints are handled in the same way, both are expressed through special functions such as membership function, while importance and role, they assume in the system, are determined by linguistic rules. In this way is much easier to combine competing objectives, and provide indication to the system without having to decide whether these have to be used as constraints or as goals. It is shown later how it is possible taking advantage of these opportunities.

### 2.2.2 Definitions

In classical set theory, fixed the universe of discourse X, an element x of X can belong or not to a certain subset A of X. It is possible to define a membership function $\mu_A(X)$, which establishes the link between the elements x and the set A [CELI09], and that can take only two values, zero or one:

$$\mu_A(X)\begin{cases}1 \text{ se x } \in \text{ A}\\ 0 \text{ se x } \notin \text{ A}\end{cases}$$

The fuzzy set theory extends classical theory, introducing the concept of set membership degree. The theory predicts that an element can belong partially to a set, according to a membership function with real values in the interval [0,1]:

$$\mu_A: X \to [0,1]$$

then a fuzzy set A can be defined as the set of ordered pairs composed by elements of X and the corresponding membership function value:

$$A = \left\{\left(x, \mu_A(x)\right) \;\middle|\; x \in X\right\}$$

If the universal set X is continuous the fuzzy set A can be represented with the notation:

$$A = \int \mu_A\,(x)/x$$

Conversely, if X is discrete, one can use the notation:

$$A = \sum_i \mu_A\,(x_i)/x_i$$

In these writings the symbols of integral and summation indicate a union, while the symbol "/" is not a fraction, but the link between the membership value and the element to which it refers.

In the fuzzy terminology, a classical type set with the Boolean membership function is also called "crisp set". There are some operations that allow convert fuzzy sets in the corresponding crisp sets.

We define as “support” of the fuzzy set A the crisp set S (A) consists of all elements of X having a membership degree in A not null :

$$S(A) = \{x \in X \mid \mu_A(x) > 0\}$$

Similarly, it is said “support-α” (α-cut) of A the crisp set $S(A)_{\alpha}$ $or$ $A_{\alpha}$ comprising all the elements of X having a membership degree in A greater than α:

$A_{\alpha} = \{x \in X \mid \mu_A(x) > \alpha\}$

A fuzzy set is called “singleton” if its support is made by only one element of X.

It is defined “core” of a fuzzy set A, the crisp set K(A) consists of all and only elements of X having membership degree A equal to one:

$$K(A) = \{x \in X \mid \mu_A(x) = 1\}$$

The “height” of fuzzy set *A*, *h*(*A*), is the largest membership grade obtained by any element in the set and it is defined as follows:

$$h(A) = sup_{x \in X} \, (\mu_A(x))$$

A fuzzy set is said “normal” if its core contains at least an element of X, and it is called “subnormal” when h(A)<1.

If a fuzzy set A satisfies the following condition it is said “convex”:

$$\forall x, y \in X, \forall \lambda \in [0,1] \Rightarrow \mu_A(\lambda x + (1-\lambda)y) \geq \min(\mu_A(x), \mu_A(y))$$

A fuzzy set A in X normal and convex is called “fuzzy number”.

The fuzzy set characteristics defined above is displayed with the aid of an example of the trapezoidal membership function as shown in the Figure below.

.

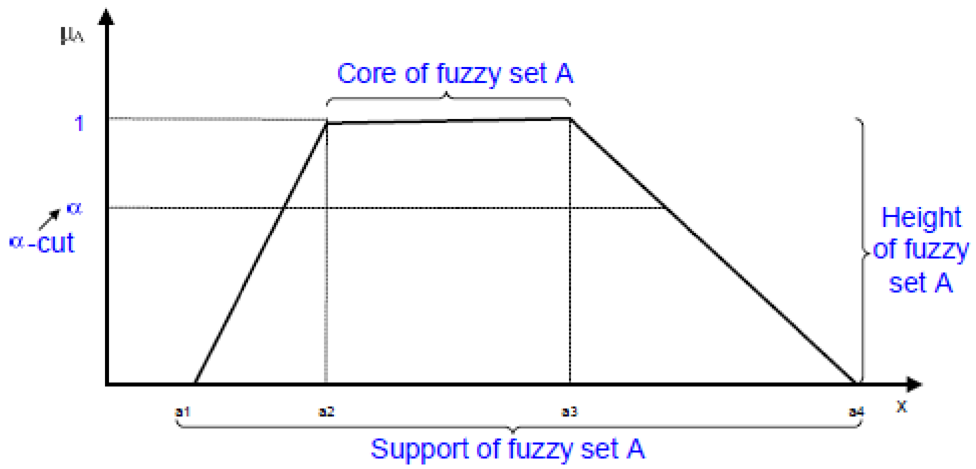
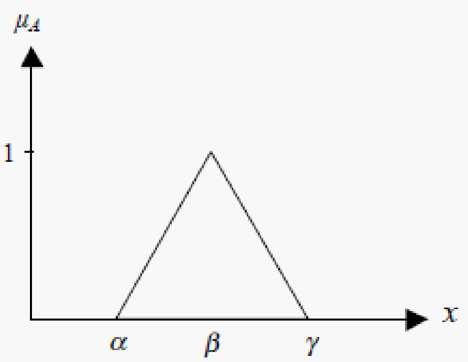



Figure 2. Fuzzy Characteristics [CELI09]

In case of discrete and limited set, the membership function can be numerically represented by couples of value. Otherwise, a function, that permit to calculate an element membership by means of an analytical expression, must be defined.

Depending on the application type it is possible to define different membership function. Below membership functions more used are represented:

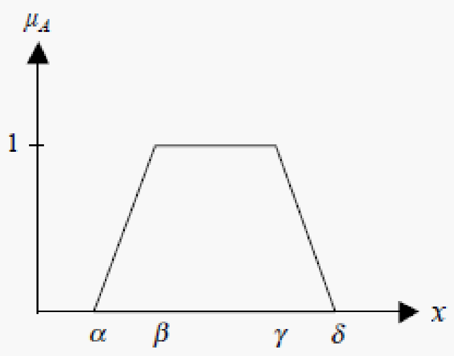
- Triangular Membership Function. It is defined by 3 parameters: the limit values α e γ and the maximum value β.




$$\mu_A(x,\alpha,\beta,\gamma)=\begin{cases} 0 & \text{per } x<\alpha \\ \dfrac{x-\alpha}{\beta-\alpha} & \text{per } \alpha\le x\le\beta \\ \dfrac{\gamma-x}{\gamma-\beta} & \text{per } \beta<x\le\gamma \\ 0 & \text{per } x>\gamma \end{cases}$$

Figure 3 Trinagular Membership Function [CHIN98]

- Trapezoidal Membership Function. It has 4 parameters: the limit values α and δ and the lower and upper values of the interval of maximum β and γ.




$$\mu_A(x,\alpha,\beta,\gamma,\delta) = \begin{cases} 0 & \text{per } x < \alpha \\ \dfrac{x-\alpha}{\beta-\alpha} & \text{per } \alpha \le x \le \beta \\ 1 & \text{per } \beta < x \le \gamma \\ \dfrac{\delta-x}{\delta-\gamma} & \text{per } \gamma < x \le \delta \\ 0 & \text{per } x > \delta \end{cases}$$

Figure 4: Trapezoidal membership function [CHIN98]

- Bell-shaped membership function. It could be get with triangular function parameters by using arcs of parabola instead of straight segments, or with a Gaussian curve by setting its distribution parameters μ and σ.

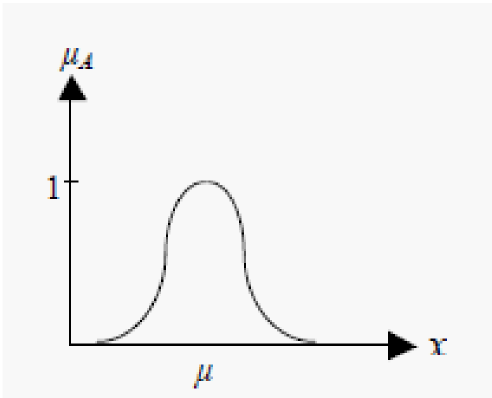



Figure 5: Bell-shape membership function [CHIN98]

For a fuzzy variable , different membership functions must be defined, corresponding to different linguistic values that the variable can assume (e.g. speed moderate, medium, high). The choice of the membership functions is a crucial step in the development of a fuzzy system, considering that it determines the fuzzification and defuzzification process characteristics [DESI04]. The fuzzification allows to calculate membership degree of each numerical value acquired by a input variable to each fuzzy set defined for it. Vice versa, the defuzzification calculates, starting from the result obtained in the fuzzy inference process, a real value for the output variable.

### 2.2.3 Operations on Fuzzy Sets

In classical set theory, four fundamental operations are defined on sets: complement, containment, intersection and union operations. These four operations are also defined for fuzzy sets as standard fuzzy set operations along with many other fuzzy set operators. Two of the most widely used operators are triangular norm (t-norm) and triangular conorm (t-conorm), which are widely accepted and has been widely used in fuzzy sets literature [CHIN98]. Hence a brief definition of t-norms and t-conorms is presented. In addition, the following definitions of standard fuzzy sets are based on De Morgan Triplets with (Min, Max, Complement).

Let entire set of finite elements denoted as $S = \{x_1, \dots, x_n\}$. A fuzzy set *A* can be defined as *A*⊆*S*. Based on the membership function, membership values of fuzzy set *A* can be expressed as:

$$A = \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \cdots + \frac{\mu_A(x_n)}{x_n} = \sum\nolimits_k \frac{\mu_A(x_k)}{x_k}$$

Note that the '+' symbol does not refer to the ordinary addition, thus it is a special representation of discrete fuzzy sets, symbolically.

The following definitions are important to understand possible operations on fuzzy set:

"Equality of fuzzy sets": it can be defined by the equality of membership functions. Let *A,B* ⊆*S* represent two fuzzy sets and equality of two fuzzy sets is defined as:

$$A = B \iff \mu_A(x) = \mu_B(x), \forall x \in S$$

"Inclusion of fuzzy sets": defined by the inequality of membership functions as:

$$A \subseteq B \iff \mu_A(x) \leq \mu_B(x), \forall x \in S$$

It should be remarked that that '*A*⊂ *B*' is defined in fuzzy sets as follows:

$$A \subset B \iff \mu_A(x) \leq \mu_B(x), \forall x \in S \; and \; \exists y \in S \; \mu_A(y) \leq \mu_B(y)$$

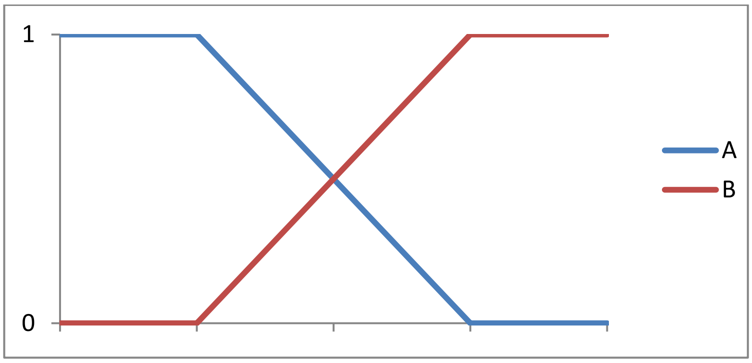



Figure 6: Two fuzzy set

“Union of fuzzy sets”: defined by the maximum of membership functions as:

$$A \cup B: \mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)]$$

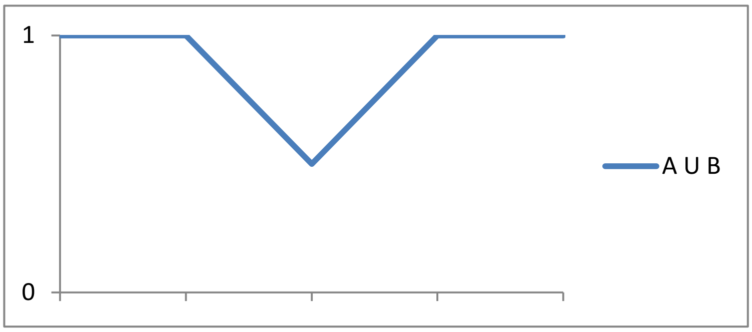



Figure 7: Union of fuzzy set

“Intersection of fuzzy sets”: defined by the minimum of membership functions as:

$$A \cap B: \mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$$

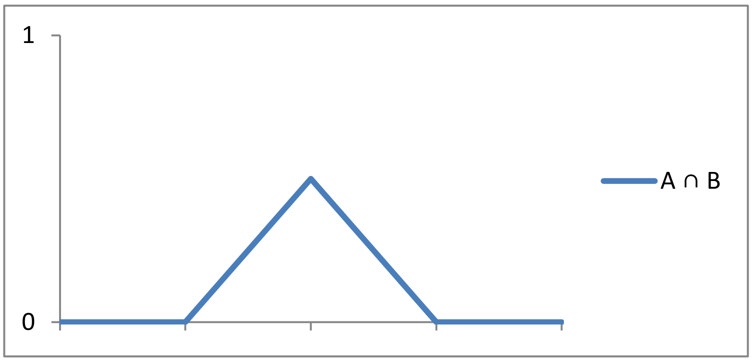



Figure 8: Intersection of fuzzy set

"Complement of fuzzy set": the complement of fuzzy set $A$, $A^c$ is defined as follows:

$$\mu_{A^c}(x) = 1 - \mu_A(x)$$

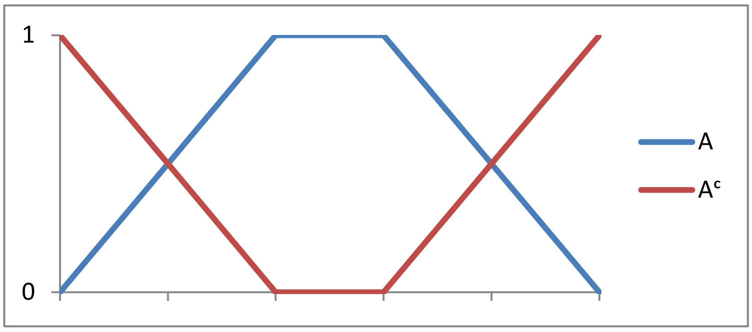



Figure 9: Complement of fuzzy set

Logical operations have been the focus of considerable discussions over the evolutionary history of fuzzy logic. Originally, "*min*" and "*max*" functions were presented to model logical conjunction and disjunction. These functions obviously generalize traditional Boolean operators, but it was immediately recognized that they were not the only possible functions

### 2.2.4 Fuzzy Rules and Implication

Human knowledge is often empirical due to experience, difficult to quantify and codify. Fuzzy logic is able to translate this type of knowledge in formal constructs, directly processable by a computer [DESI04].

Fuzzy knowledge-base system is constituted by two basic components: variables membership functions and the set of inference fuzzy rules. Fuzzy rules represent the transition between empirical knowledge and their numerical computation. These rules are qualitative, expressed in natural language, but at the same time they are a formal description of the system. In fact, once put them in relation with the membership functions, they provide a purely numeric system model, on which a computer can work too.

A fuzzy rule is usually expressed with a construct of if-then, and it can have one or more antecedent and one or more consequents. A rule with an antecedent and a consequent therefore takes the following form:

*if x is A then y is B*

A rule of this type is equivalent to the implication: Fuzzy A → B. Fuzzy implication is not an usual logical implication with the corresponding truth table, but rather a fuzzy relation on sets A and B. Fuzzy implication is, in effect, a true report, and we can therefore write:

$$\mu_{A \to B}(x, y) = \mu_A(x) \zeta \mu_B(y)$$

With ζ of implication operator. It is important to have different forms of fuzzy implications, in order to choose the one that is best suited to the system on which you are working [CELI09].

The two most commonly used functions of implication, namely fuzzy min and product implications, are also the most simple and they, as an operator of implication, use respectively the minimum triangular rule and the product algebraic one, they are:

$$Fuzzy \min implication \ \ (Mamdani) \ \mu_{A \to B}(x, y) = \min(\mu_A(x), \mu_B(y))$$

$$Fuzzy \ product \ implication \ (Larsen) \mu_{A \to B}(x, y) = \mu_A(x) \cdot \mu_B(y)$$

A fuzzy rule therefore corresponds to a fuzzy relationship. Generally, fuzzy rules are with multiple inputs and an output (MISO - multiple input, single output) or multiple inputs and multiple outputs (MIMO - multiple input, multiple output). It is needed to introduce connection operations between the different antecedents and between different consequents. In order to consider together different inputs, it is used the AND and OR connective, while the connective ALSO is used to indicate that a rule has multiple outputs. The general rule form (for example, the k-th in the knowledge base, with n inputs and m outputs) will be the following:

$$if \ x_1 \ is \ A_{k1} \ and \ \dots and \ x_i \ is \ A_{ki} \ or \ \dots \ or \ x_n \ is \ A_{kn} \ then \ y_1 \ is \ B_{k1} \ also \ \dots also \ y_m \ is \ B_{km}$$

It must be said that the OR connective is rarely used in practical applications.

AND connective and the implication operator THEN are both translated mathematically in triangular norms [WYGR13], so we can understand how from their joint use it could get different relationship types. Considering only the two most used (minimum and algebraic product), there are four combinations, or four different forms for the resulting report. The two most common forms are those that use the same norms for intersection and implication.

### 2.2.5 Fuzzy Inference Process

The inference process allows to derive output magnitudes by applying inference rules to known input values. Recalling that a fuzzy rule is a relation between antecedent and consequent it is understandable how the most suitable operation to act as a conjunction between rules and input functions is their composition.

The inference rules most used are two adopting respectively Mamdani (called max-min composition) and Larsen implication (called max-product composition).

The rules are expressed and represented below:

$$\mu_{B'}(y) = max_X\left(\min\left(\mu_{A'}(x), \mu_A(x), \mu_B(y)\right)\right) \forall (x, y) \in X \times Y$$

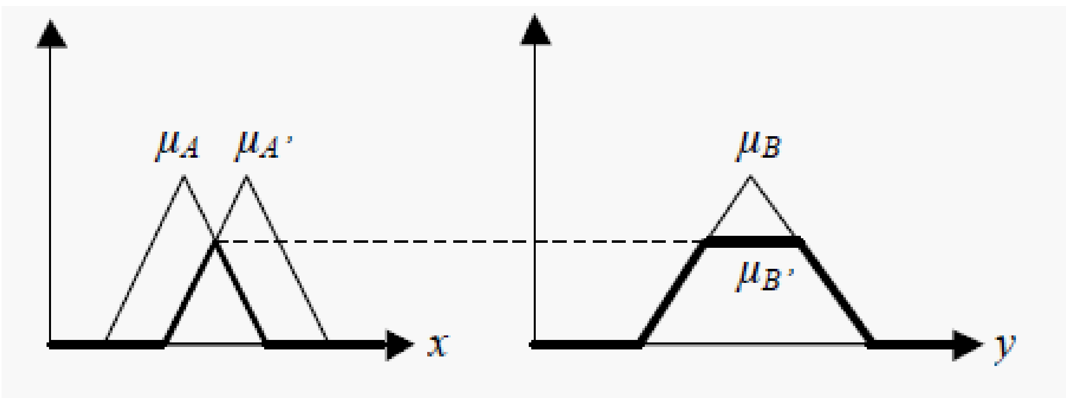



Figure 10: Mamdani operator [CELI09]

$$\mu_{B'}(y) = max_X\left(\mu_{A'}(x) * \mu_A(x) * \mu_B(y)\right) \forall (x, y) \in X \times Y$$

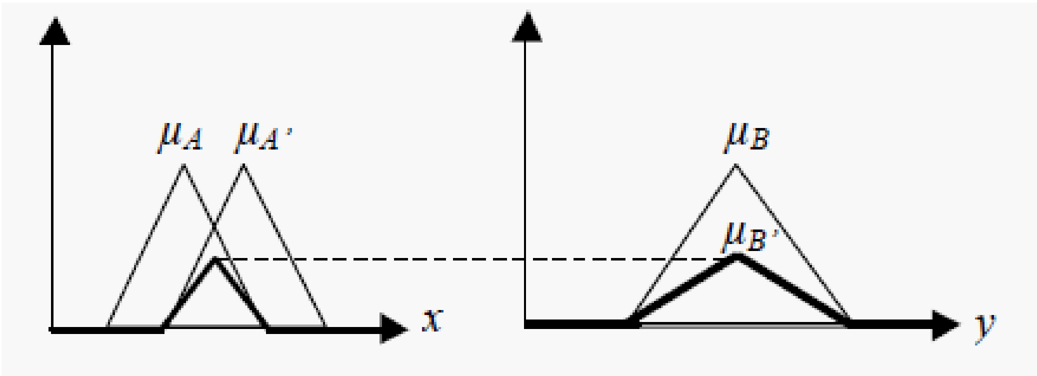



Figure 11: Larsen operator [CELI09]

In both cases, the logical process is the following: to start it is required to intersect the input set A' with the antecedent set A of the implication, taking into account that the term "intersection" can have different meanings depending on the chosen standard. Then the utmost respect to x of the obtained function is evaluated, obtaining thus the input truth value compared with the rule, also known as activation rule degree. This numerical value is intersected with the implication consequent B, assigning in this way to B an importance based on input A' truth degree. The final membership function, defined in Y, is that of the output set B'. Using a fuzzy product implication, the set B is multiplied by the antecedent truth degree, while the use of a fuzzy min implication results in set's truncation, which becomes a trapezoid having as maximum value the antecedent truth degree.

Very often, normalized fuzzy singleton are used as inputs, those are fuzzy sets in which only one element of the universe has membership degree equal to one and all others are zero. When input parameters are punctual singleton inputs introduction does not detract meaning from process but it simplifies the calculations greatly.

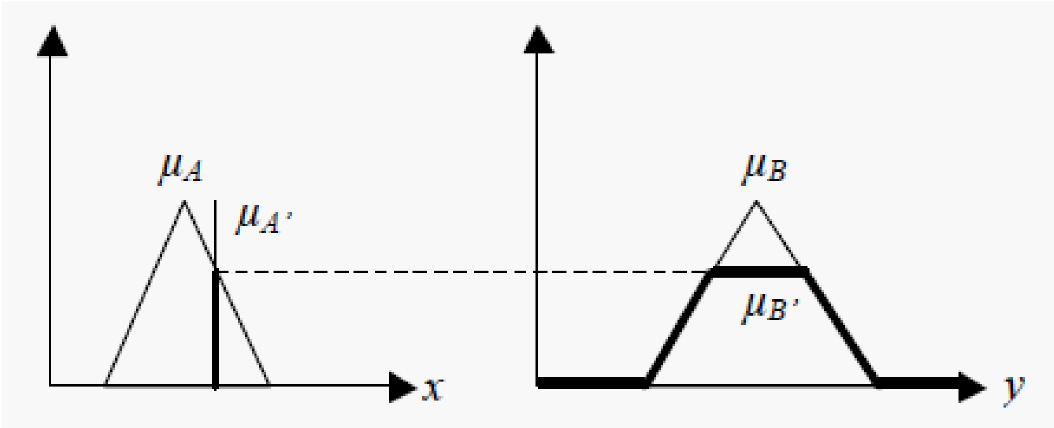



Figure 12: Fuzzy singleton [CELI09]

### 2.2.6 Fuzzy Logic Control Applications

The idea of *fuzzy control* or *fuzzy logic control*, in other words, was formulated and developed in the 1970s by L. A. Zadeh and E. H. Mamdani. It became probably the most successful application of fuzzy logic and, generally, of fuzzy sets and their methodology.

The conventional approach to control requires a sufficiently adequate (mathematical) model of the process or system to be controlled [CHIN98]. In many cases, however, such a model is not available as it is completely unknown or, say, its construction would be too costly or too time consuming. The essence of fuzzy control is to base control on rules rather than on a model. The control rules are obtained in a verbal and - thus - imprecise, fuzzy form from experienced operators or experts who know how to control the process or system. Those fuzzy rules are then translated into the language of linguistic variables and fuzzy conditional statements. Since fuzzy rules guarantee a successful manual control, we can expect the same from an automatic control based on them.

This simple and natural idea has turned out to be very fruitful and has led to a successful automation of many control processes which resisted the conventional approach. Speaking more generally, that idea has opened the door to fuzzy rule-based modeling in a lot of areas of applications, and to fuzzy rule-based systems. Below it is presented the fundamentals of fuzzy control and fuzzy rule-based systems limiting the analysis to those elements which are relevant to the thesis task.

### 2.2.7 Computational Approach to Fuzzy Rules

A suitable and convenient language in which fuzzy control rules and, generally, any fuzzy rules can be formulated is that of fuzzy conditional statements from:

IF α = A THEN β = B

The "IF" part as well as the "THEN" one may be compound. For instance, they may be conjunctions or disjunctions of some subconditions as in:

IF α = *high* AND β = *medium* THEN γ = *very low*

If necessary, one can use fuzzy conditional statements in an extended form, namely

IF α = *A* THEN β = *B* ELSE γ = *C*

How to carry out this expression? In other words, it is asked which value should be assigned to the output linguistic variable β if α = *A*. Moreover it must be said that there are many cases about which value the variable α can assume, for instance A could be precisely determined or it could be a range of values (A = about x). Below it is considered that the measures are not affected by error and thus, input variables can assume only precise values.

Concerning the operator's choice, it will be used one of the most common operator in practice:

$a \rightarrow b = a\, \boldsymbol{t}\, b$,

called an engineering implication operator, where ***t*** denotes a t-norm and $a, b \in [0, 1]$.

Its two cases are especially important in applications:

• *Mamdani operator*: $a \rightarrow b = a \wedge b$

• *Larsen operator*: $a \rightarrow b = a \cdot b$

### 2.2.8 Fuzzy Controller

Here it is presented in a simple way the original approach to fuzzy control formulated by Mamdani. The general structure of a fuzzy controller is then the following:

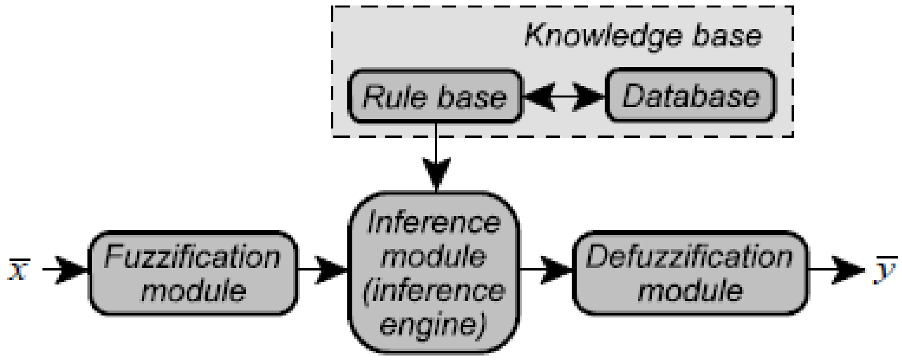



Figure 13: General structure of fuzzy controller [CELI09]

Actually, also presents a more general thing: a fuzzy rule-based system. Besides control issues, applications of such systems contain expert systems, decision support, modeling and simulation, image processing, etc. Although the further presentation nominally refers to fuzzy controllers, it thus remains valid for arbitrary fuzzy rule-based systems. Here it follows a brief description of components shown in Figure.

•$\bar{x} = (x_1, \dots, x_n)$ is a vector of measured or observed crisp values of *n* input variables.

• The *rule base* contains a collection of rules in the form of fuzzy conditional statements. Since their “IF” parts contain some fuzzy sets rather than precisely specified values, the *fuzzification module* transforms $(x_1, \dots, x_n)$ into membership values in those fuzzy sets.

• The very inference process is performed by the *inference module* using the rule base and the apparatus of approximate reasoning.

• Since the results of inference are again some fuzzy sets, the *defuzzification module* transforms them into crisp values forming the output vector $\bar{y} = (y_1, \dots, y_n)$. Clearly, a feedback may be involved and, then, that vector is used to generate a new input vector.

• The database stores all data which are necessary for a proper functioning of the fuzzification, inference, and defuzzification modules. In particular, it stores the fuzzy sets involved in the rules contained in the rule base.

• Finally, the knowledge base is composed of the rule base and database.

Presenting the workings of a fuzzy controller, it is convenient to focus on the basic case of two input variables and one output variable ( $n$ = 2, $k$ = 1), which can be easily extended to more variables. Moreover, for clarity, it is shown a concrete and familiar example of house heating, depending on air temperature and fuel price. A strictly formal and general presentation would not be readable enough.

The input variables *air_temperature* (*AT* ) and *fuel_price* (*FP*), and the output variable *heating_level* (*HL*) will be treated as linguistic variables. Assume that their values are

*AT*: *cool*, *cold*, *frosty*,

*FP*: *low*, *medium*, *high*,

*HL*: *low*, *medium*, *high*.

Interpretations of those values by means of fuzzy sets are presented in the figure 14.

The real variable $p$ denotes price per fuel unit, whereas $h$ represents heating understood, say, as heating valve opening level. Interpretations of values of each linguistic variable must always be overlapping as in Figure 14.

What one has to do next is to define the rule base. The case of house heating is familiar and there is not need to ask experts. It suffices to use a commonsense approach. The following list of rules seems to be reasonable:

IF *AT* = *cool* AND *FP* = *low* THEN *HL* = *medium*

IF *AT* = *cold* AND *FP* = *low* THEN *HL* = *high*

IF *AT* = *frosty* AND *FP* = *low* THEN *HL* = *high*

IF *AT* = *cool* AND *FP* = *medium* THEN *HL* = *low*

IF *AT* = *cold* AND *FP* = *medium* THEN *HL* = *medium*

IF *AT* = *frosty* AND *FP* = *medium* THEN *HL* = *high*

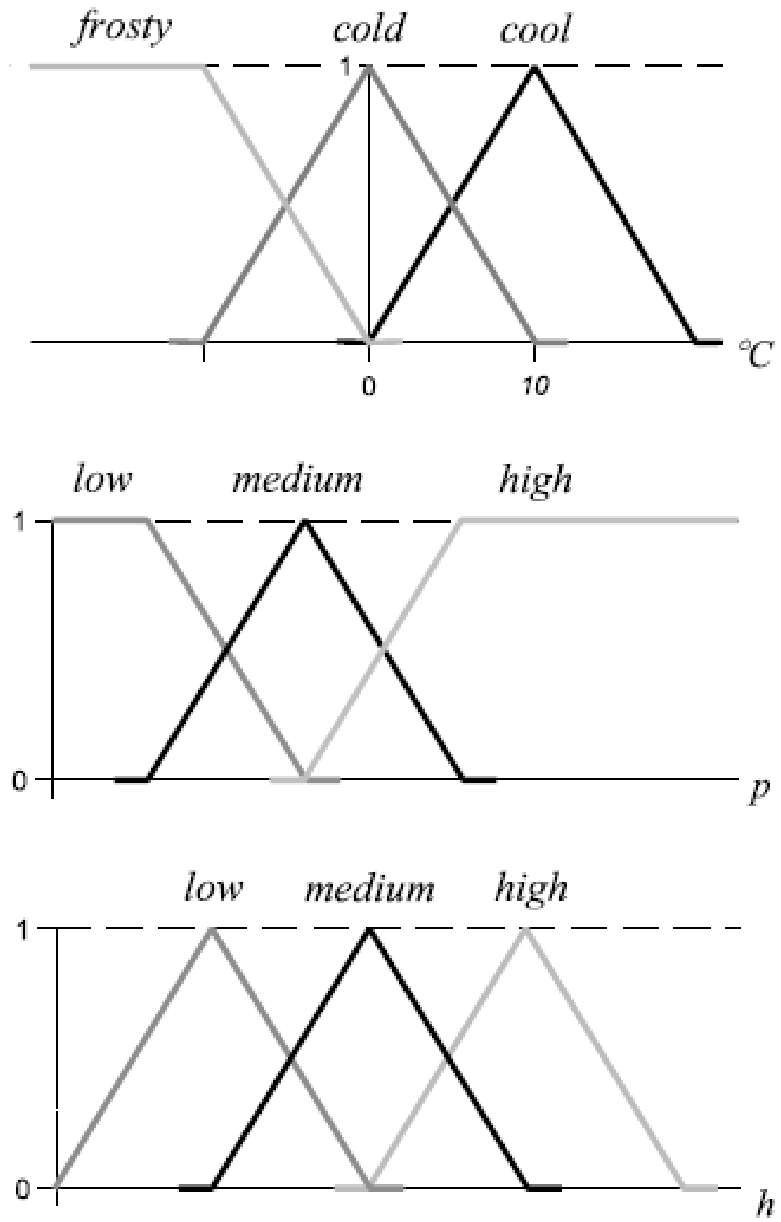



Figure 14: Membership functions for each characteristic [CELI09]

IF *AT* = *cool* AND *FP* = *high* THEN *HL* = *low*

IF *AT* = *cold* AND *FP* = *high* THEN *HL* = *low*

IF *AT* = *frosty* AND *FP* = *high* THEN *HL* = *medium*

Rewriting them in a table form is more convenient. This is shown in the table 1.

| *FP↓ AT→* | *cool* | *cold* | *frosty* |
|---|---|---|---|
| *low* | *medium* | *high* | *high* |
| *medium* | *low* | *medium* | *high* |
| *high* | *low* | *low* | *medium* |

Table 1: Fuzzy rules and their output

Speaking generally, the rules in a rule base cannot be contradictory, i.e. the base cannot contain rules with identical “IF” parts and different “THEN” parts. The number of rules should be “suitable”: not too small and not too large. Some potential rules are usually skipped whenever they refer to impossible or unimportant combinations of values of input linguistic variables. The rule base in Table involving all combinations of linguistic values of AT and FP is thus redundant in this context. For instance, fuel prices usually rise with the coming of freezing weather and, therefore, at least the rules

IF *AT* = *frosty* AND *FP* = *low* THEN *HL* = *high*

and

IF *AT* = *cool* AND *FP* = *high* THEN *HL* = *low*

could be removed from the base. The corresponding two fields in Table would then be blank.

Below the workings of the fuzzy controller previously described is presented, equipped with the rule base defined in Table. Generally, the rules from a rule base are grouped with respect to their “THEN” parts and the rules within each group are then performed one after the other.

Assume that the current air temperature and fuel price are 4 ˚C and 35 € per fuel unit, respectively. This combination of temperature and price, as any other, fulfils to a degree the conditions in the “IF” part of each rule as it can be seen in the figures below.

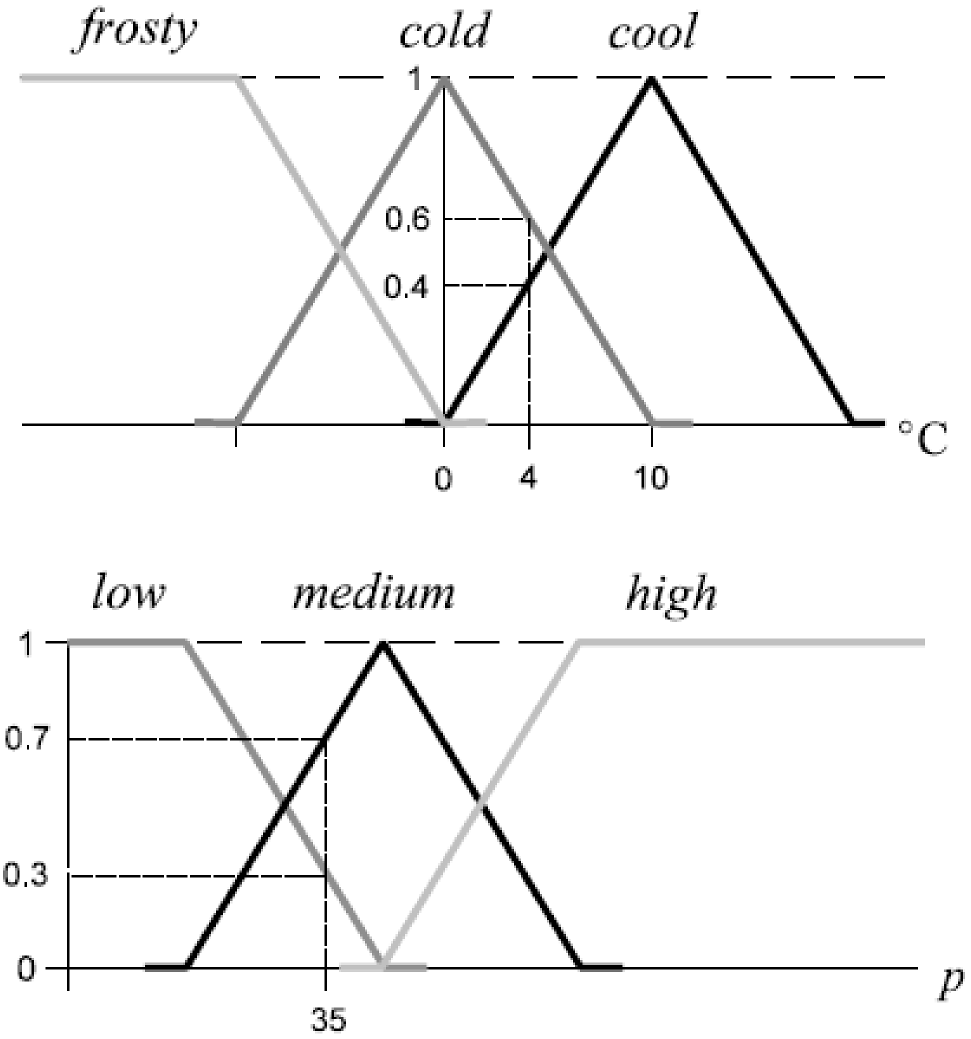



Figure 15: Measured values and their membership degrees in different fuzzy sets of temperatures (top) and prices (bottom) [CELI09].

The fuzzy controller moves on to the execution of the first rule in the base, namely

IF *AT* = *cool* AND *FP* = *low* THEN *HL* = *medium*

Step 1: Fuzzification - a transition from measured values to membership degrees.

Looking at the above figure, we get *cool*(4) = 0.4 and *low*(35) = 0.3, i.e. it is cool to degree 0.4 and the current fuel price is low to degree 0.3.

Step 2: Inference. One uses the inference mechanism whose computational side has been described earlier. The key is the degree of fulfillment of the "IF" part, namely *cool*(4) ***t*** *low*(35) = 0.4 ***t*** 0.3 with a t-norm ***t***. This degree thus equals

$$0.3\ for\ t = \wedge, and\ 0.12\ for\ t =\ t_a$$

According to theory previously presented, the final result of executing the rule is a fuzzy set $\underline{B}^{(1)}$ in the universe of heating levels, where

$$\underline{B}^{(1)}\ (h) = (0.4\ t\ 0.3) \rightarrow medium(h)$$

For simplicity, assume that *t* = ∧ has been chosen, i.e

$$\underline{B}^{(1)}\ (h) = (\ 0.3) \rightarrow medium(h)$$

It is possible to look at $\underline{B}^{(1)}$ with various operator →,

- → = Mamdani operator: $\underline{B}^{(1)}\ (h) = (\ 0.3) \wedge medium(h)$ .

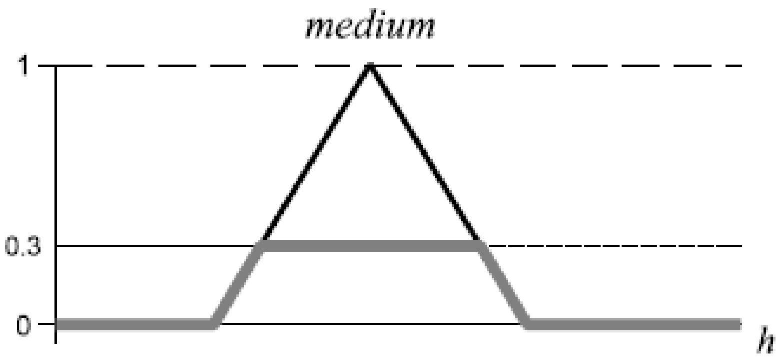



Figure 16: Output using Mamdami operator [CELI09]

- → = Larsen operator: $\underline{B}^{(1)}\ (h) = (\ 0.3) \cdot medium(h)$ .

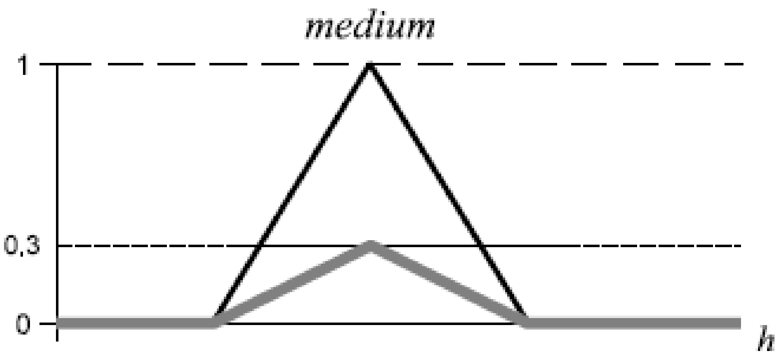



Figure 17. Output using Larsen operator [CELI09]

The fuzzy controller moves on to the next rule with “THEN *HL* = *medium*”:

IF *AT* = *cold* AND *FP* = *medium* THEN *HL* = *medium*

Step 1': Fuzzification. Now (see Figure)

*cool*(4) = 0.6 and *medium*(35) = 0.7.

Step 2': Inference. Again, choose ***t*** = ∧. The degree of fulfillment of the “IF” part is

*cold*(4) ∧ *medium*(35) = 0.6 ∧ 0.7 = 0.6.

The result of executing the second rule is thus a fuzzy set $\underline{B^{(2)}}$ with

$$\underline{B^{(2)}}\,(h) = (\,0.6) \rightarrow medium(h)$$

For instance, we thus have (see Figure)

- → = Mamdani operator: $\underline{B^{(2)}}\,(h) = (\,0.6) \wedge medium(h)$.
- → = Larsen operator: $\underline{B^{(2)}}\,(h) = (\,0.3) \cdot medium(h)$ .

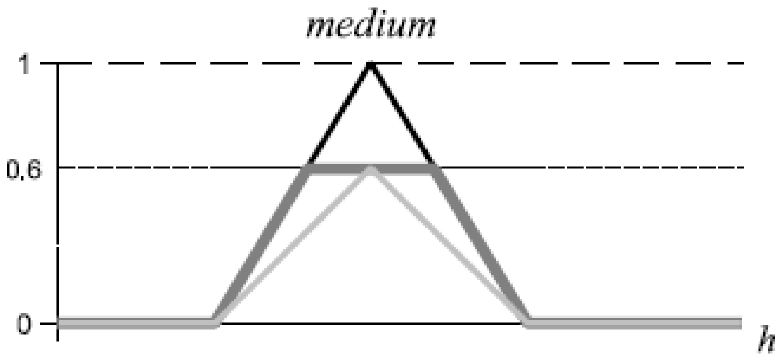



Figure 18: Mamdani (dark grey line) and Larsen (light grey) operator [CELI09]

Moving on to the rule

IF *AT* = *frosty* AND *FP* = *high* THEN *HL* = *medium*

one gets

*frosty*(4) ***t*** *high*(35) = 0 ***t*** 0 = 0 for each ***t***. So, it produces a fuzzy set $B^{(3)}$ with

$$\underline{B^{(3)}}\,(h) = (\,0) \rightarrow medium(h)$$

i.e. $B^{(3)} = 1$ for each engineering implication operator →

The results of the three rules with "THEN *HL* = *medium*" are then aggregated into one fuzzy set. One assumes that the rules in a rule base are connected by OR. So, the aggregation of $\underline{B^{(1)}}\,(h), \underline{B^{(2)}}\,(h)\ and\ \underline{B^{(3)}}\,(h)$ is the sum

$$\underline{B^{(1)}}\,(h) \cup \underline{B^{(2)}}\,(h)\ \cup\ \underline{B^{(3)}}\,(h)$$

The focus is on the case of using ***t*** = v and Mamdani operator, which is fundamental for practice. The sum of the three set then collapses to (see Figures)

$$\underline{B^{(1)}}\,(h) \cup \underline{B^{(2)}}\,(h)\ \cup\ \underline{B^{(3)}}\,(h) =\ \underline{B^{(2)}}\,(h)$$

Exactly the same way of doing will be applied to the rules containing "THEN *HL* = *low*" and those with "THEN *HL* = *high*". Aggregating all the resulting fuzzy sets one obtains a final fuzzy set *H* illustrated below.

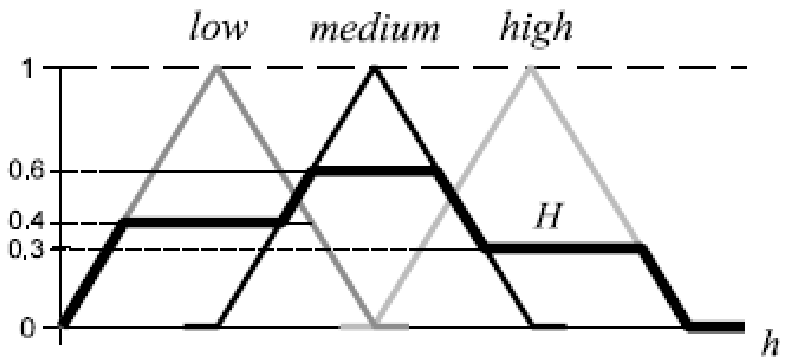



Figure 19: Aggregated result of execution of all the rules [CELI09]

Step 3: Defuzzification – returning from membership degrees to crisp values.

According to the Figure, each value of the output parameter *h* (each setting of the heating valve, in other words) fulfils to a degree all the rules treated *en bloc*. So, different values of *h* are satisfactory to different degrees *H*(*h*). Our fuzzy controller has to choose one of them, *h**, which is the best or most suitable in a sense. This defuzzification can be performed in various ways. For instance, one can take as *h** the smallest or the greatest *h* for which *H* attains its maximum. Alternatively, *h** can be defined as the arithmetic mean of those two values. An especially important and frequently used variant of defuzzification is however the COG method. Here *h** becomes the first coordinate of the center of gravity of the figure determined by *H* and the *h*-axis.

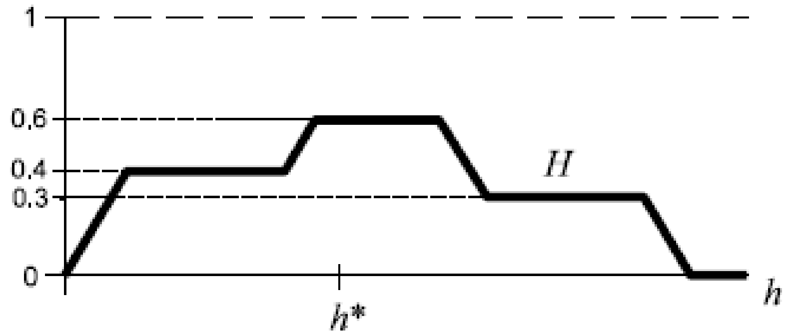



Figure 20: Result of defuzzification by the COG method [CELI09]

Now *h** is now a compromise, a point of balance taking into account various possible values of *h* which are satisfactory to different degrees.

## 2.3 Pros and Cons of the actual approaches on Case Based Reasoning Methods and Fuzzy Logic

After the presentation of case based reasoning and fuzzy logic, analyzing them to figure out when they are more useful, which adavantages they lead and when a higher attenction is required, is necessary. Whereas for CBR is easy to understand its advantages and especially its disadvantages; for fuzzy logic is more difficult to do it as its results depend particularly on the quality of analysis and process. How it is explain below, fuzzy logic does not present clear disadvantages but there are some aspects and steps that require high caution.

### 2.3.1 Case-Based Reasoning

Case-based reasoning provides the following advantages for a reasoned:

- Case-based reasoning allows the reasoner to propose solutions to problems quickly, avoiding the time necessary to derive those answers from scratch. While the case-based reasoner has to evaluate proposed solutions like any reasoner does, it gets a head start on solving problems because it can generate proposals easily. There is considerable advantage in not having to redo time-consuming computations and inferences. This advantage is helpful for almost all reasoning tasks, including problem solving, planning, explanation, and diagnosis.

- Case-based reasoning allows a reasoner to propose solutions in domains that are hard to understand completely[KOLO92]. Many domains are impossible to understand completely, often because much depends on unpredictable human behavior, e.g., the economy. Others nobody understands yet, like how some medications and diseases operate. Other times, it is simply being in situations hard to understand well, but in which it is need to act anyway, e.g., choosing which graduate students to accept into a program. Case-based reasoning allows to make assumptions and predictions based on what worked in the past without having a complete understanding.
- Case-based reasoning gives a reasoner a means of evaluating solutions when no algorithmic method is available for evaluation. Using cases to aid in evaluation is particularly helpful when there are many unknowns, making any other kind of evaluation impossible or hard. Instead, solutions are evaluated in the context of previous similar situations. Again, the reasoner does his/her evaluation based on what worked in the past.
- Cases are particularly useful for interpreting open-ended and ill defined concepts [SHIH11]. Indeed, this is one use attorneys put cases so extensively, but it is also important in everyday situations. Case-based methodology for interpretation can be more accurate that a generalization-based method when classifications are ill-defined. A Case-based reasoning system is significantly more capable and accurate than classification systems based on more traditional classification methods.
- Remembering previous experiences is particularly useful in warning of the potential problems that have occurred in the past, alerting a reasoner to take actions to avoid repeating past mistakes.
- Cases help a reasoner to focus its reasoning on important parts of a problem by pointing out what features of a problem are the important ones. What was important in previous situations will tend to be important in new ones. Thus, if in a previous case, some set of features was implicated in a failure, the reasoner focuses on those

> features to insure that the failure will not be repeated. Similarly, if some features are implicated in a success, the reasoner knows to focus on those features. Such focus plays a role in both problem solving and interpretive case-based reasoning. In interpretive case-based reasoning, justifications and critiques are built based on those features that have proven responsible for failures and successes in the past. In problem solving, a reasoner might attempt to adapt his solution so that it includes more of what was responsible for previous successes and less of what was responsible for failures.

Of course, there are also pitfalls in using cases to reason. A case-based reasoner might be tempted to use old cases blindly, relying on previous experience without validating it in the new situation[LOPE01]. A case-based reasoner might allow cases to bias him/her/it too much in solving a new problem. And, often people, especially novices, are not reminded of the most appropriate sets of cases when they are reasoning.

Moreover using cases and their related solutions do not assure that those are the best possible, thus it is needed to be very careful in both phases, during the first case analysis and specially during the final validation.

Other problems could show up during the retrieving process. Here it could happen that two cases need to be judged similar even though they share few surface features [KOLO92]. One way to deal with this problem is to use more than just the surface representation of a case for comparison. Cases must also be compared at more abstract levels of representation. The issue we must address is which of the abstract ways of representing a case are the right ones to use for comparisons.

Another problem is that a new situation and a case might share some derivable features while not sharing surface features. In predicting who will win a battle, for example, the ratio of defender strength to attacker strength is predictive but neither defender strength nor attacker strength by itself is. Cases need to be judged similar based on the ratio (a derived feature) rather than the individual values (surface features). The issue here is to come up with a way of generating derived features for cases in an efficient way. It is necessary to have a guidance in generating derived features because some are expensive to derive, and even if all were cheap, it would be expensive to generate all possible derived features of a case. And of course it is much better if it is possible to derive fast retrieval algorithms for massive libraries of cases. These problems comprise what it is called the indexing problem. Broadly, the indexing problem is the problem of retrieving applicable cases at appropriate times (despite all the problems cited above). In general, it has been addressed as a problem of assigning labels to cases that point out under which conditions each case can be used to make useful inferences.

Researchers are working on specifying what kinds of indexes are most useful, designating vocabularies for indexes, creating algorithms and heuristics for automating index choice, organizing cases based on those indexes, searching memory using those indexes, and choosing the best of the retrieved cases. The tension between using indexes to designate usefulness and direct search while at the same time not allowing them to dominate what can be recalled is one of the most important issues in case-based reasoning.

Finally other important problems are the concepts of uncertainty, imprecision and incompleteness. These are problems that do not concern a single phase but pervade the whole CBR reasoning process. Indeed uncertainty and imprecision are present in the semantics of abstract features used to index the cases, in the evaluation of the similarity measures computed across these features, in the determination of relevancy and saliency of similar cases, and in the modification rules used in the solution adaptation phase.

Incompleteness is also present in the partial domain theory used in indexing and retrieval, in the (usually) sparse coverage of the problem space by the existing cases, and in the description of the problem. A solution is to use probability theory to model the uncertainty associated with the main assumption of CBR that to similar problems correspond similar solutions. An application of that solution is to apply a Bayesian network modeling at CBR [SHIH11]. This model uses two networks, one for ranking categories, and another for identifying exemplars within categories. This view leads to the notion of modeling similarities by conditional probabilities. Probability theory cannot however model imprecision easily. Fuzzy logic could provide better techniques to deal with imprecision.

### 2.3.2 Fuzzy Logic

As far as fuzzy logic's pros and cons first of all it must be said that most of advantages that fuzzy logic presents, are linked to its skill of processing similar to human reasoning. Rather than talking about fuzzy logic's cons, it would be better talking about pitfalls or, even better, difficulties. Below both aspects are analyzed in detail.

Fuzzy Logic’s advantages:

- Fuzzy logic calls into question and changes binary logic's concept, according to which, predicates can assume only two states: true or false. Every calculator is based on this type of logic, but it is easy to estimate how much this logic might be inaccurate and not adherent to reality, that has multiple aspects. "crisp" reasoning works only with concepts of the equal and the different (not equal or not included) [DESI04]. Unlike the binary logic, fuzzy reasoning introduces the notion of "similarity degree" as a membership of a concept to a default prototype that works as comparison function; fuzzy sets have no rigid boundaries but they include a limit value's variation, that is like a subjective judgment approximation of each person. An object's membership degree to a fuzzy set can assume any value between [0,1], unlike a traditional set, which is restricted to the limit values 0 or 1. Fuzzy theory traces a curve between opposite objects, between A and not-A. If only a few information are available, it is possible to turn the vague notions into fuzzy set curves. More information are available, more accurate and more adherent to reality, the curve will be.
- Ability to operate under conditions of uncertainty, imprecision and incompleteness typical of reality, through linguistic variables's usage [WYGR13]. Usually controls adopt mathematical formulas and numerical methods to establish the correspondence between input and output variables, instead human, and likewise expert reasoning adopts emperical rules, coming from experience and good sense, many times very simple but hard to translate in analytical terms. Fuzzy theory moves in this direction, approaching human decision criteria by the use of linguistic rules, instead of mathematical ones, in order to define the way in which the variables influence each others. Thus, fuzzy systems work well in every situations easy manageable by a person but these result very difficult to deal with analytical methods

- Use of qualifiers. Defining modifiers or edges, it is allowed a greater adherence to natural language [CHIN98]. In fact often, in order to explain a concept it is easier to use such expression like "very", "pretty", "enough"; these qualifiers help to make a rigid process, as it can be a calculator's one, more flexible. Generally expressions like "very" move toward a higher membership degree, expressions like "pretty" move toward a lower one.
- Conceptual simplicity. Characteristic of fuzzy logic is the easy use and understanding, due to its affinity with human reasoning.
- Adaptability to various types of problem. Furthermore fuzzy systems are ideally suited to work in conditions of uncertainty in data acquisition and in data lack. They can be adapted to time-varying or highly non-linear processes.

Like it is said in the introduction, more than disadvantages in this case it would be better talking about difficulties and problems, as for the execution of the whole fuzzification-inference rules-defuzzification process there are not real rules and regulations but only some guidelines. Thus, it is necessary to be very careful in all the points where choices are required for determination of which alternative has to be chosen. Steps that present greatest pitfalls are analyzed below, all these points have in common that there are not all-purpose rules for managing the choice but it is needed to rely on the experience of the user, his skills, systematic tests and on a careful starting analysis.

- The membership function selection is one of the most important process choice, if the function's determination is wrong, the whole execution will be miscalculated. The difficulty is represented by the fact that the entire process should reflect an expert's behavior who analyzes the current situation. In addition, it is needed to determine the function type that represents best the linguistic variables (triangular, trapezoidal or bell-shaped).
- Determination of the inference rules. First of all it is necessary to pay attention not to have conflicting rules, in other words rules with the same "if" and different "then". Moreover in this phase, depending on the antecedents it is required to choice which consequents should "go out", even here the inference rules should reflect an expert's reasoning, no matter how simple the compilation of the rules could be, a special at-

tention is needed in this stage as it determines the inference engine behavior, so it is crucial in the results development.

- Defuzzification. The inference result is a fuzzy set obtained by the union of the single inference result, so the choice of which method has to employ for the result transformation from a fuzzy value into a single crisp value, representative of the inference, becomes essential. While for the other choice types there is no preferred method, but every case has to be analyzed in details, here perhaps the center of gravity method involves less risk and greater reliability, it is obtained by calculating the x-coordinate of the geometric figure's center of gravity that represents the fuzzy set's outcome.

## 2.4 Summary

In the this chapter CBR and fuzzy logic methods have been analyzed and debated. In the next chapter  the analysis moves to the application scenario concerning machine tools environment. Components that have to be considered within this subject, in order to forecast machine tools energy consumption will be defined. For each component diagram bases and other considerations have to be discussed.

After that every component will be analized in order to understand which configurations can be found and how they work in machine tools environment. This will lead to figure out which component's characteristics should be considered as paramount responsibles in determining a component diagram trend.

# 3 Analysis of Application Scenario

In this chapter first of all the application scenario is defined. This can be considered as the group of components that has to be taken into account as machine energy consumption paramount responsible. After that group is defined, a separated analysis for each component is required. The scope of this analysis is to understand for each component which types are more utilized in machine tool environment, their functioning and their tasks. That analysis will allow hereafter to figure out which components characteristics must be considered to estimate their energy consumption and their diagrams trend.

## 3.1 Defining the Application Scenario (5 pre-developed component groups)

Before to study which component characteristics are more important to assign components to drive maps, the analysis of the application scenario is needed. In fact the paramount objective of the macro project is to forecast the energy consumption of machine tool depending on an application scenario. Thus the further analysis will be based on the previous work that has defined which groups must be included in the application scenario and thus which are the reference diagrams. Furthermore this work explains which bases diagrams have. Other important considerations will be taken into account in the next paragraph.

The previous work [DORE13] has revealed that the most significant diagrams for the energy consumption forecasting are:

- Drive diagram: this diagram group contains all component responsible of motion start. The diagram bases are torque and speed and the output is the necessary electric power for the component running. For every drive type there must be a different diagram.
- Machine diagram: here components that transmit and often transform motion from the drives to the guides or to the spindle are evaluated. Friction losses has to be considered, thus in this diagram type the bases are the output speed and torque in order to calculate the requested input torque, that is the torque drive has to procure.
- Cutting diagram: in this group the cutting process is analyzed, in particular what happens in the contact area between cutting tool and workpiece. What has to be calcu-

lated is the emitted torque, that changes in relation to the material removal rate and the tool rotational speed. Therefore these variables will have a trend particularly different depending on workpiece type, tool and requested operation.

- Drive heat diagram: in this case, emitted thermal power has to be considered, these can lead to energy inefficiency and machine precision loss, the information are taken from external sensors; however diagram bases are torque and speed and thus the assignment is the same as in drive diagram instance.
- Cooling diagram: here the task is to calculate cooling system electric energy consumption. Cooling system has to guarantee both cutting oil's and hydraulic oil's correct temperature. Energy consumption depends on temperatures reached by different motion transmission mechanisms and by tools during the operations on the workpiece.

All those notions are summarized in the table 2.

| Diagram Type | Drive Diagram | Gear Diagram | Cutting Diagram | Drive Heat Diagram | Cooling Diagram |
|---|---|---|---|---|---|
| Considerations | Drive Loss | Friction Losses Static Weight | Cutting Force | Emitted Thermal Power | Cooling Power |
| Base | Torque Speed | Speed Moving Weight | Material Removal Rate Speed | Torque Speed | Thermal Power |
| Source | Drive Manufacturer | Energy Logger | Energy Logger | Energy Logger Via External Sensors | Energy Logger Via External Sensors |
| Diversity of Diagram per | Drive Type | Axis Type | Work Piece Cutter Force Type | Drive Type | Cooling System |

Table 2: Application scenario

## 3.2 Analysis of Considered Components

Once the application scenario has been determined, a separated analysis for each component is required. For each component operating conditions, working environment, tasks and further considerations will be discussed.

### 3.2.1 Analysis of the drive

Drives are probably the most important components of machine tool. There are two main type of drive: feed drives and spindle drives. The firsts are used to position the machine tool components carrying the cutting tool and workpiece to the desired location; hence their positioning accuracy and speed determine the quality and productivity of machine tools. The latter provide the mechanical motion to the spindle-tool system. They rotate the shaft to create relative motion between tool and workpiece and provide power and torque to the cutting tool in order to perform operations such as drilling, milling, and grinding.

There are some motors that are suitable for both applications, and other, depending on their characteristics, that are more advisable only for one [LOPE09]. Each motor presents its own movement mechanism and they present different performance, everyone with own pros and cons. Thus, the motor type can be considered as the first characteristic for comparison. Below motors most widely used in machine tool are presented and described.

- Brushed Direct Current Motors. This kind of motors use internal commutation to create the motion. Until a determined speed, they can maintain the torque constant at its maximum level. There are two versions of this motor type, each one has a different preferred application: one has a permanent magnet rotor and windings on the stator and its preferred application is for guides movement since low feeds but with a high torque are requested and in this motor type there is not the possibility to control field current and thus to have high speeds; the other one has a wound stator and it is used especially for the spindle since it can provide also high speeds even if with a torque decrease. However torque reduction is not a problem in machine tool since at high speeds only finishing operations are requested and they do not need high torques; whereas a problem might be the heat production due to the contacts with brushes and windings are internal in the motor.

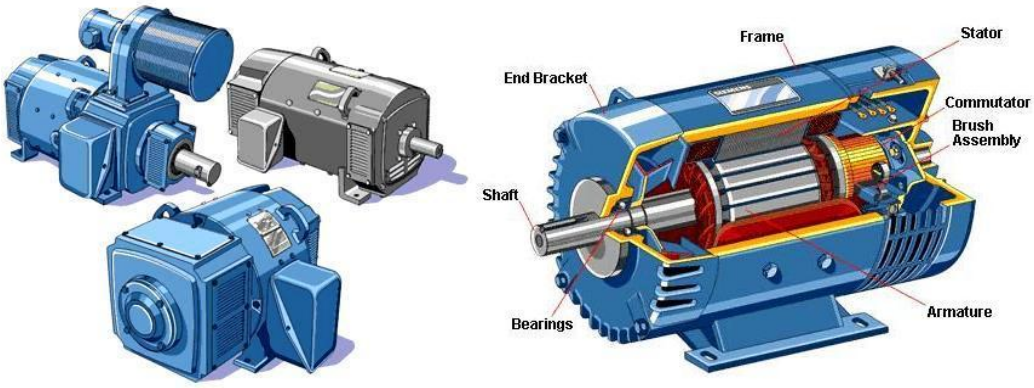



Figure 21: Brushed motor

- Brushless Motors. They use external commutation to create the alternating current from the direct current. Brushless motors have many applications in the field of machine tools, despite the fact that they are more expensive than conventional DC motors, since their design features provide greater performance and efficiency. The main characteristic is that the change in polarity of the rotor can be carried out without brushes. In this way, efficiency is enhanced by reducing friction and at the same time with a lower heat production.

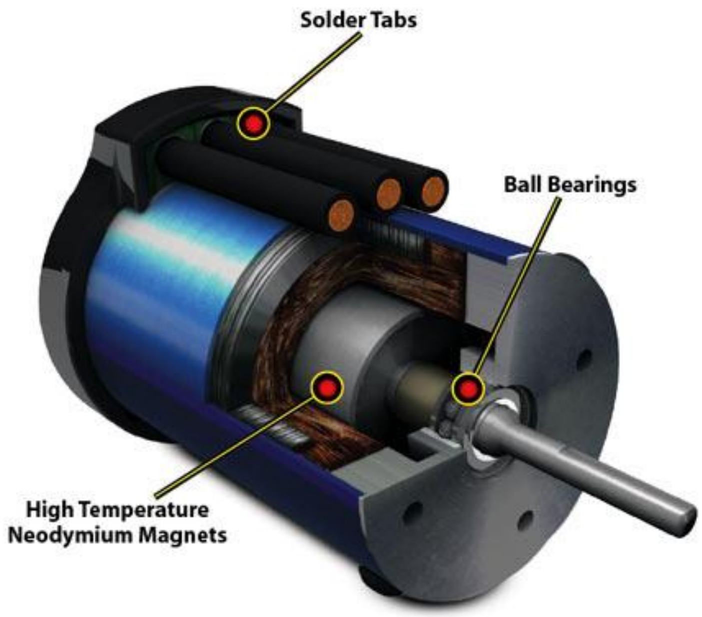



Figure 22: Brushless motor [BRUS13]

- Alternating Current Motors. AC motors convert alternating current into rotating torque. The AC motor consists of a stator, which is the fixed part that produces a rotating magnetic field by means of an alternating current, and the rotor, which rotates due to the rotating torque generated by the magnetic field. In machine tools, synchronous and asynchronous AC electric motors are currently used. Synchronous AC motors are distinguished by having a rotor that spins synchronously with the oscillating field of the current that drives it. The main advantages of synchronous AC motors are that their position and speed can be accurately controlled with open loop controls and that speed is independent of the load, so it can be accurately maintained.
- Stepping Motors. Stepping Motors. This is a type of brushless DC motor with similar operating principles. Nevertheless they have more stator poles than a classic brushless motor and they have a multitude of magnet rotor poles. Its structure allows to move the rotor of only one “step” by giving power to the following electromagnets respect to the starting position. Furthermore this type of motor can provide what is called detent torque, in fact powering only two stator poles the motor remains in that position opposing to motion a force proportional to magnetic field intensity.

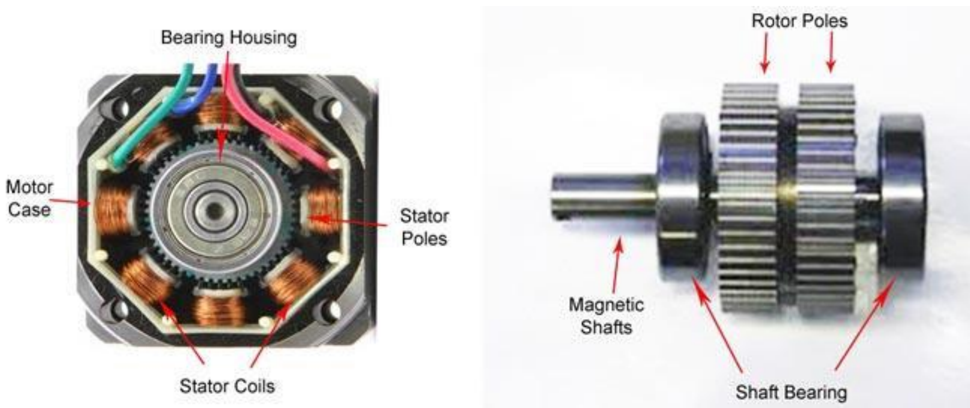



Figure 23: Stepping motor [STEP13]

### 3.2.2 Analysis of the gear

In this section components that have to transmit the rotary movement from drives will be analyzed. These mechanisms often have to transform the rotary energy to translatoric one, as in guide instances, but sometimes like for the spindle they only have to transfer the energy.

The mechanical power transmission from the motor to the actuating element can be accomplished by various driving gear. These include gearings, worm gearings, belt drives, chain drives and friction gears [LELI09], everyone has different advantages and disadvantages which are evaluated during the particular case design. In machine tool application, the two mechanisms most widely used are rack and pinion and, especially, ball screw.

As each axis requires different needs there will be a diagram for each axis: gear diagram could be divided into two classes depending on their task: diagram class that shows the trend of axes which have to move the slides and the other one includes axes which give cutting motion. The former has its trend measured by the speed that slides can reach and the transportable moving weight, instead the latter, usually referring to the spindle, takes into account rotational speed and torque transferable to the cutting tool (or to the workpiece).

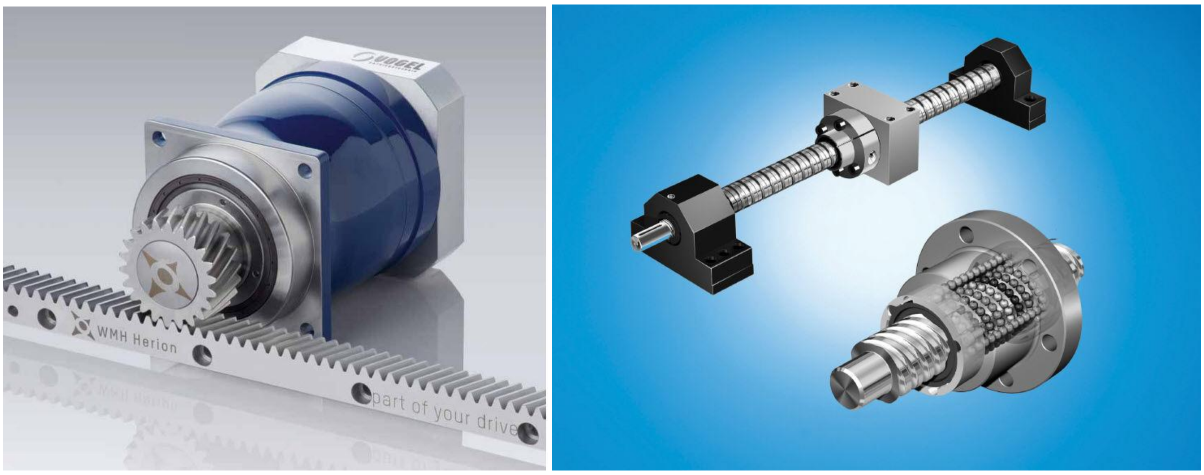



Figure 24: (a) Rack and pinion [RACK13] and (b) ball screw [BOSC09]

### 3.2.3 Analysis of workpiece and cutting tool

How it is explained in the diagram definition, cutting diagrams draw what happens at forces level in the contact area. To comprehend which characteristics influence cutting diagram behaviour, two different components have to be analyzed: tool and workpiece. Moreover it might be necessary operation type analysis and classification, as, in relation to the different requested operation, force types and cutting parameters change a lot. A parameter that could be used for the assignment of a particular scenario to diagrams is the material's machinability. In effect machinability takes into account not only workpiece's feature but also machining process parameters and operating conditions such as cutting tool material and geometry.

However in using machinability as comparison characteristics, two problems arise: machinability, as told above, depends on a various quantity of parameters but there is not a widely accepted way to quantify it, generally it is evaluated basing on the specific case and needed operating condition. Furthermore it could be said that machinability measures not only cutting forces needed to work the piece but also surface precision, time requested and tool wear. In this way, some parameters and factors which do not have any influence on the diagram bases would be anyway included in the analysis.

Therefore an examination of which factors condition diagram bases is convenient, better if dividing from tool, work material and operation type. Moreover, it must be said, there will be a different diagram, and so a new assignment every time that one between cutting tool, machining process parameters and workpiece changes.

Figure 25: Milling and drilling tools [BIG28]

### 3.2.4 Analysis of the heat production in the drive

Another diagram that is required for energy consumption forecasting is the drive heat one. This takes into account the heat generation due to drive operation. Also this diagram has as bases the torque and speed produced by the motor, but instead measuring absorbed electrical power, it measures power transformed in heat, in relation to torque and speed. Heat can be generated in different ways and due different elements operation.

Even if this diagram is very important for energy consumption forecasting, here an analysis for the assignment is not necessary since this assignment is the same of drive case.

### 3.2.5 Analysis of the cooling system

Cooling systems are used for all those areas where due to components work and components friction there are temperature increases causing variations of the components size and so precision losses, besides a higher components wear. Cooling systems are used in machine tools e.g. for the cooling of the spindle, the drives, the cooling lubricant, the control cabinet and the hydraulic system. So many uses make cooling system one of the main consumers of energy of machine tools [BREC12], despite cooling system can be considered an ancillary component.

# 4 Concept Development

Starting from components analysis, in this chapter characteristics for comparison and assignment of components to drive maps will be discussed and defined. These characteristics should reflect components behavior, for this reason characteristics that have an influence on diagrams bases have to be indentified and selected.

In the second paragraph there is the core of the present work, a fuzzy logic based assignment algorithm is described. This is divided into three steps: fuzzification, inference rules determination and defuzzification.

In the end different fuzzy operator and defuzzification methods are discussed and by means of an example those that should be preferred are shown.

## 4.1 Definition of characteristics for comparison and assignment of components to drive maps

Once the groups diagrams are defined, proceeding with the definition of most important characteristics for comparison and assignment of components is possible. Detailed analysis for each group is needed to figure out how component has to work, which is its environment and how that could influence component work. An analysis as general as possible is required, since in this way the energy consumption forecasting of any application context, is allowed.

There would be events that presents different possibilities to implement a specific function like in the gear case, that there are more then one mechanism to convert or to transfer the movement from the drive to the axis or spindle. In these cases the choice have to fall on a common characteristic presents in each possibilities. As the assignment is particularly useful when there in no diagrams and so there is no information to describe component behaviour, chosen features should be as simple as possible to be calculated.

Moreover characteristics choice should be done analyzing different impacts of the feature on diagram trend, in this manner these should be more significant for the assignment.

Below the paragraph has been divided in relation to the reference group. For each component, based on the analysis made in the paragraph 3.2, the characteristics for comparison and assignment of components to drive maps have been determined.

The figure 26 summarizes the assignment process: this can be divided into two main groups: the study of components that includes component selection and analysis and the choice of characteristics; and the fuzzy process that includes fuzzification, inference rules determination and defuzzification processes.

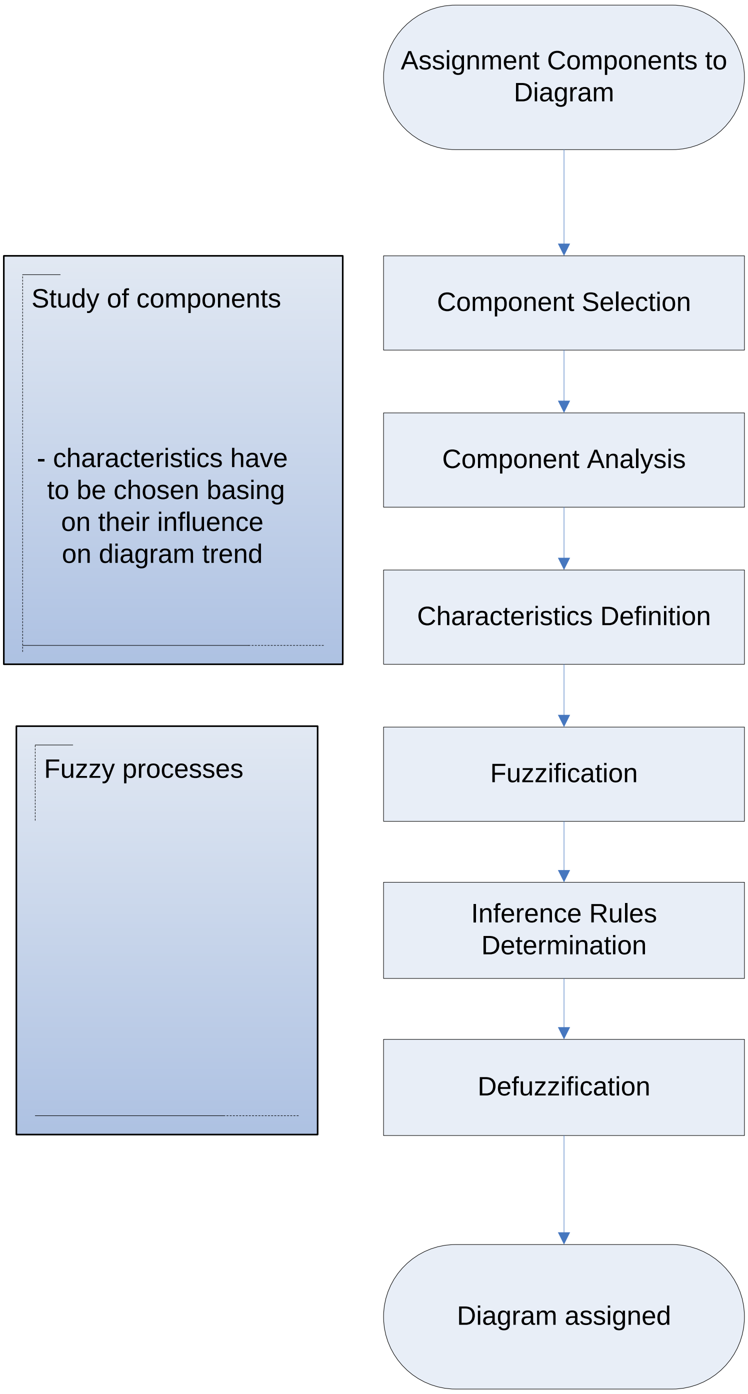



Figure 26: Flow chart of the assignment process

### 4.1.1 Drive characteristics

Once the different motors have been analyzed, finding other suitable characteristics for comparison is necessary. These have to influence the energy consumption of the motor and its performance in terms of rotational speed and torque that can be delivered. Thus, other three features can be identified:

- Rated torque. This is a motor's nameplate data and it permits to figure out force that motor can transfer to the gear and therefore later to the spindle or to the slides.
- Rated speed. This is another very important parameter, as almost all motors used in machine tool application have the peculiarity of presenting a steady torque until a fixed rpm, that is the motor's rated speed or also called base speed. So this characteristic is fundamental to understand motor diagram trend, in fact if the motor runs at a higher speed than its rated speed value, transmittable torque exponentially decreases. The product of rated speed and torque defines the optimum mechanical power of the motor, obtaining an fundamental information of the motor's performance.
- Rated current. Also this parameter is a motor's nameplate data and it's very important since, unlike torque and speed that refer to mechanical power, with the needed voltage it defines electrical energy consumption of the motor. Thus, while torque and speed give information about mechanical motor's performance, rated current gives information about its efficiency. Same mechanical power, a greater requested current entails a greater electrical energy consumption and so a lesser motor's efficiency.

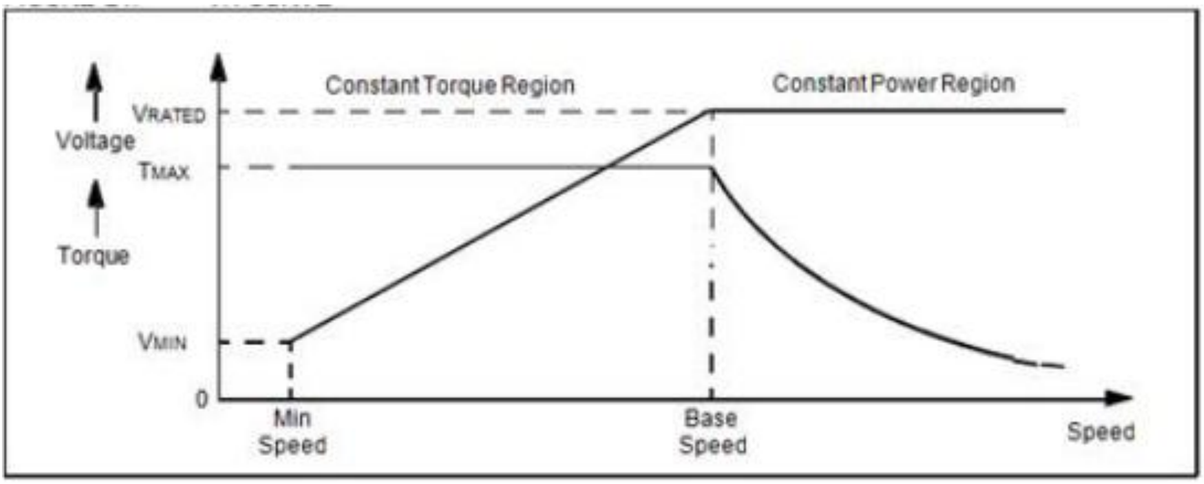



Figure 27: Example of Torque (Nm)/Power (kW) and Speed (rpm) motor diagram

### 4.1.2 Gear characteristics

The features choice must consider features that have an influence on the variables mentioned in the paragraph 3.2.2. It must be said that an ideal configuration does not exist but the selected solution is always a compromise between required applications and pros and cons of each gear type and its configurations.

After an analysis of both motion transmission mechanisms, the chosen characteristics are:

- Size: this feature needs to be very general because, depending on gear type, relevant dimensions are different. For instance, rack and pinion size is evaluated by measuring pinion's diameter and number of teeth, instead ball screw size depends on its length and diameter. Diameter affects both forces that screw can bear, so also the transferable torque and the load capacity. Actually, this is also affected by other parameters as ball diameter, number of load bearing balls (number of circuits, surface hardness of roller races and manufacturing tolerances [LELI09]). The figure 26 shows the dynamic load capacity of ball screws according to the diameter of the screw. On the other hand length is equally important, it has a direct influence on the maximum rotational speed that can be reach. Moreover it is necessary to focus on the relation between these to sizes because wrong ratio can lead to a bend of the screw that affects machine's performance and ball screw life.

  As regards allowable forces to be transmitted by rack and pinion, as said before, depend on its size, particularly on its modulus, that is the ratio between pinion diameter and the number of teeth. This obviously has percussions on the dynamic load capacity. The figure 27 shows certain illustrative values of the allowable force for each pinion according to the modulus.
- Transmission: transmission can be considered as the ratio between output and input speed. If the gear is a rack and pinion to permit a higher speed, pinion diameter has to be reduced, but this reduces its modulus, reducing the load capacity. Instead in ball screw instance the guide speed is related to the screw pitch, to a greater pitch corresponds a higher speed, nevertheless machine loses mechanical advantage.

In gear design, keeping attention on these parameters trade-off is very important because their configuration affects machine tool performance, particularly regarding work precision.

However, not being an interesting parameter for the assignment to diagram, it will not take into consideration in this work. In the following analysis only ball screw case will be considered as they represents more than 90% of machine tool application and almost 100% of the case with a guide’s length less than 5 meters [LOPE09].

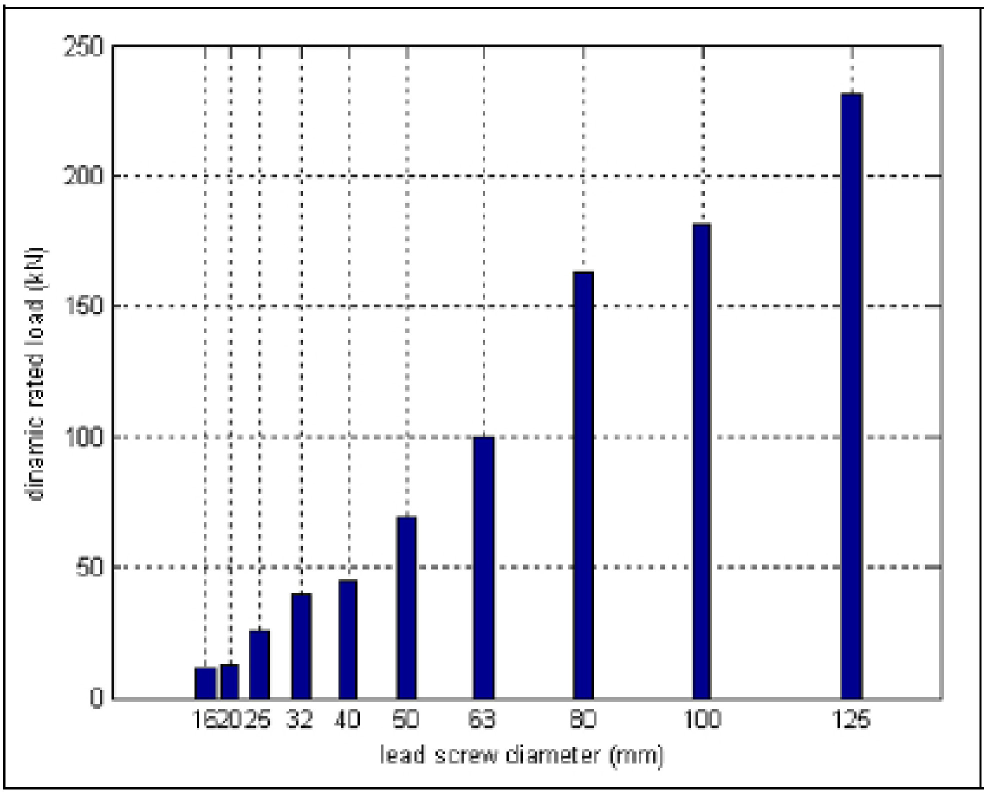



Figure 28: Dinamic load capacity (kN) depending on ball screw diameter (mm) [LOPE09]

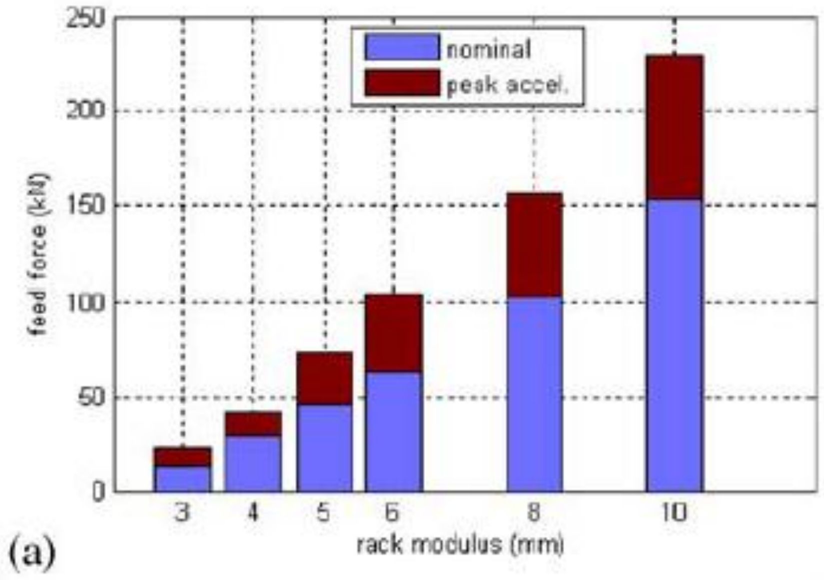



Figure 29: Transmittable force expressed in kN depending on rack modulus (mm) [LOPE09]

### 4.1.3 Cutting characteristics

Concerning workpiece analysis, the characteristic that has to be included is material work hardness. Hardness can be considered as material resistance degree to be penetrated (it measures the resistance opposed by the material while another one trying to penetrate it). Therefore it is particularly significant because it can be directly linked to machining feed and speed, in fact material with low hardness can be processed faster than a harder one.

About cutting tool characteristics choice is more complicated because there are more parameters that could change. The analysis of each characteristic is carried on considering that every factor is stable except that taken into account. The characteristics that have been chosen are:

- Cutting tool material hardness. Depending on its hardness, the tool can reach different speed ranges and can move forward with different feed; increasing hardness higher range of speed and feed are permitted.
- Cutting tool diameter. Besides conditioning cutting speed, this is the main responsible of cutting width. This is easy to understand in fact a bigger tool allows to work a bigger surface. Moreover it can bear higher strengths and so, higher rotational speed.

- Cutting edge length. As diameter permits to machine a superior width surface, cutting edge length permits to work a material more deeply. Nevertheless, contrary to cutting tool diameter, it is inversely proportional to the acceptable rotational speed.

As all these parameters can be different from each other depending on machine and operation type, the following fuzzy application will be focused on milling tools.

The table 3 summarizes all the chosen characteristics for the assignment to cutting diagrams. Furthermore their influence, positive or negative, on diagram bases it is represented.

| Diagram type | Characteristics | Main influnece on (diagram bases) |
|---|---|---|
| cutting | work material hardenss | speed, feed |
| | cutting tool diameter | cutting width |
| | cutting tool material | speed, feed |
| | cutting edge lenght | cutting deep |

Table 3: Characteristics and influenced parameters of cutting diagram

### 4.1.4 Cooling system characteristics

Electric energy consumption depends on thermal power required to cool work environment. Total thermal power is calculated by summing various thermal powers necessary in machine operations and different components. Generally cooling systems have two tanks, one for cutting, one for hydraulic oil. Many cooling system classifications might be possible for example referring at which refrigerant is used or condensation type (water, air). But these do not allow to figure out the energy behaviour. Cooling system's efficiency and thus its electric energy consumption does not depend on the cooling process but on different external conditions [BIOL13]. In particular the following characteristics are extremely important:

- Cooling capacity. This parameter is a measure of cooling system's performance, it expresses the removable heat quantity by the cooling system. Cooling capacity usually changes depending on the temperature level. It is a nameplate data of the cooling system. It is necessary to pay attention not to over dimension the system because it would consume more energy than required.

- Expansion fluid temperature. Cooling systems have a higher efficiency and lower energy consumption when they work with higher expansion temperatures, that is the required water temperature.
- Work environment temperature. A cooling system is more efficient if it has to work with low temperature than an installation situated in a hotter location. Hot environments generate a greater energy consumption.

## 4.2 Definiton of Assignment Algorithm

To assign component to drive maps, a fuzzy logic based method has been chosen. As it is said in the chapter 2, fuzzy logic can be applied to decisional and control system. In these cases it is very useful as it is oriented towards an integration between man and machine: expert has to "teach" the elaboration system how working and making choices, as reasoning can be imitated and apply again and again to a big amount of data. Fuzzy system skill of reproducing expert judgment on a big amount of data is one of the reason to consider fuzzy logic able to reach an integration like the one mentioned above. Furthermore fuzzy logic could be very useful all the time it's necessary to cope with conflicting objectives. In this case it can be used to cope efficiency/precision trade-off, indeed building a general algorithm is necessary, and aiming for efficiency or precision is possible by respectively increasing or decreasing the number of membership functions.

Before starting with assignment algorithm explanation, some considerations are necessary. As machine tool field is huge and includes a multitude of different machines, operation types, cutting tools and drives limiting the scope of application of this analysis is necessary. Concerning the machine scenario the study will be focused on the milling process and the tools will be analyzed are the end-mills. Furthermore drives will be considered only of two different types: asynchronous motor for the main drive (that is drive related to the spindle) and synchronous motor for the movement of the axes. As concerning gear analysis, this will focus on ball screw case.

The algorithm and the following assignment will be divided in groups in relation to referred component and hypothesis, whereas after by means of a flow chart, how facing with a case as general as possible and methods to perform calculations will be explained. Once determined the most important characteristics, at this point it is necessary to start with the real fuzzy logic analysis that will be divided for each component in three steps:

- Fuzzification: in this stage, values range for each component will be evaluated and based on these, membership function will be chosen and determined. Triangular and trapezoidal functions are preferred for their greater similarity and simplicity to be applied at linguistic variables rather than bell-shape functions. For every case, limit values has been chosen basing on literature research and particular shape choice will be explained, anyway for every characteristic the goal is to have classes as balanced as possible. In most cases the number of classes is three as it can be a good compromise between precision and efficiency, particular attention must be done on this size as increasing the number of classes, the number of rules will increase exponentially. However a choice of a odd number of class is recommended as in this way a middle class is always present and this supports the similarity with human reasoning. Below in the figure 30 a flow chart of fuzzification process is presented.

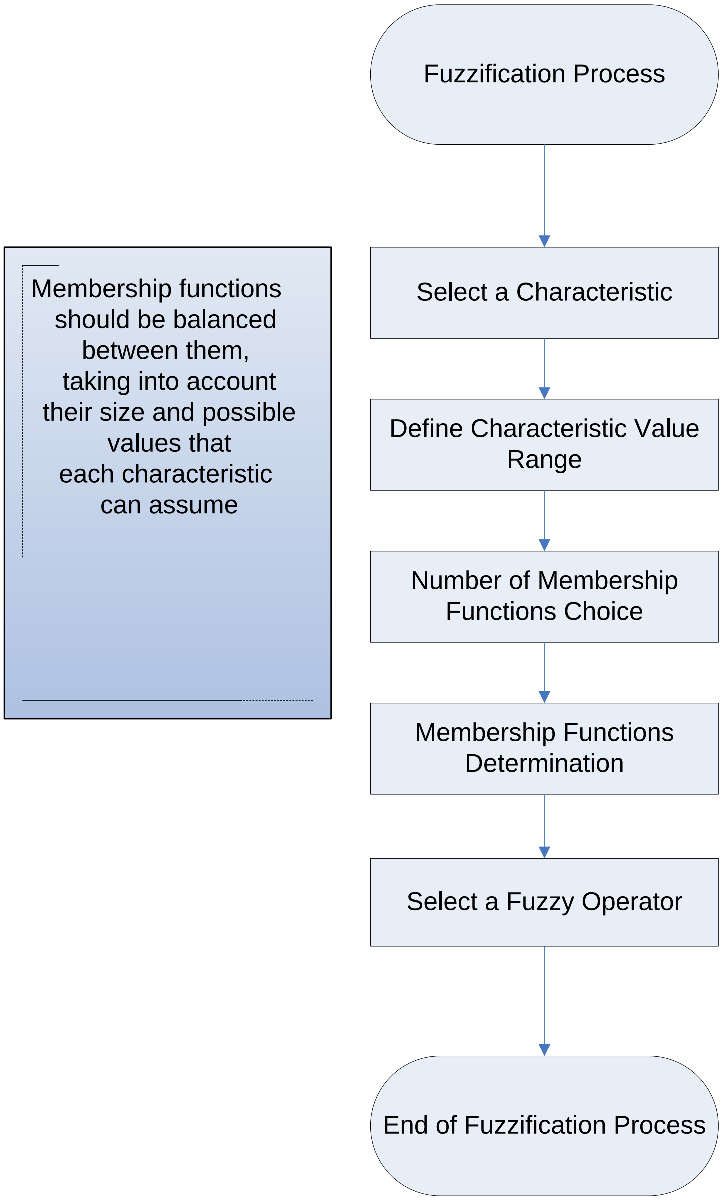



Figure 30: Fuzzification process flow chart

- Inference rules determination. For every component, a table including all the rules will be shown and the concept used for the assignment of each “if-part” to its “then-part” (diagram number) will be discussed. Here in the figure 31 a general explanation is shown by means of a flow chart.

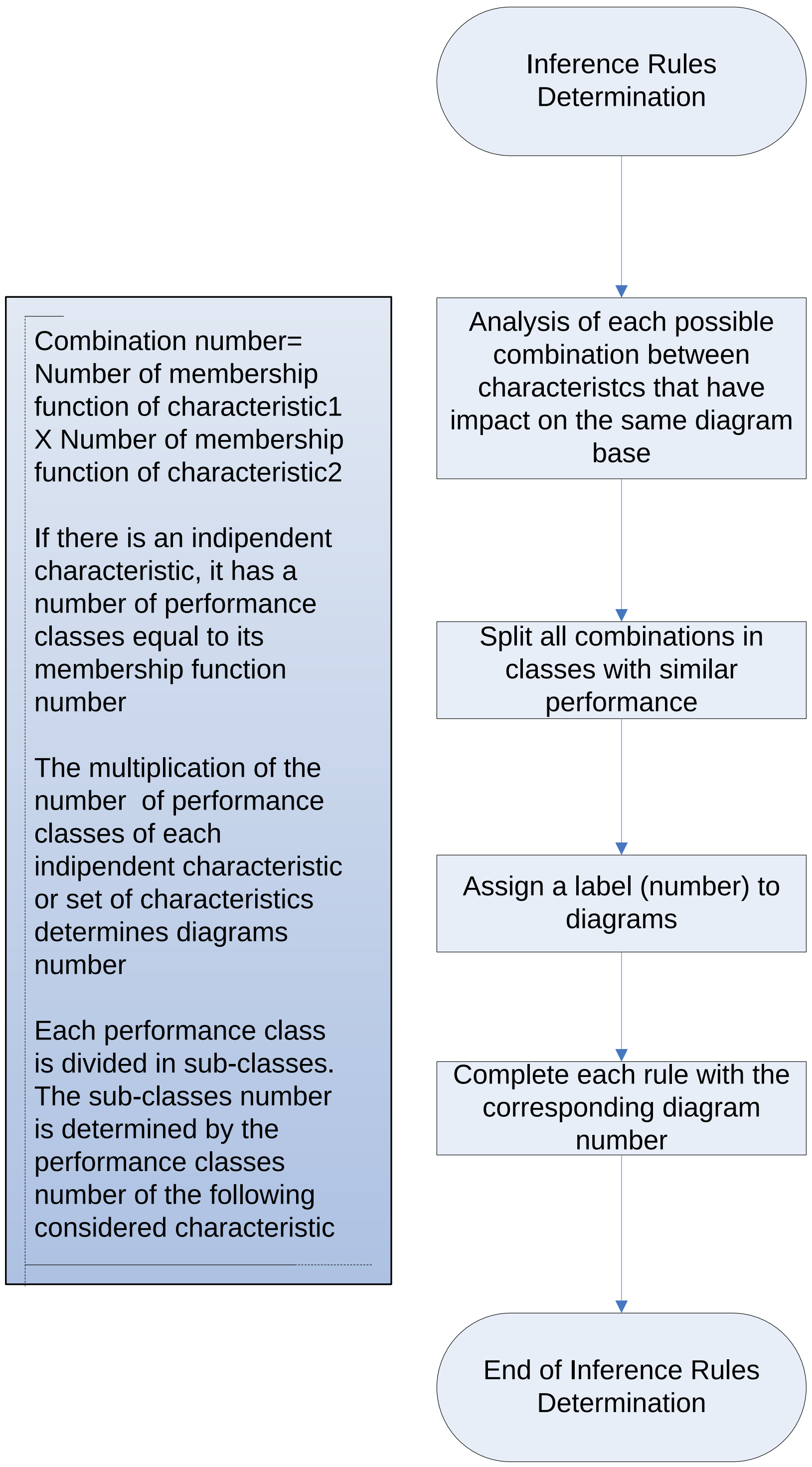



Figure 31: Flow chart of inference rules determination

- Defuzzification. This step requests an analysis more detailed as this application has the peculiarity that the result is not a precise value, what in fuzzy word is called crisp number, obtained from the equivalent value on the x-axis, but the output must be an assignment to a diagram that is labelled with a number. Chosen defuzzification methods are COG and maximum value method which are explained in the chapter 2.2. Anyway pros and cons of each method, their applicability in this case and a more detailed defuzzification analysis will be discussed after the first two steps.

### 4.2.1 Definition of the assignment algorithm for the drive

Once determined the most important characteristics, now it is possible to proceed with the fuzzy process. How it is said in the previous introduction, first of all, for the drive it is better to split the analysis in two different cases before starting fuzzy logic process, as fuzzy logic has to work with instances with a truth degree included in the interval [0,1], whereas in this case a true or false analysis is requested (drive can be Synchronous or Asynchronous). As fuzzy logic is based on rules which usually are “if… then..” type, using an if-then rules to divided the two situation could be appropriated. Thus, in the first step specifying drive application is needed; the rule allowable to divide cases is:

*If main drive is analyzed then drive is an Asynchronous Motor;*

*Else the drive is a Synchronous Motor.*

In both case the characteristics must be analyzed are: rated speed, torque and current. For each feature, first thing that has to be studied is the range of the possible values that characteristic can assume. Those values are extracted from Siemens’ site, big manufacturer of drives and components for machine tools.

Starting the analysis from the synchronous motor and in particular from its rated speed, the values range varies from 2000 and 6000 rpm [SIEM07]. As in most of the cases, even here three different classes has been chosen, dividing from low, medium and high rated speed. The membership functions with their related limit values are shown in the diagrams.

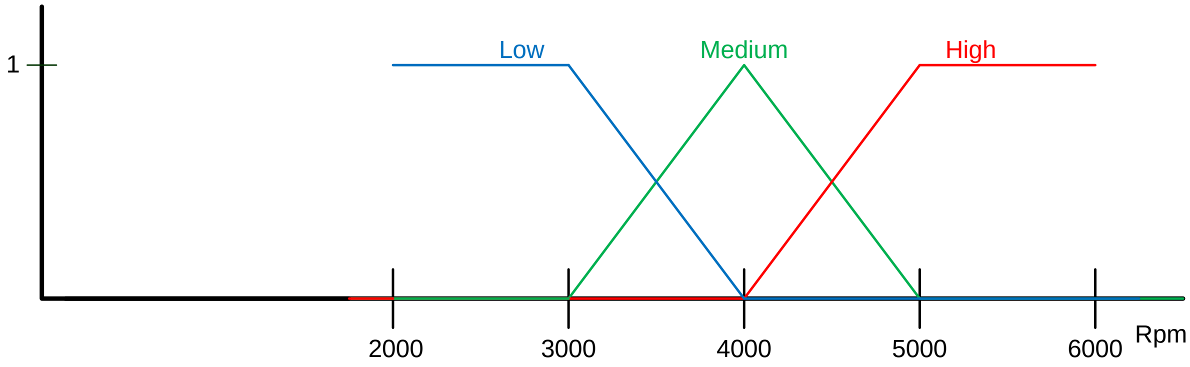



Figure 32: Rated speed synchronous motor membership functions

As concerning the rated torque for the drive responsible of the axes movement, this can be from 2 till reaching 60 Nm [SIEM07]. The membership functions are represented in the diagram below.

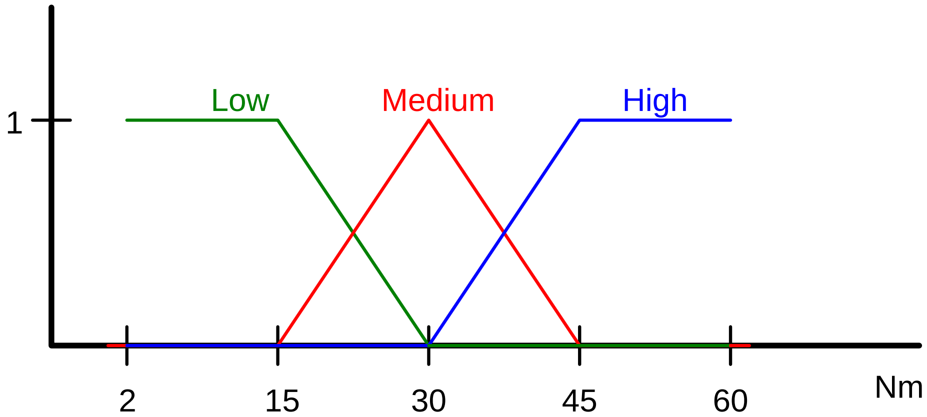



Figure 33: Rated torque synchronous motor membership functions

Rated current range goes from 2 to 18 A [SIEM07]. The related diagram shows the different membership functions, also in this case three different functions, low, medium and high, can well represent the interval of values. As it will be shown later, drive for axes case is simpler than main drive analysis as requested power is much lower and thus the values range is confined.

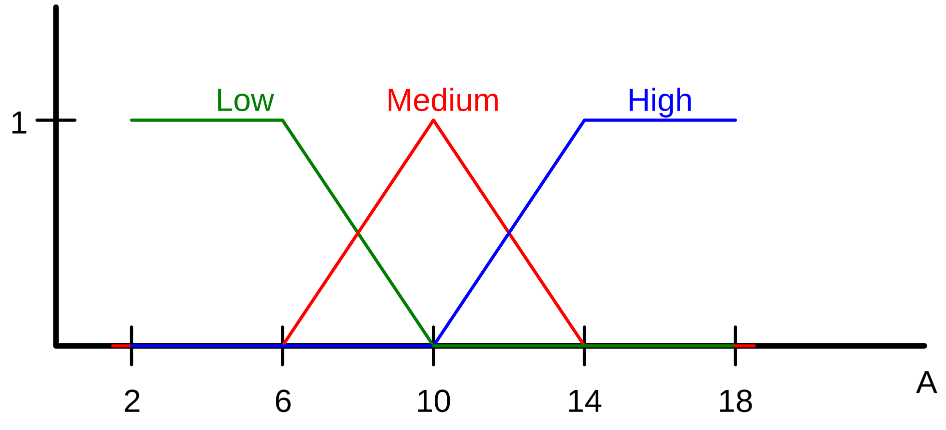



Figure 34: Rated current synchronous motor membership functions

Regarding asynchronous motor, this is usually employed as main drive, to transfer the movement to the spindle and thus to give the cutting motion to the tools. If for the rated speed the range does not change, and rather it is even more confined, this cannot be said for the rated torque and current. In the diagram below limit values and membership functions for rated speed instance are shown.

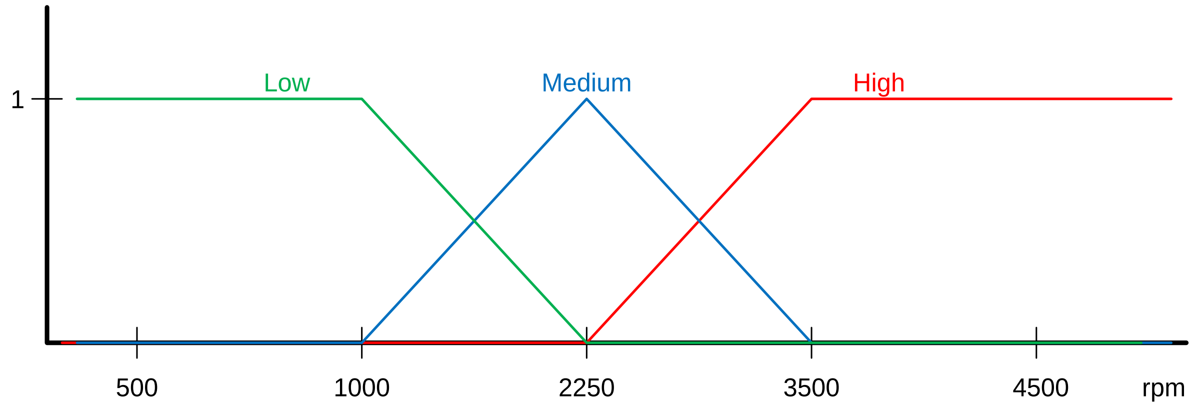



Figure 35: Rated speed asynchrnous motor membership functions

As beforehand said, both torque and current ranges are highly more extended than drive for axes case. Therefore an analysis more detailed is needed, this time a choice of five different classes has been considered more appropriated. Moreover linguistic operators such as “very”, have been used; like a odd number choice of classes, linguistic operators facilitate the values disposition in classes. Rated torque range comprehends values from few tens to around 1500 Nm, whereas rated current can be from around 10 to 210 A [SIEM12]. Classes have been divided in very low, low, medium, high and very high both for torque and current. In the diagrams all the membership functions are shown with their related limit values.

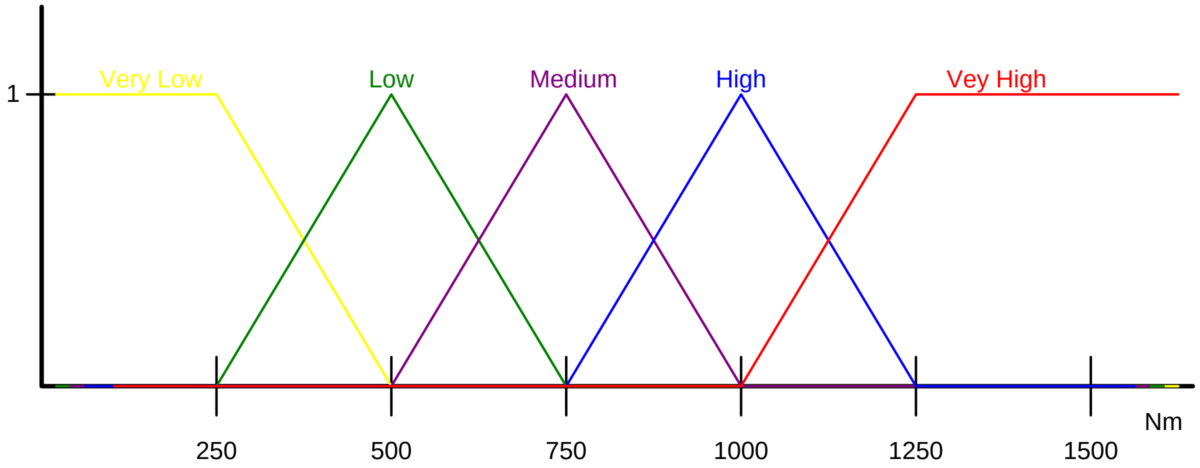



Figure 36: Rated torque asynchronous motor membership functions

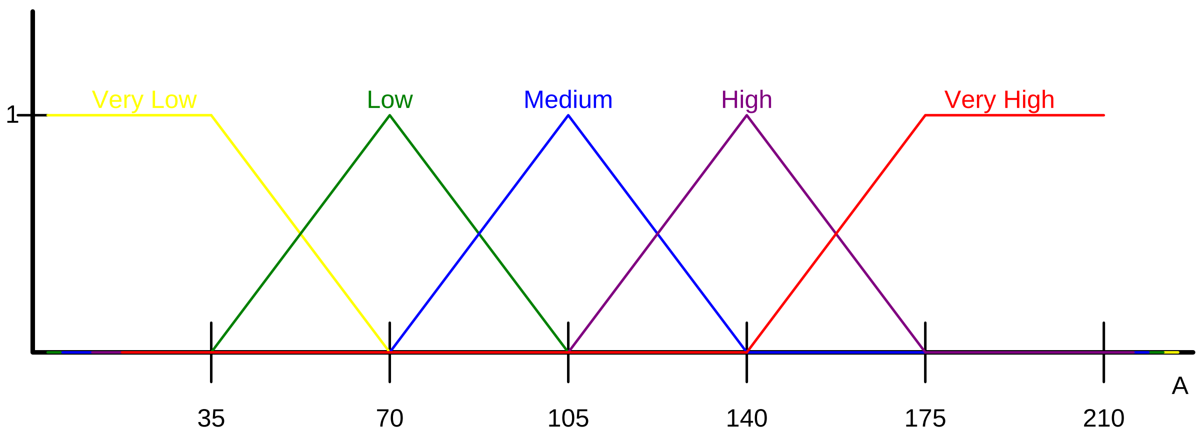



Figure 37: Rated current asynchronous motor membership functions

At this point the fuzzification process can be retained concluded and proceeding with inference rules determination is possible. Inference rules number is determined by multiplying the classes number of each characteristic, so for synchronous motor there will be 27 rules and for asynchronous one, 75 rules.

The rules, as already said, are all if-then type. The *if* part contains the hypothesis related to the membership classes while the *then* part specifies, under those conditions, which diagram represents the behaviour better. It has been presumed that all caught diagrams are divided in classes and labelled with an identification number. To clarify that concept an example of inference rules for synchronous case is presented.

*If the Rated Current is Low and the Rated Speed is Low and the Rated Torque is Low then that drive have to be assigned to Diagram number 7;*

*If the Rated Current is Medium and the Rated Speed is High and the Rated Torque is Low then that drive have to be assigned to Diagram number 5;*

All needed rules are expressed in the following tables respectively for synchronous and asynchronous motor.

| Synchronous Motor | | | |
|---|---|---|---|
| **Rated Current** | **Rated Speed** | **Rated Torque** | **Diagram** |
| Low | Low | Low | 7 |
| Low | Low | Medium | 7 |
| Low | Low | High | 8 |
| Low | Medium | Low | 7 |
| Low | Medium | Medium | 8 |
| Low | Medium | High | 9 |
| Low | High | Low | 8 |
| Low | High | Medium | 9 |
| Low | High | High | 9 |
| Medium | Low | Low | 4 |
| Medium | Low | Medium | 4 |
| Medium | Low | High | 5 |
| Medium | Medium | Low | 4 |
| Medium | Medium | Medium | 5 |
| Medium | Medium | High | 6 |
| Medium | High | Low | 5 |
| Medium | High | Medium | 6 |
| Medium | High | High | 6 |
| High | Low | Low | 1 |
| High | Low | Medium | 1 |
| High | Low | High | 2 |
| High | Medium | Low | 1 |
| High | Medium | Medium | 2 |
| High | Medium | High | 3 |
| High | High | Low | 2 |

| Rated Current | Rated Speed | Rated Torque | Diagram |
|---|---|---|---|
| High | High | Medium | 3 |
| High | High | High | 3 |

Table 4: Synchronous motor inference rules and assigned diagrams

Obviously each drive should have its own diagram, but to increase the efficiency, for every characteristic a classes division has been chosen. The same reasoning has been done to avoid that every rule would have had a different assignment, thus 3 and 5 rated current classes and also 3 and 5 power classes has been chosen respectively for synchronous and asynchronous motor, identifying 9 diagram categories for the drive for axes and 25 ones for the main drive.

| Asynchronous Motor | | | |
|---|---|---|---|
| **Rated Current** | **Rated Torque** | **Rated Speed** | **Diagram** |
| Very Low | Very Low | Low | 21 |
| Very Low | Very Low | Medium | 22 |
| Very Low | Very Low | High | 23 |
| Very Low | Low | Low | 21 |
| Very Low | Low | Medium | 22 |
| Very Low | Low | High | 23 |
| Very Low | Medium | Low | 22 |
| Very Low | Medium | Medium | 23 |
| Very Low | Medium | High | 24 |
| Very Low | High | Low | 22 |
| Very Low | High | Medium | 24 |
| Very Low | High | High | 25 |
| Very Low | Very High | Low | 23 |
| Very Low | Very High | Medium | 24 |
| Very Low | Very High | High | 25 |
| Low | Very Low | Low | 16 |
| Low | Very Low | Medium | 17 |
| Low | Very Low | High | 18 |
| Low | Low | Low | 16 |
| Low | Low | Medium | 17 |
| Low | Low | High | 18 |
| Low | Medium | Low | 17 |
| Low | Medium | Medium | 18 |
| Low | Medium | High | 19 |
| Low | High | Low | 17 |
| Low | High | Medium | 19 |
| Low | High | High | 20 |

| | | | |
|---|---|---|---|
| Low | Very High | Low | 18 |
| Low | Very High | Medium | 19 |
| Low | Very High | High | 20 |
| Medium | Very Low | Low | 11 |
| Medium | Very Low | Medium | 12 |
| Medium | Very Low | High | 13 |
| Medium | Low | Low | 11 |
| Medium | Low | Medium | 12 |
| Medium | Low | High | 13 |
| Medium | Medium | Low | 12 |
| Medium | Medium | Medium | 13 |
| Medium | Medium | High | 14 |
| Medium | High | Low | 12 |
| Medium | High | Medium | 14 |
| Medium | High | High | 15 |
| Medium | Very High | Low | 13 |
| Medium | Very High | Medium | 14 |
| Medium | Very High | High | 15 |
| High | Very Low | Low | 6 |
| High | Very Low | Medium | 7 |
| High | Very Low | High | 8 |
| High | Low | Low | 6 |
| High | Low | Medium | 7 |
| High | Low | High | 8 |
| High | Medium | Low | 7 |
| High | Medium | Medium | 8 |
| High | Medium | High | 9 |
| High | High | Low | 7 |
| High | High | Medium | 9 |
| High | High | High | 10 |
| High | Very High | Low | 8 |
| High | Very High | Medium | 9 |
| High | Very High | High | 10 |
| Very High | Very Low | Low | 1 |
| Very High | Very Low | Medium | 2 |
| Very High | Very Low | High | 3 |
| Very High | Low | Low | 1 |
| Very High | Low | Medium | 2 |
| Very High | Low | High | 3 |
| Very High | Medium | Low | 2 |
| Very High | Medium | Medium | 3 |
| Very High | Medium | High | 4 |
| Very High | High | Low | 2 |
| Very High | High | Medium | 4 |
| Very High | High | High | 5 |
| Very High | Very High | Low | 3 |
| Very High | Very High | Medium | 4 |
| Very High | Very High | High | 5 |

Table 5: Asynchronous motor inference rules and assigned diagrams

In this way both electrical power and mechanical power are considered. The current class expresses the electrical efficiency while the torque and speed ratio states the mechanical performance of the drive. Referring to the asynchronous motor the rated current intensity detects the diagram class between [1,5], [6,10], [11,15], [16,20], [21,25] and the torque and speed ratio identifies the exact number within that class. The same concept has been applied for the synchronous motor but, as beforehand said, with 3 class for the rated current and 3 class for the mechanical power.

For both the assignment it has been considered that the number suggests the drive class from the lower class (diagram number 1: low torque and speed with high rated current) to the upper class (diagram number 9 or 25: high torque and speed with low rated current). Below defuzzificazion diagrams for synchronous and asynchronous motors are shown. In defuzzification diagrams there is no need of a measure unit but they only include as many functions as the related table states in the output of the rules.

For remained components, reasoning is the same of drive case: values range study, membership functions and inference rules determination.

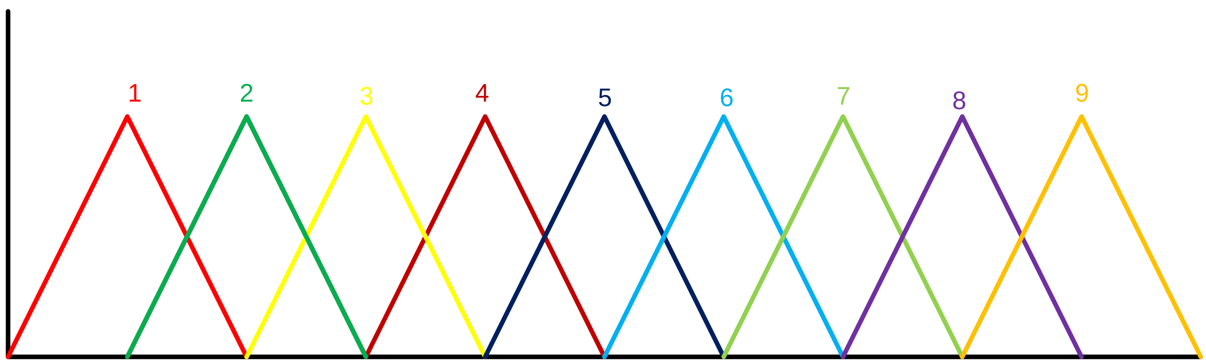



Figure 38: Synchronous motor defuzzification diagram

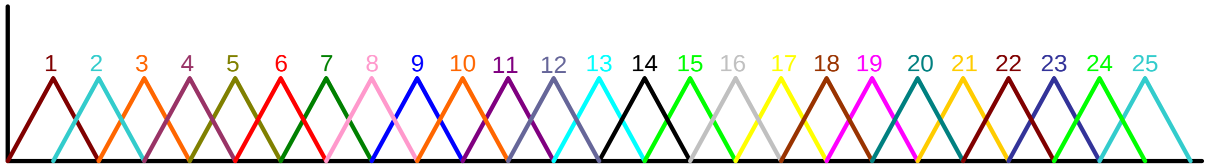



Figure 39: Asynchronous motor defuzzification diagram

### 4.2.2 Definition of the assignment algorithm for the gear

How explained in the gear analysis (4.1.2), only ball screw analysis will be carried on. Following figure shows possible sizes for each ball screw characteristic. Only ball screw length up to 5 metres will be considered, as usually in machine tools environment over this length linear motor drives are preferred. The figure 29 shows for each standard length (dark grey in the picture) wich pitch and diameter are preferred, moreover made to order ball screw lengths are presented (light grey in the picture).

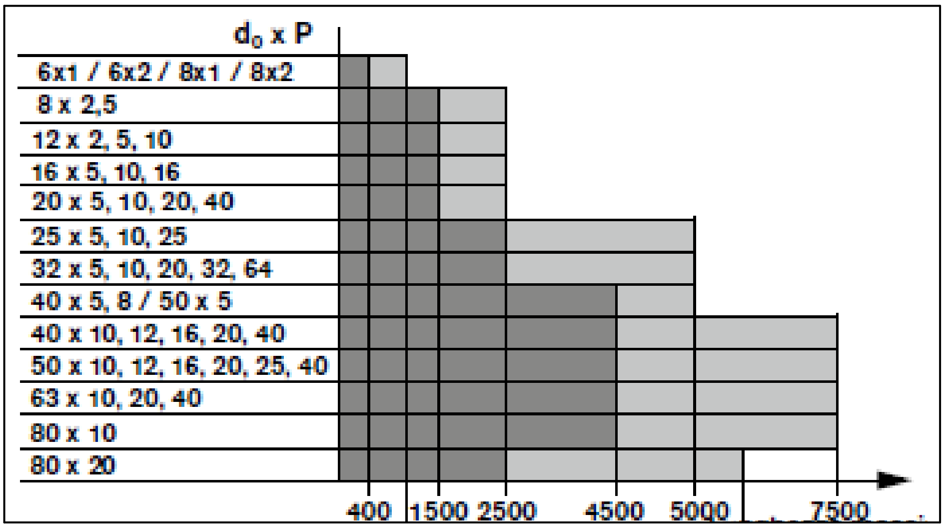



Figure 40: ball screw lengths, pitchs and diameters. The physical quantities are expressed in millimetres [BOSC11]

Basing on this values, membership functions related to ball screw length, diameter and pitch are developed. As for most of characteristics, three categories classification has been chosen, but unlike previous functions, medium pitch class has a different shape, a scalene triangular shape has been considered more appropriated in order to get three balanced classes, not from the point of view of the class size, but as regards the possible values assumable by the feature “pitch”.

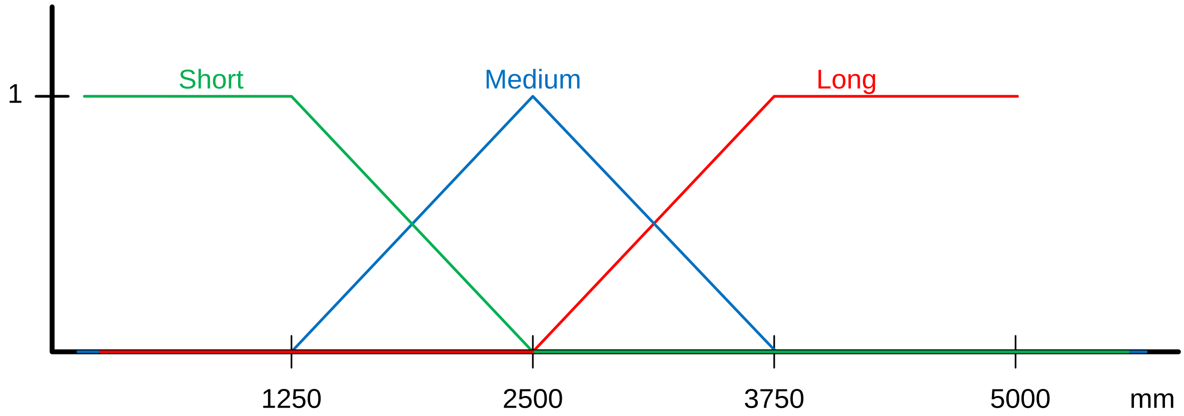



Figure 41: Ball screw length membership functions

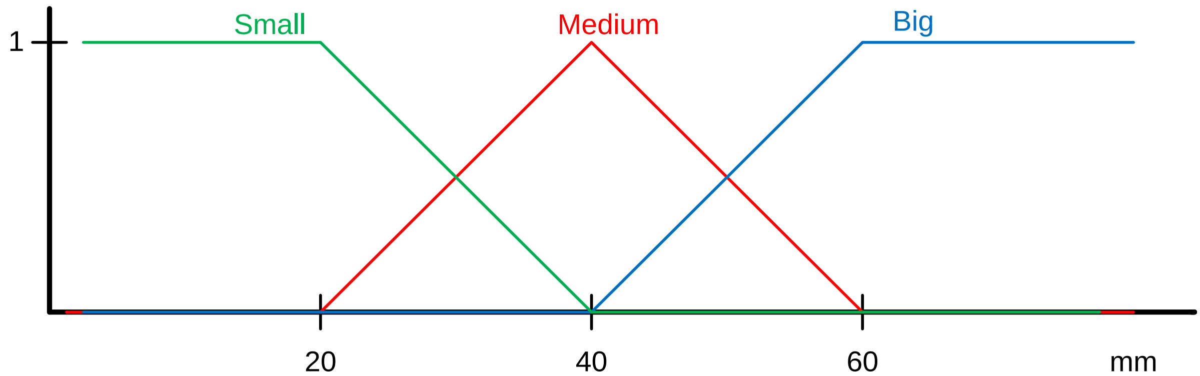



Figure 42: Ball screw diameter membership functions

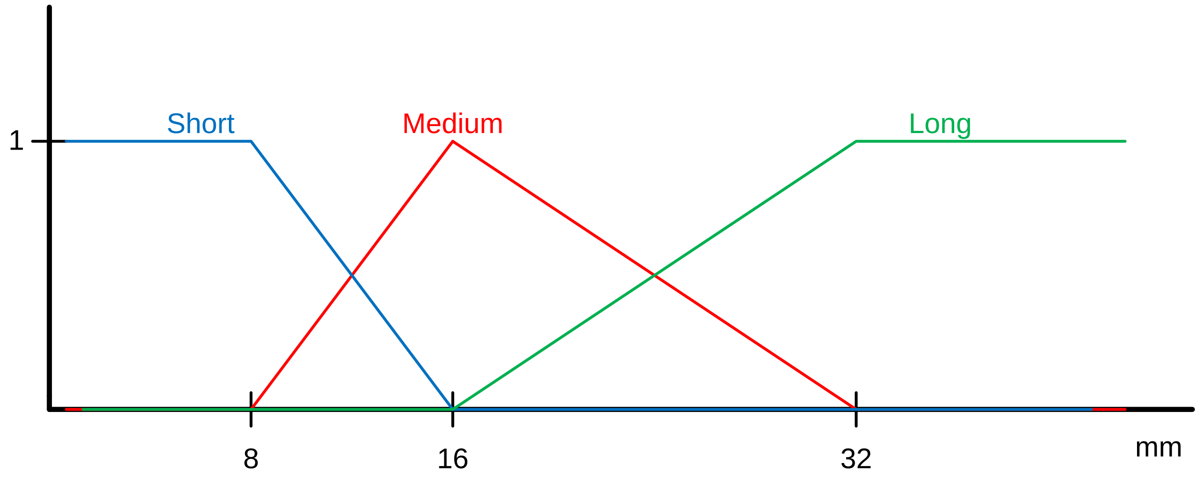



Figure 43: Ball screw pitch membership functions

Once all the membership functions are determined, proceeding with fuzzy process is possible. As well as drive case, all inference rules are if-then type, taking into account each characteristic and class, ball screw classification leads to 27 rules (3 x 3 x 3). In the table below all the rules are summarized.

| Ball Screw | | | |
|---|---|---|---|
| **Length** | **Diameter** | **Pitch** | **Diagram** |
| Short | Small | Short | 2 |
| Short | Small | Medium | 3 |
| Short | Small | Long | 3 |
| Short | Medium | Short | 5 |
| Short | Medium | Medium | 6 |
| Short | Medium | Long | 6 |
| Short | Big | Short | 8 |
| Short | Big | Medium | 9 |
| Short | Big | Long | 9 |
| Medium | Small | Short | 1 |
| Medium | Small | Medium | 2 |
| Medium | Small | Long | 3 |
| Medium | Medium | Short | 4 |
| Medium | Medium | Medium | 5 |
| Medium | Medium | Long | 6 |
| Medium | Big | Short | 7 |

| | | | |
|---|---|---|---|
| Medium | Big | Medium | 8 |
| Medium | Big | Long | 9 |
| Long | Small | Short | 1 |
| Long | Small | Medium | 1 |
| Long | Small | Long | 2 |
| Long | Medium | Short | 4 |
| Long | Medium | Medium | 4 |
| Long | Medium | Long | 5 |
| Long | Big | Short | 7 |
| Long | Big | Medium | 7 |
| Long | Big | Long | 8 |

Table 6: Ball screw inference rules and assigned diagrams

In this case, to reduce diagrams quantity, a division between components influencing the load capacity and slides' speed has been done; thus three classes for the diameter and also three classes for pitch and length ratio have been identified, so this means the assignment can chose 9 different diagrams. As in the previous case the performance of the ball screw grows with the increasing of the diagram number. Defuzzification diagram will be the same as synchronous drive instance.

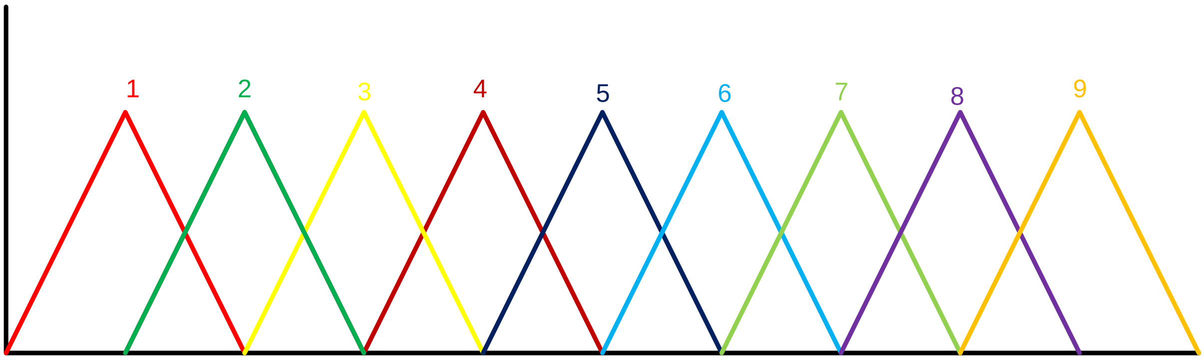



Figure 44: Ball screw defuzzification diagram

### 4.2.3 Definition of the assignment algorithm for the cutting

In this case, characteristics that have to be analyzed are four: workpiece hardness, cutting tool diameter, cutting tool edge length and cutting tool hardness, thus more inference rules are expected. As in most of the previous cases, the membership functions chosen number is three. For workpiece hardness and cutting tool diameter, being membership interval of values not vey extended, functions shape are normal triangle or trapezoid. Instead for cutting tool edge length and hardness, a symmetrical shape has not been chosen but different shape has been preferred in order to represent those characteristics behaviour better and their similarity; for example taking into account a given cutting tool varying its edge length in a medium-long range, it is possible to highlight that cutting parameters do not change as much as in a short-medium range, this is the reason for the membership functions different slopes.

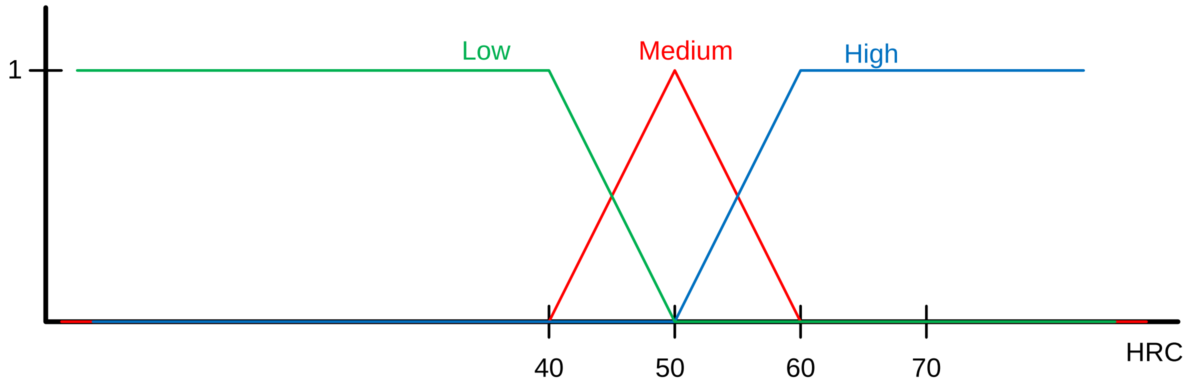



Figure 45: Workpiece hardness membership functions. Values range took from the literature [MITSb13]

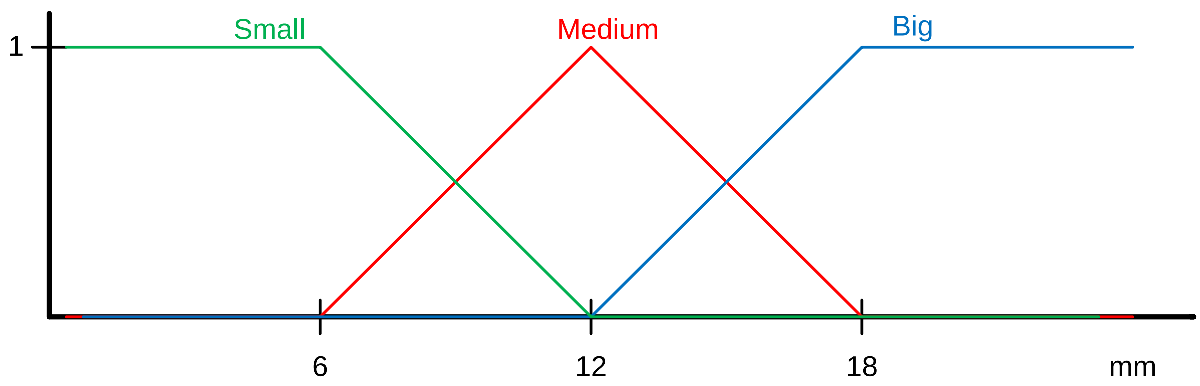



Figure 46. Cutting tool diameter membership functions. Values range took from the literature [MITSa13]

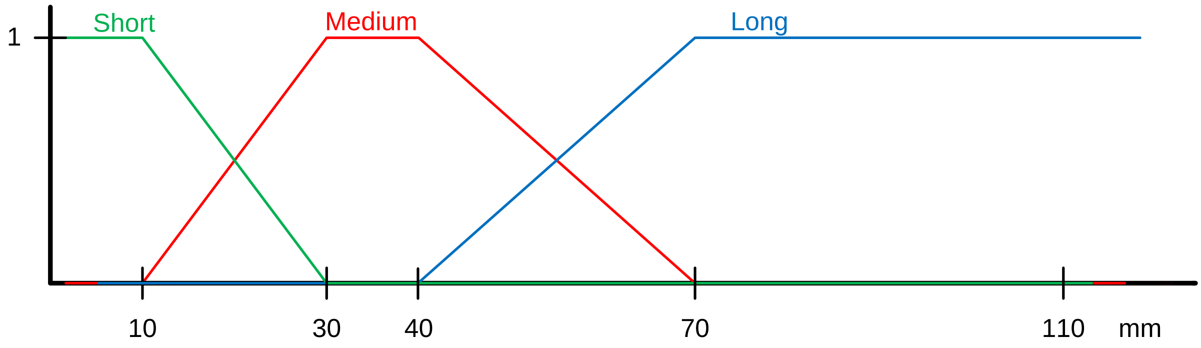



Figure 47: Cutting tool edge length membership functions. Values range took from the literature [MITSa13]

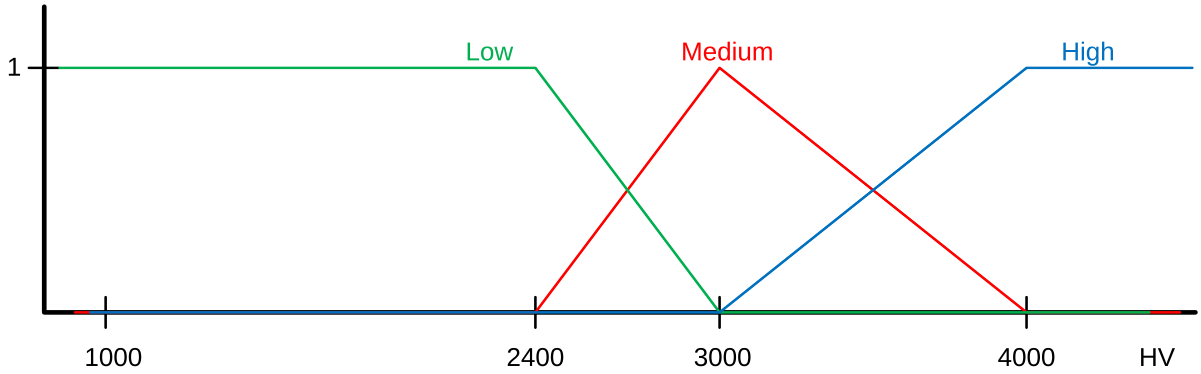



Figure 48: Cutting tool hardness membership functions. Values range took from the literature [MITSa13]

As regards the inference rules, four characteristics each one with three classes mean eighty-one rules. In the following table there are all the rules with their respective assignments.

| Cutting | | | | |
|---|---|---|---|---|
| **Tool Hardness** | **Diameter** | **Edge Length** | **Material Hardness** | **Diagram** |
| Low | Small | Short | Low | 2 |
| Low | Small | Short | Medium | 3 |
| Low | Small | Short | High | 3 |
| Low | Small | Medium | Low | 5 |
| Low | Small | Medium | Medium | 6 |
| Low | Small | Medium | High | 6 |
| Low | Small | Long | Low | 8 |
| Low | Small | Long | Medium | 9 |
| Low | Small | Long | High | 9 |
| Low | Medium | Short | Low | 5 |
| Low | Medium | Short | Medium | 6 |
| Low | Medium | Short | High | 6 |
| Low | Medium | Medium | Low | 8 |
| Low | Medium | Medium | Medium | 9 |
| Low | Medium | Medium | High | 9 |
| Low | Medium | Long | Low | 11 |
| Low | Medium | Long | Medium | 12 |
| Low | Medium | Long | High | 12 |
| Low | Big | Short | Low | 8 |
| Low | Big | Short | Medium | 9 |
| Low | Big | Short | High | 9 |
| Low | Big | Medium | Low | 11 |
| Low | Big | Medium | Medium | 12 |
| Low | Big | Medium | High | 12 |
| Low | Big | Long | Low | 14 |
| Low | Big | Long | Medium | 15 |
| Low | Big | Long | High | 15 |
| Medium | Small | Short | Low | 1 |
| Medium | Small | Short | Medium | 2 |
| Medium | Small | Short | High | 3 |
| Medium | Small | Medium | Low | 4 |
| Medium | Small | Medium | Medium | 5 |
| Medium | Small | Medium | High | 6 |
| Medium | Small | Long | Low | 7 |
| Medium | Small | Long | Medium | 8 |
| Medium | Small | Long | High | 9 |
| Medium | Medium | Short | Low | 4 |
| Medium | Medium | Short | Medium | 5 |
| Medium | Medium | Short | High | 6 |
| Medium | Medium | Medium | Low | 7 |
| Medium | Medium | Medium | Medium | 8 |
| Medium | Medium | Medium | High | 9 |

| Tool Hardness | Diameter | Edge Length | Material Hardness | Diagram |
|---|---|---|---|---|
| Medium | Medium | Long | Low | 10 |
| Medium | Medium | Long | Medium | 11 |
| Medium | Medium | Long | High | 12 |
| Medium | Big | Short | Low | 7 |
| Medium | Big | Short | Medium | 8 |
| Medium | Big | Short | High | 9 |
| Medium | Big | Medium | Low | 10 |
| Medium | Big | Medium | Medium | 11 |
| Medium | Big | Medium | High | 12 |
| Medium | Big | Long | Low | 13 |
| Medium | Big | Long | Medium | 14 |
| Medium | Big | Long | High | 15 |
| High | Small | Short | Low | 1 |
| High | Small | Short | Medium | 1 |
| High | Small | Short | High | 2 |
| High | Small | Medium | Low | 3 |
| High | Small | Medium | Medium | 3 |
| High | Small | Medium | High | 4 |
| High | Small | Long | Low | 7 |
| High | Small | Long | Medium | 7 |
| High | Small | Long | High | 8 |
| High | Medium | Short | Low | 4 |
| High | Medium | Short | Medium | 4 |
| High | Medium | Short | High | 5 |
| High | Medium | Medium | Low | 7 |
| High | Medium | Medium | Medium | 7 |
| High | Medium | Medium | High | 8 |
| High | Medium | Long | Low | 10 |
| High | Medium | Long | Medium | 10 |
| High | Medium | Long | High | 11 |
| High | Big | Short | Low | 7 |
| High | Big | Short | Medium | 7 |
| High | Big | Short | High | 8 |
| High | Big | Medium | Low | 10 |
| High | Big | Medium | Medium | 10 |
| High | Big | Medium | High | 11 |
| High | Big | Long | Low | 13 |
| High | Big | Long | Medium | 13 |
| High | Big | Long | High | 14 |

Table 7: cutting tool inference rules and assigned diagrams

In the assignment to diagrams, on the one hand three different levels of ratio between cutting tool and workpiece hardness have been considered as they have a fundamental influence on permitted speed and feed of the operation, and on the other hand five different classes depending on tool edge length and diameter have been considered as they specify the contact surface between tool and workpiece. The intersection between these classes has generated 15 different possibilities that will form defuzzification diagram. Diagram number has to be analyzed as follows: there are five classes ([1,3], [4,6], [7,9], [10,12], [13,15]) representing the contact surface, within each class the hardness ratio identifies the speed and feed category; e. g. diagram number 1 has a small contact surface and the speed and feed class is the lower, while diagram number 3 has also a small contact surface but with a high speed and feed.

### 4.2.4 Definition of the assignment algorithm for the cooling system

As regards the cooling system expansion fluid temperature, environment temperature and cooling capacity have to be analyzed and divided in classes. While for cooling capacity and fluid expansion temperature there are not problems as their limit values can be found in the literature [HYDA07], this not happens for the environment temperature, thus as high limit value, 40°C has been chosen because over this threshold, analyzed drives might suffer failures and malfunctioning [SIEM07]. Also in this diagrams, an asymmetric membership function has been preferred to have more balanced classes. Analyzed characteristics diagrams with their respective functions are shown below.

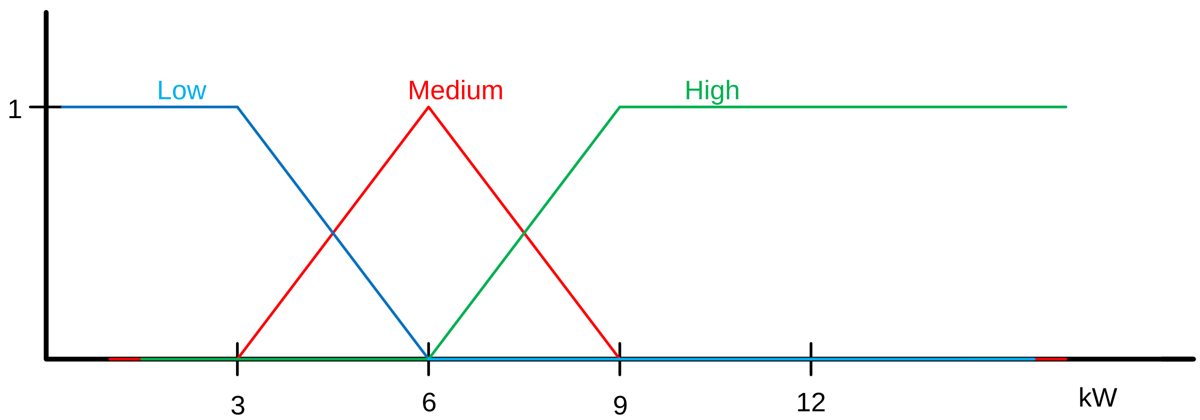



Figure 49: Cooling capacity membership functions

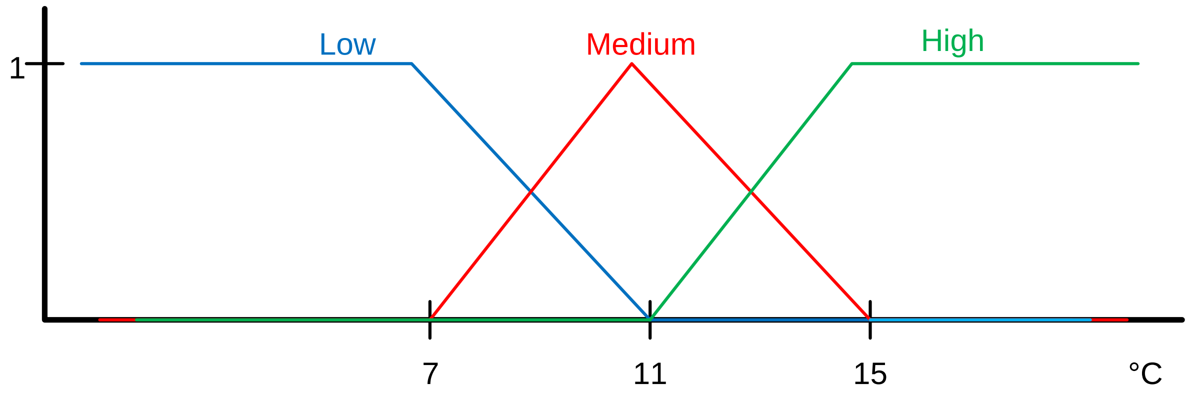



Figure 50: Fluid expansion temperature membership functions

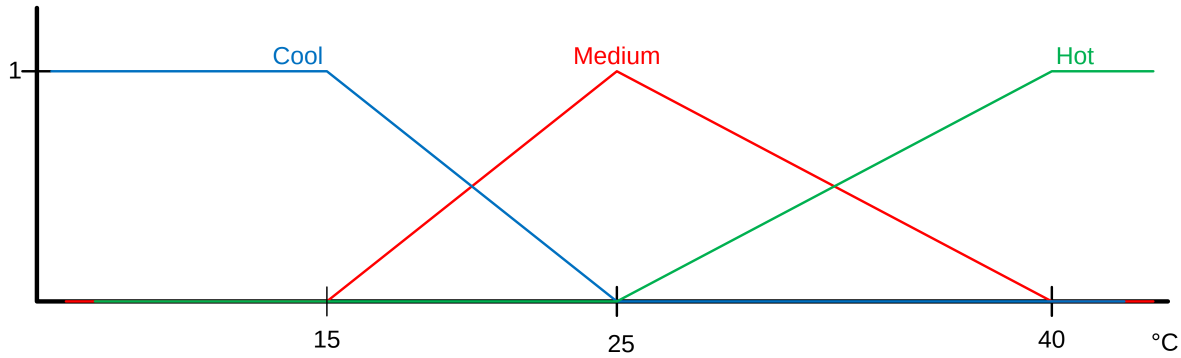



Figure 51: Environment temperature membership functions

As well as other cases, the number of the rules is given by the multiplication between the number of classes of each category, 27 inference rules have to be filed in. In cooling system case, for the assignment to diagrams and to reduce diagrams quantity, on the one hand cooling capacity has been considered as an independent information and divided in three classes, these express the power of the cooling system; on the other hand both environment and expansion fluid temperature are responsible of cooling system's performance, the relation between these temperature levels states, within the power category, the performance class. Having divided that class in three performance level, so the assignment can choose between 9 diagrams, the numbers have to be seen that the ranges [1,3], [4,6], [7,9] specify the power class and within that range the number specify the relation between expansion fluid and environment temperature; also here a superior number matches with a higher performance. The following table shows all the rules and their assignment.

| Cooling | | | |
|---|---|---|---|
| Cooling Capacity | Expansion Fluid Temperature | Environment Temperature | Diagram |
| Low | Low | Cool | 2 |
| Low | Low | Medium | 1 |
| Low | Low | Hot | 1 |
| Low | Medium | Cool | 3 |
| Low | Medium | Medium | 2 |
| Low | Medium | Hot | 2 |
| Low | High | Cool | 3 |
| Low | High | Medium | 3 |
| Low | High | Hot | 2 |
| Medium | Low | Cool | 5 |
| Medium | Low | Medium | 4 |
| Medium | Low | Hot | 4 |
| Medium | Medium | Cool | 6 |
| Medium | Medium | Medium | 5 |
| Medium | Medium | Hot | 5 |
| Medium | High | Cool | 6 |
| Medium | High | Medium | 6 |
| Medium | High | Hot | 5 |
| High | Low | Cool | 8 |
| High | Low | Medium | 7 |
| High | Low | Hot | 7 |
| High | Medium | Cool | 9 |
| High | Medium | Medium | 8 |
| High | Medium | Hot | 8 |
| High | High | Cool | 9 |
| High | High | Medium | 9 |
| High | High | Hot | 8 |

Table 8: Cooling system inference rules and assigned diagrams

### 4.2.5 Defining fuzzy operators and defuzzifications methods for the concept

Once that each component has been analyzed, before starting the analysis of a given scenario, determining fuzzy operators and defuzzificaton method is necessary.

How it is explain in the paragraph regarding fuzzy logic, there is not a best fuzzy operator; generally they lead to same or similar results but also some prior tests using both operator might be appropriate to choice more carefully one of them.

Whereas as concerns defuzzification method choice, the issue is more complicated. Indeed, as it was mentioned in the introduction, usually the preferred method is the COG one, that is the centre of gravity method. Nevertheless, in this application, this method can lead to an unclear or conflicting outcome. This problem arises as in this case the requested result is not expressed on the x-axis but by defuzzification functions themselves. That concept is explained more clearly with the help of an example. The figure below shows the defuzzification graph related to synchronous motor.

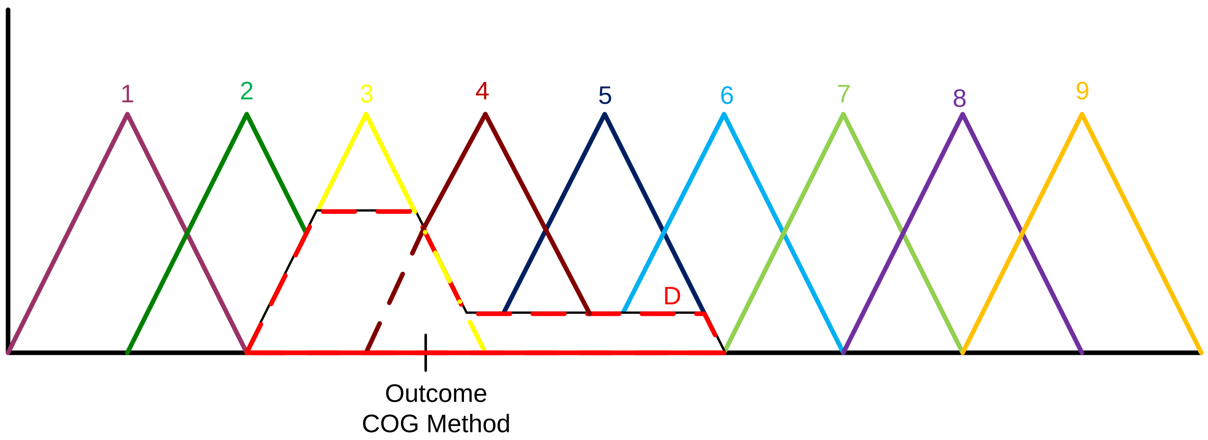



Figure 52: Example of synchronous motor output using center of gravity method

Supposing that the red figure is the result of all inference rules, using the COG method this will return the value shows in the illustration. However with this outcome, it is not possible to understand if that drive has to be assigned to diagram 3 or diagram 4. Instead maximum value method suggests to take the minimum or maximum value for which D is maximum. As a value on x-axis is not requested, it is possible to modify this method, considering that the diagram that will be assigned is the one corresponding at the number where the function presents its maximum and so taking the diagram that have a greater truth degree. Thus, in this case there would not be problems in assigning the considered drive to diagram number 3.

At this point another reasoning could be appropriate, indeed in spite of there is not usually a best fuzzy operator, adopting this defuzzification method, the use the $t_a$ operator could be better as it gives a greater weight to all items of the rules. This can help to avoid that two rules with same “if” and different “then” present the same truth degree. If for instance a rule has its entries with a truth degree of 0.8 and 0.4 and a second rule has its ones with values equal to 0.5 and 0.4, using an operator $t = \wedge$, the result would be:

$$rule_1 = 0.4$$

$$rule_2 = 0.4$$

Thus in this case, the two rules have the same truth degree for different “then” and using that defuzzification method this can lead to some problems.

Instead using the other operator $t = \ t_a$ the result would be rather different:

$$rule_1 = 0.32$$

$$rule_2 = 0.20$$

Supposing that the $rule_1$ is related to diagram number 2 and $rule_2$ to diagram number 3. The graph below clears the difference between the two operator usage.

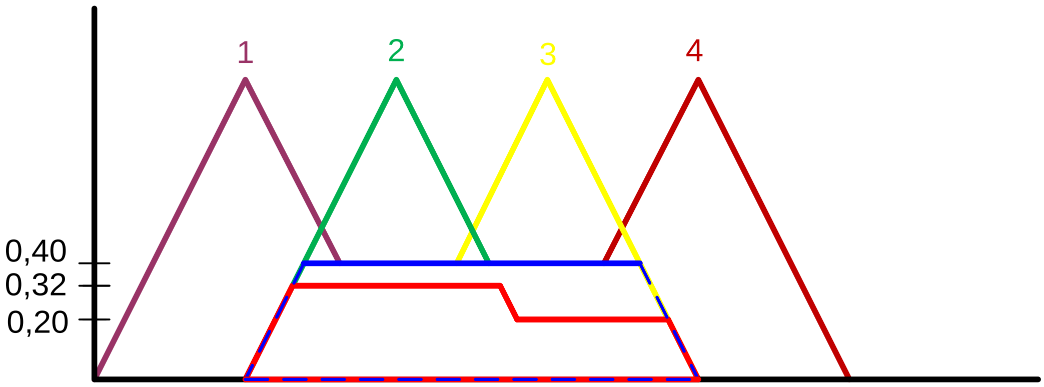



Figure 53: Different results depending on used fuzzy operator

The blue figure represents the output in case $\wedge$ operator has been used. It is easy to understand that in this instance, the assignment is not clear because the two diagrams are equivalent. Whereas red figure shows the result using $t_a$ operator. In this case the assignment would be clearly to diagram 2. Thus if the maximum value defuzzification method is used, and in this type of application it is preferred, the $t_a$ fuzzy operator is highly recommended.

## 4.3 Summary

This chapter presented an algorithm to assign components to drive map that describes its behaviour better. This process is particularly useful for those cases with a lack of real-time data regarding component's behaviour and its energy consumption, generally coming from energy logger or its manufacturer. Moreover this algorithm can be used to improve the efficiency of the energy consumption process implementing it to reduce the amount of data. This is made possible using this algorithm to integrate the trend of more components in an only one diagram basing on components architectural characteristics.

The first part of the chapter shows characteristics that are selected and explains why and their influence on diagram trend. The starting basis in the application scenario that considers for any machines drive, gear, cutting tool and cooling system as paramount components to figure out machine tool energy consumption.

Whereas the second part, basing on selected characteristics, presents how the algorithm should be applied to each component. The algorithm uses fuzzy logic to divide components characteristics in different performance classes and following to make the assignment.

In the next chapter assignment algorithm will be evaluated by means of a structured walkthrough that shows how and why some diagrams will be selected. Furthermore the applicability of that approach will be discussed through an expert dialog and its results will be shown and commented.

# 5 Evaluation

Usually the evaluation should include both validation and verification. The first one points to assure that a product, service, process or system meets the needs of the customer and other identified stakeholders; it often involves acceptance and suitability with external customers. Whereas the latter can be considered as the evaluation of whether or not a product, service, process or system complies with regulation, requirement, specification, or imposed condition; it is often an internal process [WIKI14].

As this algorithm has not been implemented yet and it is nearly impossible to evaluate a concept in its entirety without implementing it, what is possible to do is to validate the concept logically, showing that the developed concept can be taken for the aim of its development.

Validation of an algorithm or a concept is so important as it can lead, throught analysis and study of its steps and phases, to finding out troubles, problems and mistakes and related solutions or other ways to proceed before it is really implemented, saving time and costs.

## 5.1 Evaluation Procedure

The evaluation process will be carried on by means of a structured walkthrough that has the aim to show how some diagrams will be selected or mixed and by means of an experts dialog with the goal of identifying possible inconsisenticies, mistakes and improvements. With the expert dialog both application scenario analysis and fuzzy process will be evaluated, with a special focus on the definition and choice of the characteristics for each component. This concept has been analized by machine tools experts, in particular by Karl Doreth, engineer at IFW of Hannover, and by the manufacturer MAG Europe in Eislingen fils. As previously said, they especially analized and judged components' characteristics choice.

### 5.1.1 Structured walkthrough of the algorithm

As previously said, it is impossible to evaluate the concept debated in its entirety as it has not been implemented yet, but it is possible to validate it logically following for instance the concept in each step with a concrete example.

Below a real case is presented and discussed with the goal of guiding through it. Presuming that there are not data available from any data logger, it is required to assign each component to the diagram that represents its behaviour better.

In table 9 the actual application scenario is shown.

| Component | Main Drive | Drive for Axes | Ball Screw | Cutting Tool | Cooling System |
|---|---|---|---|---|---|
| Characteristic 1 | Rated Current = 38 A | Rated Current = 7,2 A | Lenght = 4500 mm | Hardness = 3200 HV | Capacity = 4 kW |
| Characteristic 2 | Rated Torque = 38 Nm | Rated Torque = 9,3 Nm | Diameter = 50 mm | Edge Diameter = 16 mm | Environment Temperature = 27 °C |
| Characteristic 3 | Rated Speed = 3000 rpm | Rated Speed = 3000 rpm | Pitch = 10 mm | Edge Lenght = 50 mm | Expansion Fluid Temperature = 12 °C |

Table 9: Application scenario for the evaluation of the concept

Furthermore it must be said that it is supposed to work a piece of pre-tempered steel with a hardness of 48 HRC.

Once that the application scenario has been defined, now the analysis will be separated for each component. Starting from the first one, the main drive, taking its related data and inserting them into their appropriate diagram is required, in fact the first step is to understand for each characteristic which membership functions it fulfils and its membership degrees.

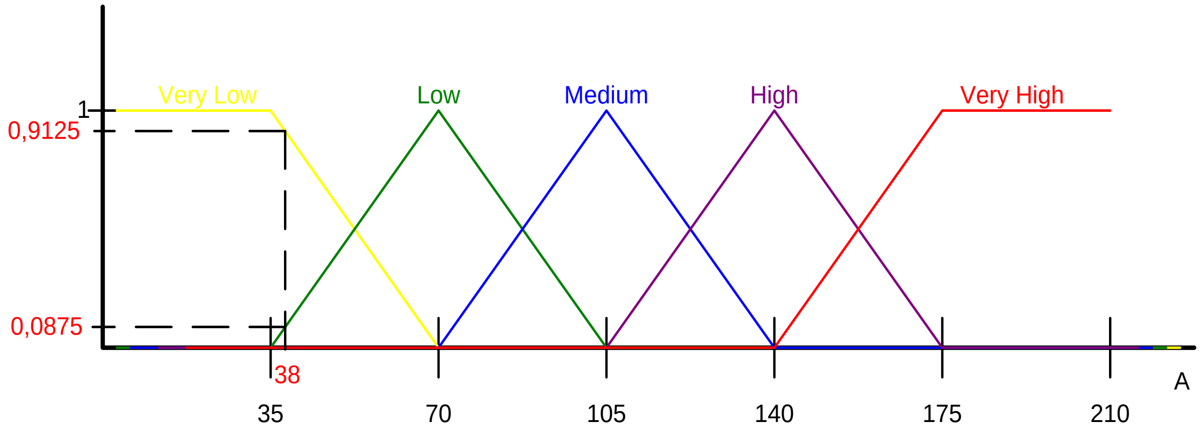



Figure 54: Determination of membership functions and their membership degrees for the given rated current of the main drive

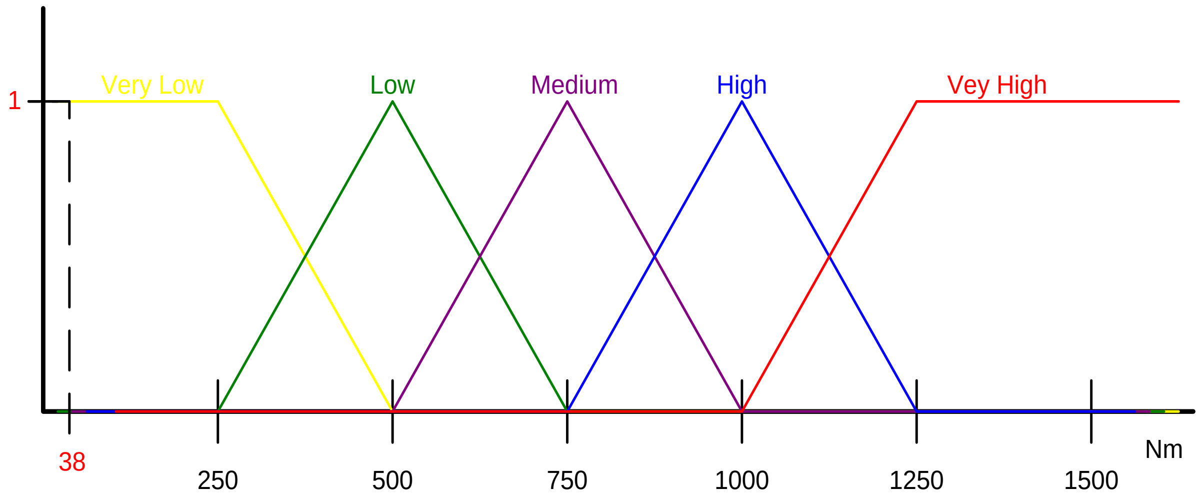



Figure 55: Determination of membership functions and their membership degrees for the given rated torque of the main drive

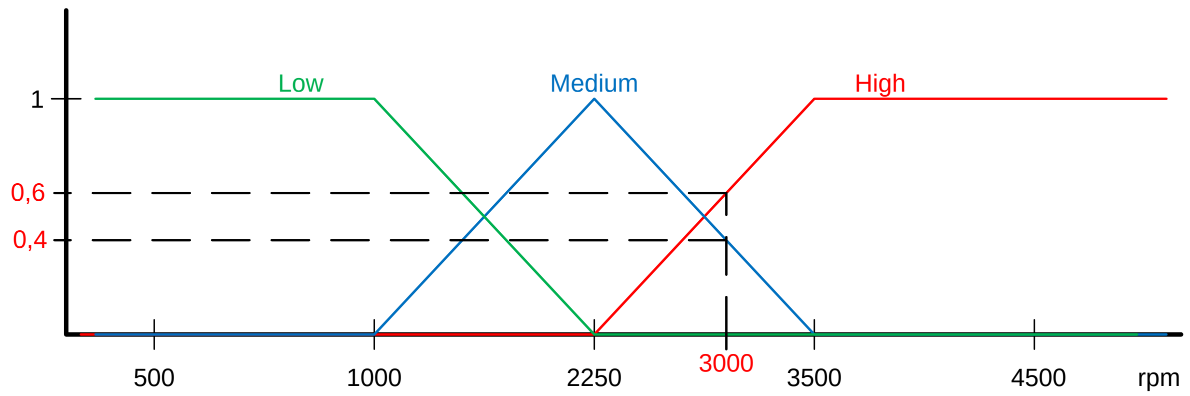



Figure 56: Determination of membership functions and their membership degrees for the given rated speed of the main drive

Once that all data have been inserted, membership degrees for each membership function fulfilled have been determined graphically.

Now proceeding with the second step is possible. Of the main drive seventy-five rules only those present a membership degree different from zero for each characteristic have to be selected. Generally the number of rules, that has to be selected, is the multiplication of the membership functions fulfilled for each characteristic, so in this case the number of rules is: 2 x 1 x 2 (rated current can be considered as very low or low, only very low for the rated torque and medium or high rated speed). The table below shows the selected rules with their different outputs.

| Asynchronous Motor | | | | |
|---|---|---|---|---|
| **Rule Number** | **Rated Current** | **Rated Torque** | **Rated Speed** | **Diagram** |
| 1 | Very Low | Very Low | Medium | 22 |
| 2 | Very Low | Very Low | High | 23 |
| 3 | Low | Very Low | Medium | 17 |
| 4 | Low | Very Low | High | 18 |

Table 10: Selected rules for the asynchronous motor case

At this point assigning the diagram, that represents the behaviour of the considered asynchronous motor better, is possible. Thus, basing on the considerations made in the paragraph 4.2.5, maximum value defuzzification method and $t_a$ as fuzzy operator will be used for the next step, the defuzzification of the rules and the following assignment of the component to a drive map. The implementation of the rules with those conditions provides the following results:

$$rule_1 = 0.9125 \times 1 \ \times 0.4 = 0{,}3650$$

$$rule_2 = 0.9125 \times 1 \ \times 0.6 = 0{,}5475$$

$$rule_3 = 0.0875 \times 1 \ \times 0.4 = 0{,}0350$$

$$rule_4 = 0.0875 \times 1 \ \times 0.6 = 0{,}0525$$

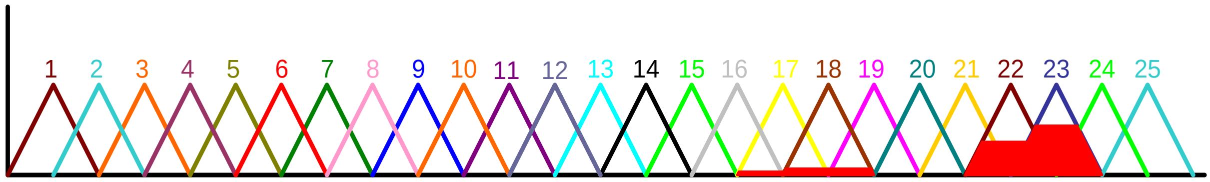



Figure 57: Defuzzification output of the asynchronous motor and its related assignment

The two red figures are the output of the implementation of all the rules, so using maximum value method is clear that the assigned diagram is the diagram labelled with number 23.

The same reasoning is required to assign the drive for axes, a synchronous motor, to a drive map. Thus, the first thing that has to be done is to insert the drive parameters in the fuzzyfication diagrams.

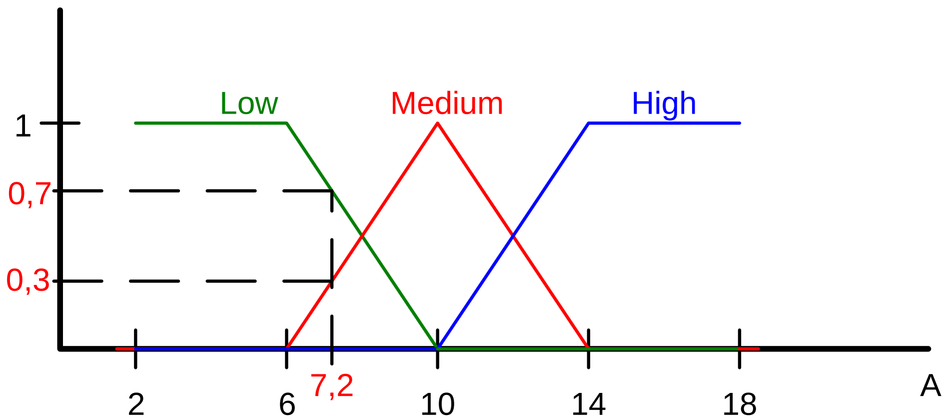



Figure 58: Determination of membership functions and their membership degrees for the given rated current of the drive for axes

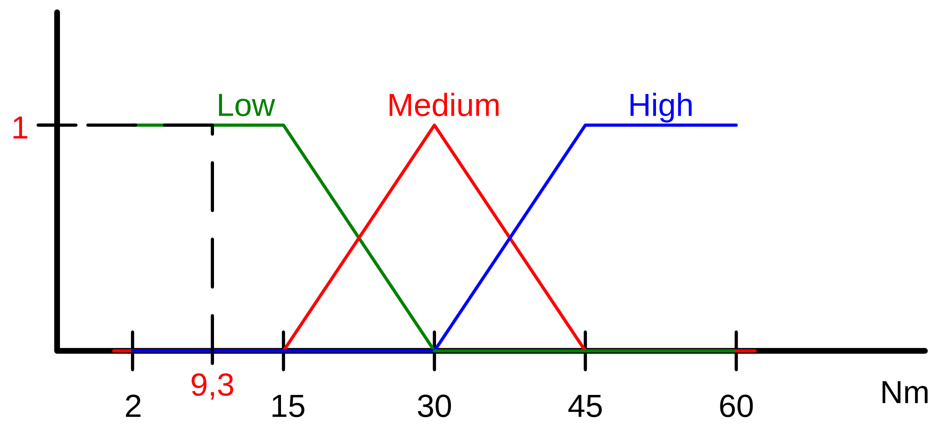



Figure 59: Determination of membership functions and their membership degrees for the given rated torque of the drive for axes

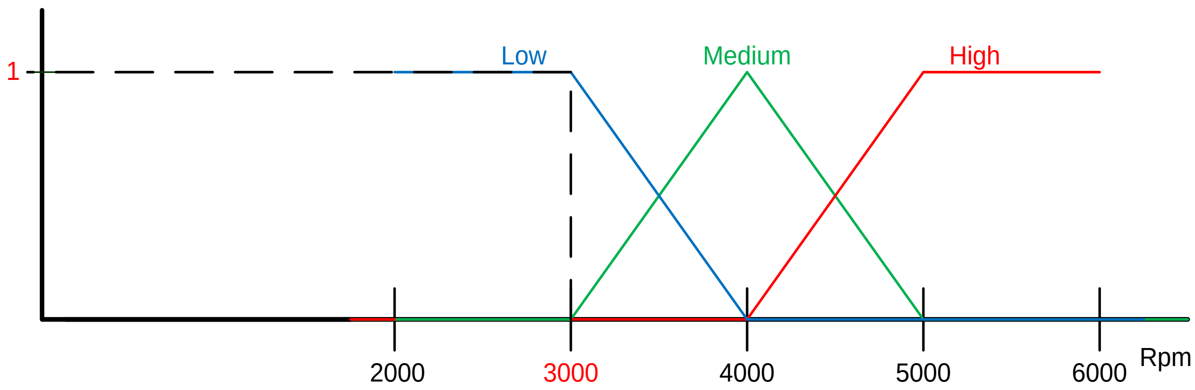



Figure 60: Determination of membership functions and their membership degrees for the given rated speed of the drive for axes

Now that all the membership degrees of the inserted parameters have been identified is possible to proceed with rules implementation. Only two rules can be applied because two of the three characteristics belong only to one membership function. The rules can be directly written as:

$$rule_1 = if\ the\ rated\ current\ is\ low\ and\ the\ rated\ torque\ is\ low$$
$$and\ the\ rated\ speed\ is\ low,\ then\ the\ drive\ has\ to\ be\ assigned\ to\ diagram\ number\ 7$$

$$rule_2 = if\ the\ rated\ current\ is\ medium\ and\ the\ rated\ torque\ is\ low$$
$$and\ the\ rated\ speed\ is\ low,\ then\ the\ drive\ has\ to\ be\ assigned\ to\ diagram\ number\ 4$$

At this point the defuzzification of the rules is necessary, using as before the maximum value method and as $t_a$ fuzzy operator. Below the rules results and the defuzzification diagram are reported.

$$rule_1 = 0.7 \times 1 \ \times 1 = 0{,}7$$

$$rule_2 = 0.3 \times 1 \ \times 1 = 0{,}3$$

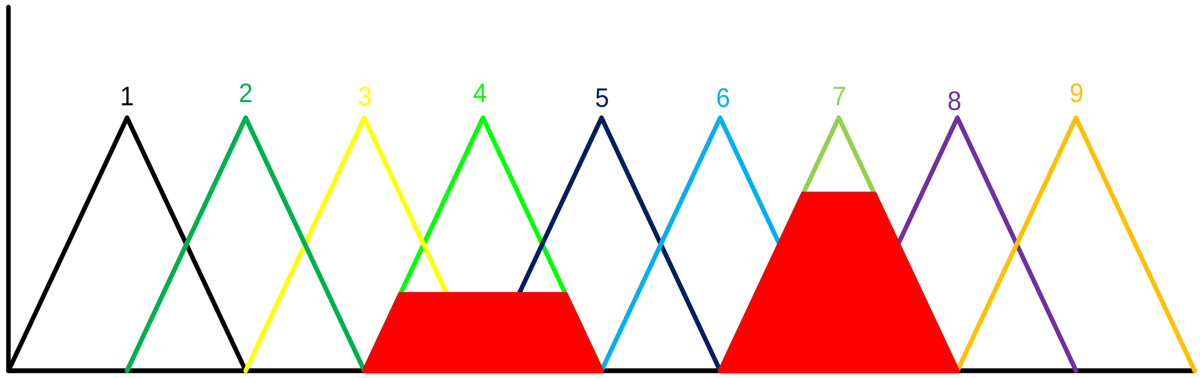



Figure 61: Defuzzification output of the synchronous motor and its related assignment

Also in this case the result is rather clear: although the diagram number 4 could describe the drive behaviour pretty well, the assignment will certainly be at the diagram number 7.

After the two different type of motors have been evaluated and assigned to a drive map, it is possible to proceed with the analysis of the ball screw. In this case the three characteristics are: ball screw length, diameter and its pitch. As in the previous cases, first of all parameters values, extracted from the application scenario, are inserted in the fuzzification diagrams.

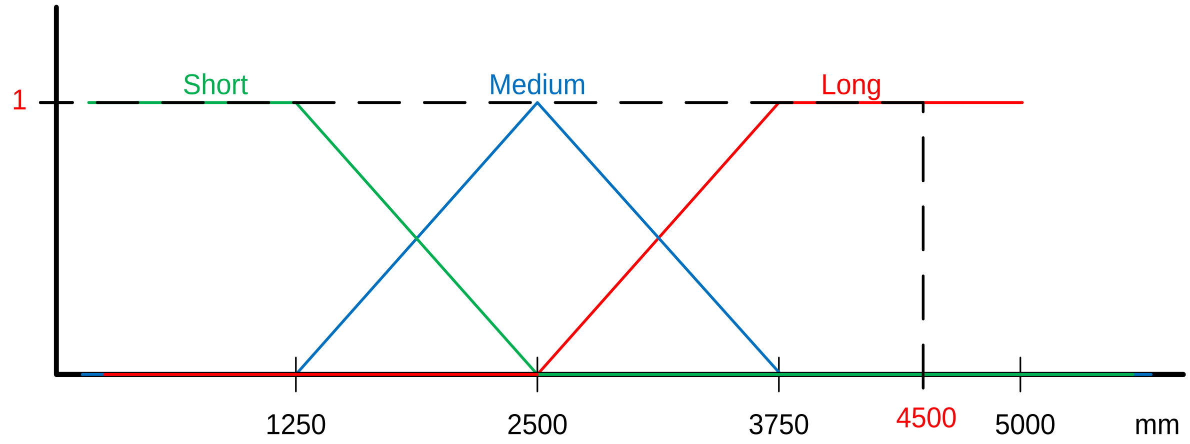



Figure 62: Determination of membership functions and their membership degrees for the given ball screw length

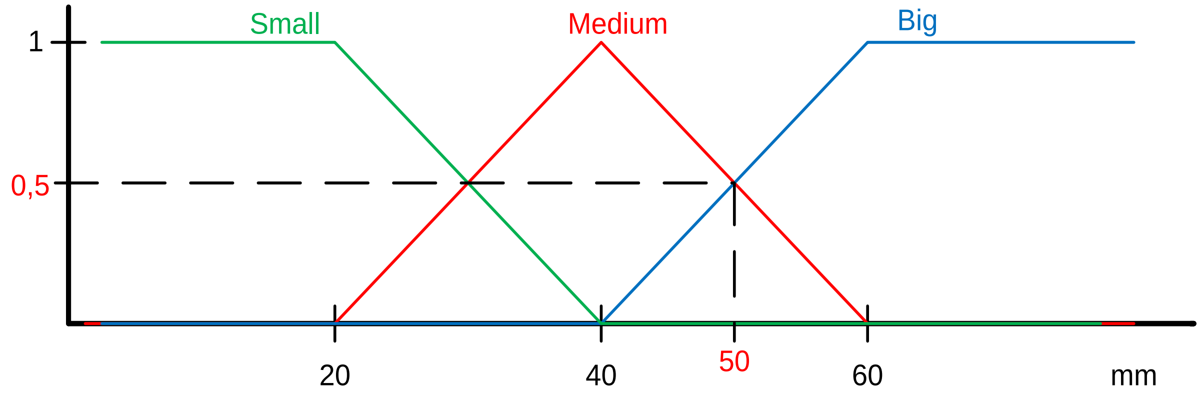



Figure 63: Determination of membership functions and their membership degrees for the given ball screw diameter

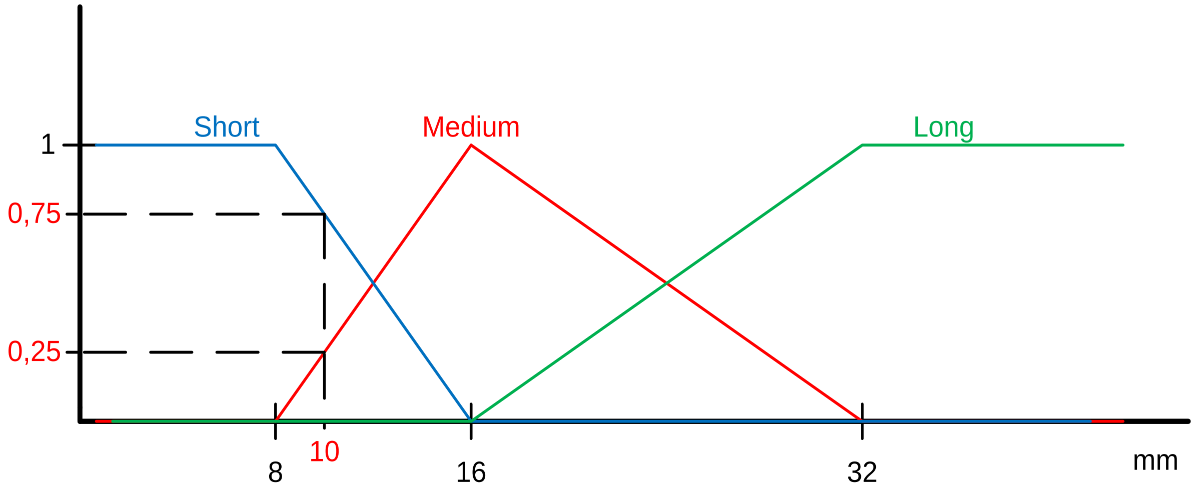



Figure 64: Determination of membership functions and their membership degrees for the given ball screw pitch

With a superficial look, it seems that only two rules are needed because the diagrams provide one value for the ball screw length, one value for the diameter and two values for the pitch, so it could be a case like synchronous motor instance (number of rules = 1 x 1 x 2 = 2). Instead the second diagram provides two equal values but belonging to two different membership functions: one belongs to the “medium” function and the other one to the “big” one. Thus the number of rules required is: 1 x 2 x 2 = 4. In the table below each rule with its label, inputs and output is reported.

| Ball Screw | | | | |
|---|---|---|---|---|
| **Rule Number** | **Length** | **Diameter** | **Pitch** | **Diagram** |
| 1 | Long | Big | Short | 8 |
| 2 | Long | Big | Medium | 8 |
| 3 | Long | Medium | Short | 5 |
| 4 | Long | Medium | Medium | 5 |

Table 11: Selected rules for the ball screw case

Once that it is possible to award a value to each rule input, thus it is possible to work out which diagram should be assigned to that ball screw. As in the other cases, the total value of each rule is calculated, using $t_a$ as fuzzy operator and consequently the defuzzification diagram is built, the component is assigned to a drive map by means of maximum value method.

$$rule_1 = 1 \times 0.5 \ \times 0.75 = 0.375$$

$$rule_2 = 1 \times 0.5 \ \times 0.25 = 0.125$$

$$rule_3 = 1 \times 0.5 \ \times 0.75 = 0.375$$

$$rule_4 = 1 \times 0.5 \ \times 0.25 = 0{,}125$$

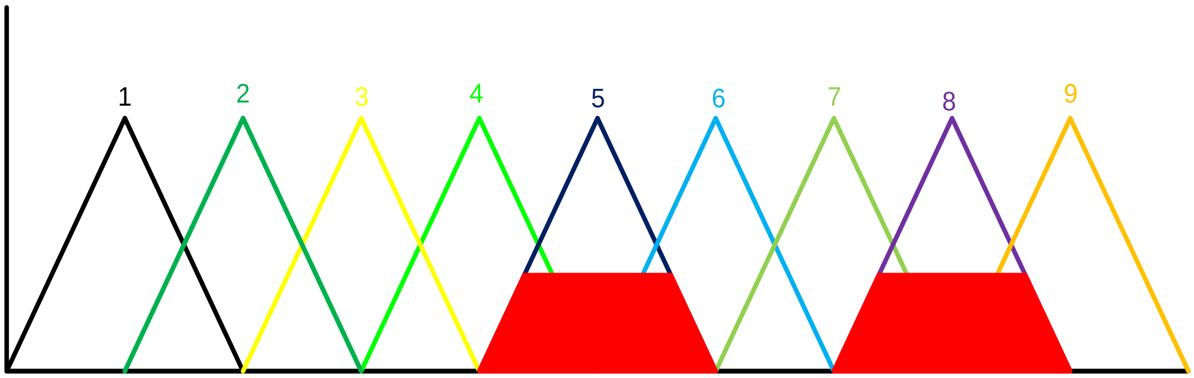



Figure 65: Defuzzification output of the ball screw and its related assignment

Apparently this case presents an unclear result: the assignment would seem uncertain because there are two diagrams that get same description value of component behaviour. Nevertheless if the rules, that provide this result, are analyzed, it is possible to figure out that the two rules have in common two of three value inputs (long length and short pitch) and the third input is different (diameter could be considered medium or big) but with same degree of truth (0.5 both for medium and big diameter). Thus in this case, as ball screw diameter affects only its load capacity, instead assigning the component to diagram number 5 or to number 8, the best thing to do is to take these two diagrams and mix them only on the basis of their load capacity finding out another diagram that has same behaviour regarding the permitted speed and a middle way behaviour relating to transferable torque.

Passing to cutting tool analysis, besides evaluating cutting parameters (cutting tool hardness, cutting tool edge diameter and length), considering workpiece hardness is necessary. So in this case there will be four fuzzification diagrams.

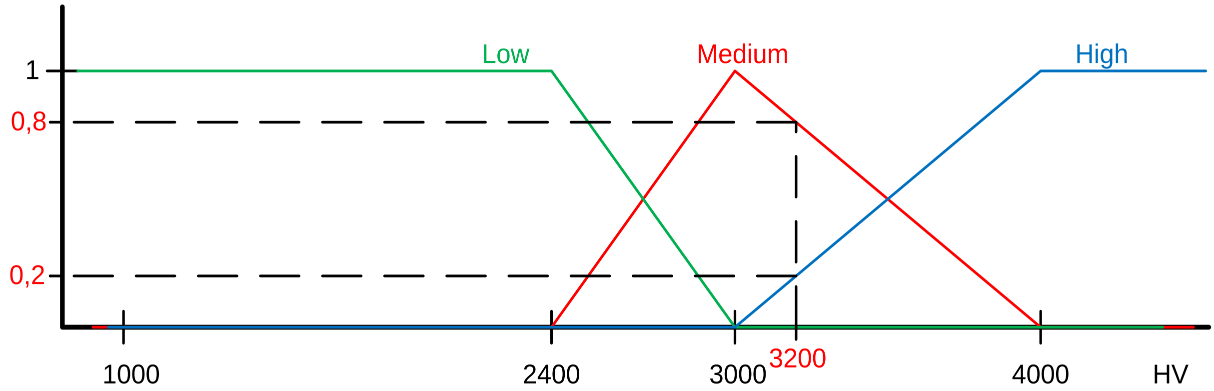



Figure 66: Determination of membership functions and their membership degrees for the given cutting tool hardness

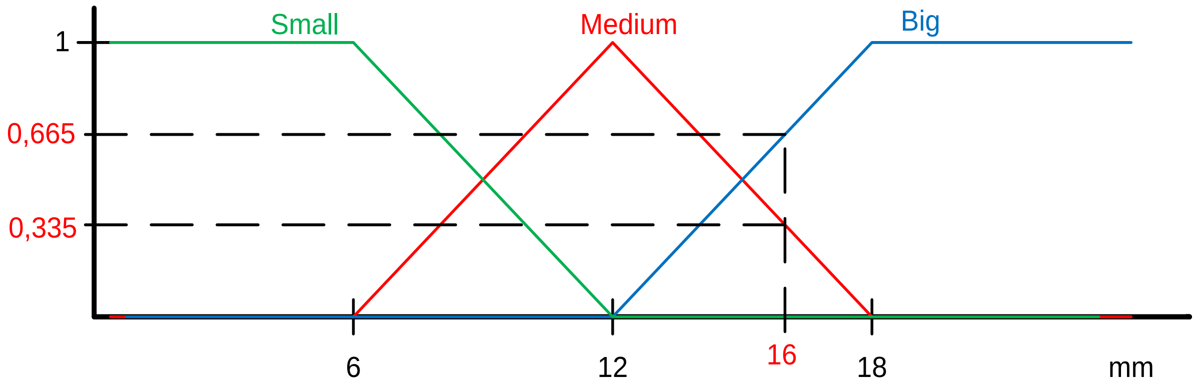



Figure 67: Determination of membership functions and their membership degrees for the given cutting tool edge diameter

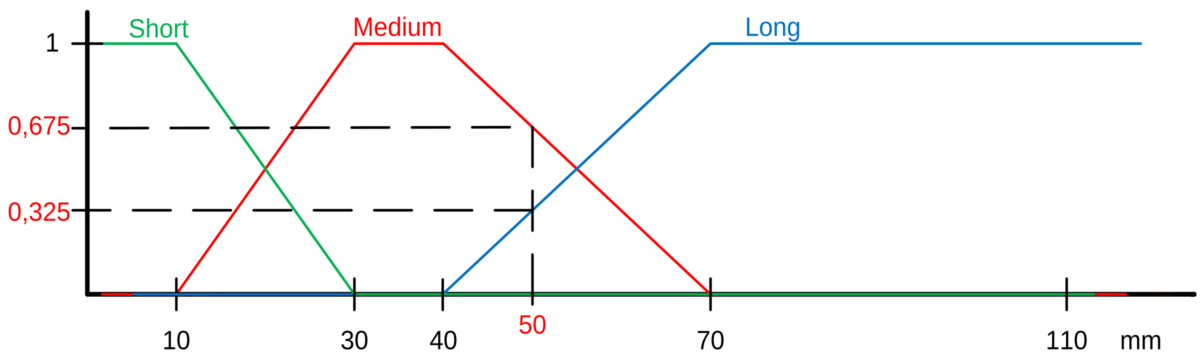



Figure 68: Determination of membership functions and their membership degrees for the given cutting tool edge length

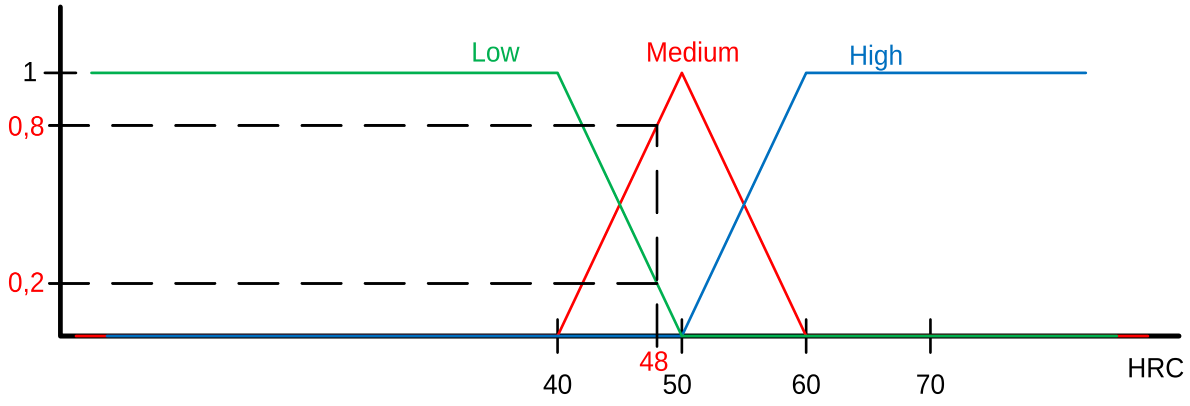



Figure 69: Determination of membership functions and their membership degrees for the given workpiece hardness

In this case, differently from the previous ones, there are more rules that have to be taken in account. In effect each component parameter belongs to two different membership functions so the total number of rules is sixteen. Below the table summarizes all the rules with their inputs and output.

| Cutting | | | | | |
|---|---|---|---|---|---|
| **Rule Number** | **Tool Hardness** | **Cutting Edge Diameter** | **Cutting Edge Length** | **Workpiece Material** | **Diagram** |
| 1 | Medium | Medium | Medium | Low | 7 |
| 2 | Medium | Medium | Medium | Medium | 8 |
| 3 | Medium | Medium | Long | Low | 10 |
| 4 | Medium | Medium | Long | Medium | 11 |
| 5 | Medium | Big | Medium | Low | 10 |
| 6 | Medium | Big | Medium | Medium | 11 |
| 7 | Medium | Big | Long | Low | 13 |
| 8 | Medium | Big | Long | Medium | 14 |
| 9 | High | Medium | Medium | Low | 7 |
| 10 | High | Medium | Medium | Medium | 7 |
| 11 | High | Medium | Long | Low | 10 |
| 12 | High | Medium | Long | Medium | 10 |
| 13 | High | Big | Medium | Low | 10 |
| 14 | High | Big | Medium | Medium | 10 |
| 15 | High | Big | Long | Low | 13 |
| 16 | High | Big | Long | Medium | 13 |

Table 12: Selected rules for the cutting case

Now that all the rules are filled in, proceeding with their real value calculation is possible:

$$rule_1 = 0.8 \times 0.665 \times 0.675 \times 0.2 = 0.07182$$

$$rule_2 = 0.8 \times 0.665 \times 0.675 \times 0.8 = 0.28728$$

$$rule_3 = 0.8 \times 0.665 \times 0.225 \times 0.2 = 0.02394$$

$$rule_4 = 0.8 \times 0.665 \times 0.225 \times 0.8 = 0.09576$$

$$rule_5 = 0.8 \times 0.335 \times 0.675 \times 0.2 = 0.03618$$

$$rule_6 = 0.8 \times 0.335 \times 0.675 \times 0.8 = 0.14472$$

$$rule_7 = 0.8 \times 0.335 \times 0.225 \times 0.2 = 0.01206$$

$$rule_8 = 0.8 \times 0.335 \times 0.225 \times 0.8 = 0.04824$$

$$rule_9 = 0.2 \times 0.665 \times 0.675 \times 0.2 = 0.01795$$

$$rule_{10} = 0.2 \times 0.665 \times 0.675 \times 0.8 = 0.07182$$

$$rule_{11} = 0.2 \times 0.665 \times 0.225 \times 0.2 = 0.00598$$

$$rule_{12} = 0.2 \times 0.665 \times 0.225 \times 0.8 = 0.02394$$

$$rule_{13} = 0.2 \times 0.335 \times 0.675 \times 0.2 = 0.00904$$

$$rule_{14} = 0.2 \times 0.335 \times 0.675 \times 0.8 = 0.03618$$

$$rule_{15} = 0.2 \times 0.335 \times 0.225 \times 0.2 = 0.00301$$

$$rule_{16} = 0.2 \times 0.335 \times 0.225 \times 0.8 = 0.01206$$

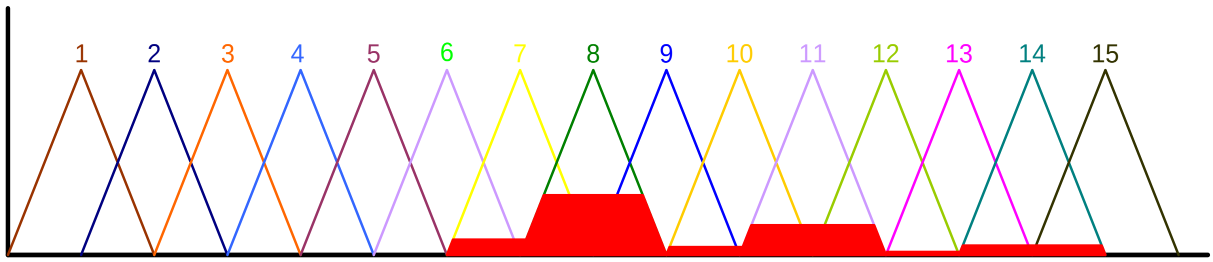



Figure 70: Defuzzification output of cutting operating condition and its related assignment

Both with the rules calculation and graphically, also in this case the assignment is rather clear: the diagram that should be assigned to these operating conditions (cutting tool and workpiece) is diagram number 8.

In the end, the last component that have to be assigned to a drive map is the cooling system. Cooling capacity, environment temperature and fluid expansion temperature are the chosen characteristics. As usual application scenario values have been inserted in fuzzification diagrams.

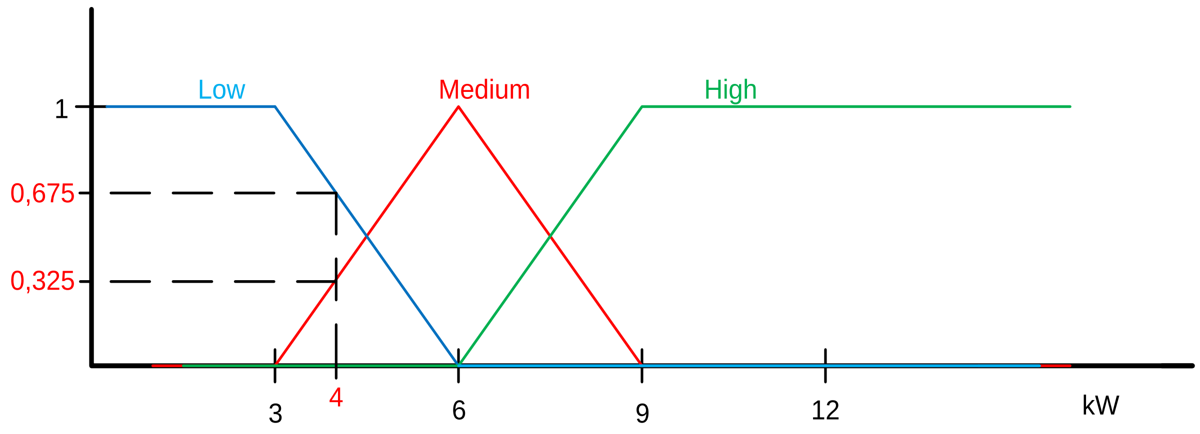



Figure 71: Determination of membership functions and their membership degrees for the given cooling capacity

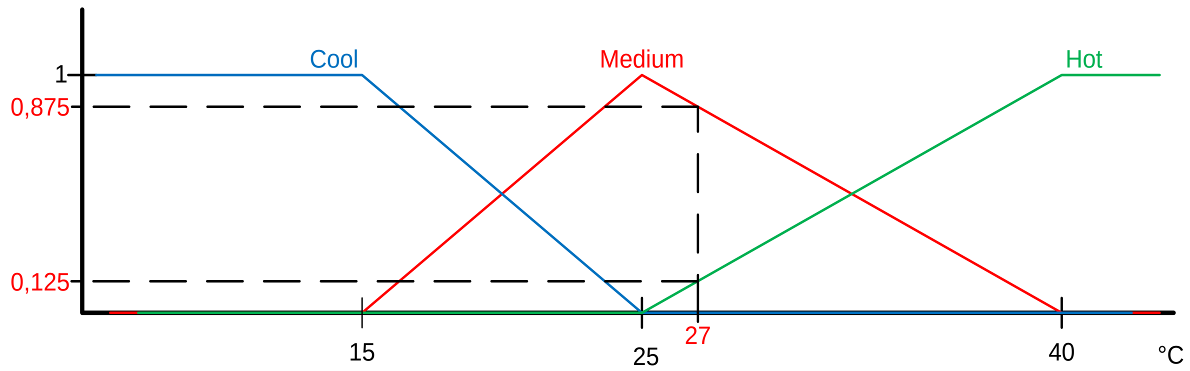



Figure 72: Determination of membership functions and their membership degrees for the given environment temperature

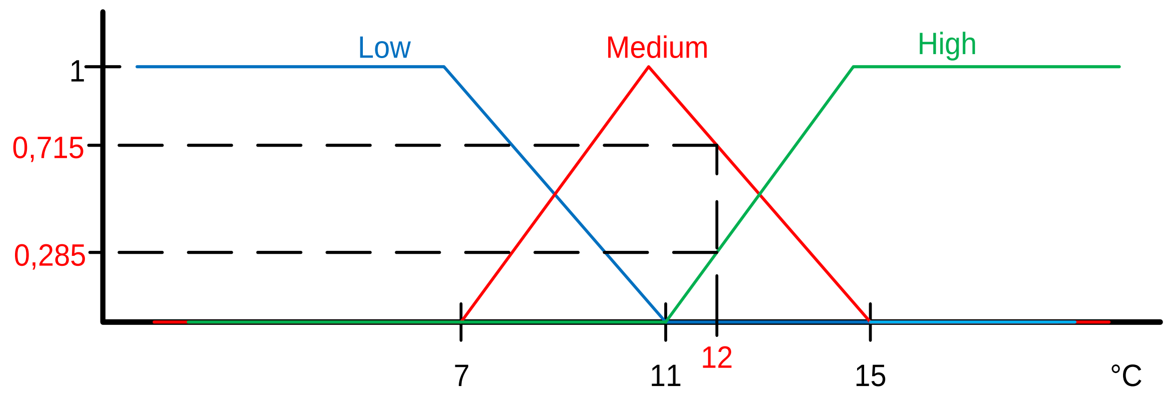



Figure 73: Determination of membership functions and their membership degrees for the given fluid expansion temperature

All characteristics belong to two membership functions, thus, since three features have been taken into account, there are height inference rules that have to be implemented. The table below shows them with their related assignments.

| Cooling System | | | | |
|---|---|---|---|---|
| **Rule Number** | **Cooling Capacity** | **Environment Temperature** | **Fluid Expansion Temperature** | **Diagram** |
| 1 | Low | Medium | Medium | 2 |
| 2 | Low | Medium | High | 3 |
| 3 | Low | Hot | Medium | 2 |
| 4 | Low | Hot | High | 2 |
| 5 | Medium | Medium | Medium | 5 |
| 6 | Medium | Medium | High | 6 |
| 7 | Medium | Hot | Medium | 5 |
| 8 | Medium | Hot | High | 5 |

Table 13: Selected rules for the cooling system case

At this point, it is possible to proceed with the last assignment. As in the previous cases for the defuzzification step, $t_a$ operator and maximum value method are used; with the results of rules implementation the assignment diagram has been built.

$$rule_1 = 0.675 \times 0.875 \ \times 0.715 = 0.4223$$

$$rule_2 = 0.675 \times 0.875 \ \times 0.285 = 0.1683$$

$$rule_3 = 0.675 \times 0.125 \ \times 0.715 = 0.0603$$

$$rule_4 = 0.675 \times 0.125 \ \times 0.285 = 0.0240$$

$$rule_5 = 0.325 \times 0.875 \ \times 0.715 = 0.2033$$

$$rule_6 = 0.325 \times 0.875 \ \times 0.285 = 0.0810$$

$$rule_7 = 0.325 \times 0.125 \ \times 0.715 = 0.0290$$

$$rule_8 = 0.325 \times 0.125 \ \times 0.285 = 0.0116$$

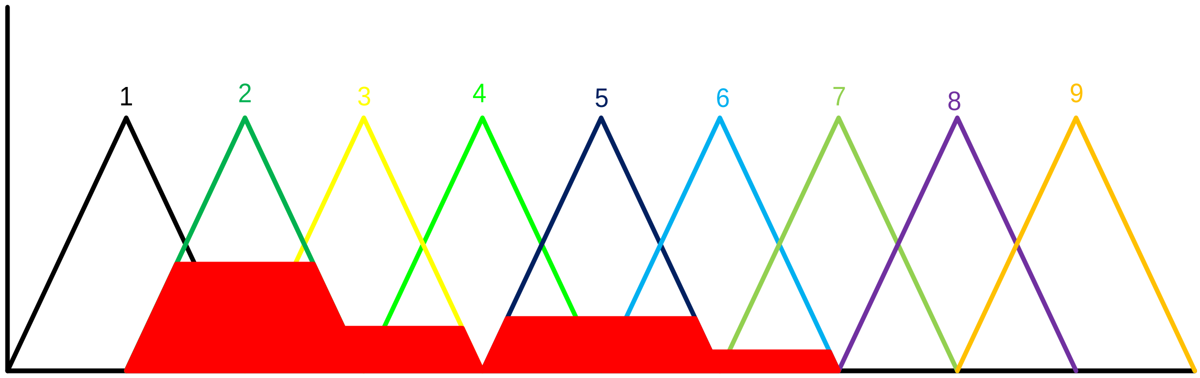



Figure 74: Defuzzification output of cooling system and its related assignment

Looking at the rules and the picture it is immediately understandable that cooling system should be assigned to diagram number 2.

With this last assignment the walkthought of the concept can be considered settled since each component belonging to the application scenario has been evaluated and assigned to a drive map.

## 5.2 Evaluation Results

The results of the evaluation process led to two major considerations: the walktrought of the algorithm based on fuzzy logic showed that it could be applicable to machine tools field and how facing and coping different situations while the experts dialog, with the engineer Karl Doreth and the manufacturer MAG Europe in Eislingen fils, led to consider the characteristics choice for each component belonging to the application scenario correct, but it highlighted that, within the component “drive”, probably also the operational state of the drive should be considered. On this matter for the moment there is not any certainty about that but further studies are needed. Moreover it must be said that issue does not change the concept, or better, the fact that also the operational state should be considered, does not affect the division in performance classes or in power classes of the drive since it can be considered an additional partition that has to be done a priori, before the classification of the drive, like the identification of drive type between main drive and drive for axes. In effect the study of the operational state could not be analyzed with fuzzy logic as the operational state cannot assume continous values but only pre-fixed values as for example it is shown below the states “S1”, “S2” or “S3”.

The convinction, that also operational state should be considered, comes from the circumstance that, as well as continuos duty, standard intermittent duty types are also defined to load duty cycles varying on periods. This involves operation that comprises a time with constant load and an off period. The figures below show the different values in both cases.

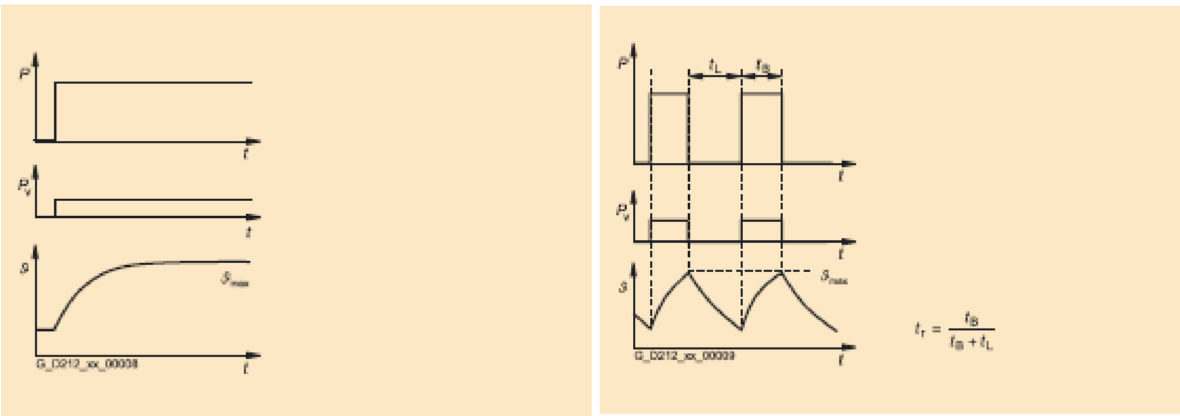



Figure 75: Continous operation and intermittent operation [SIEM07]

The operating conditions can be very different, that means the time with constant load can vary a lot, but anyway there are some referential standard fixed values [SIEM07]. Those more used are :

- S3 – 60%
- S3 – 40%
- S1 – 25%

The corresponding motor characteristics are provided for these specifications. The load torque must be below the corresponding thermal limiting characteristic curve of the motor. An overload must be taken into consideration to load duty cycles with varying on periods. The figure 76 illustrates an example of the different curves depending on those values.

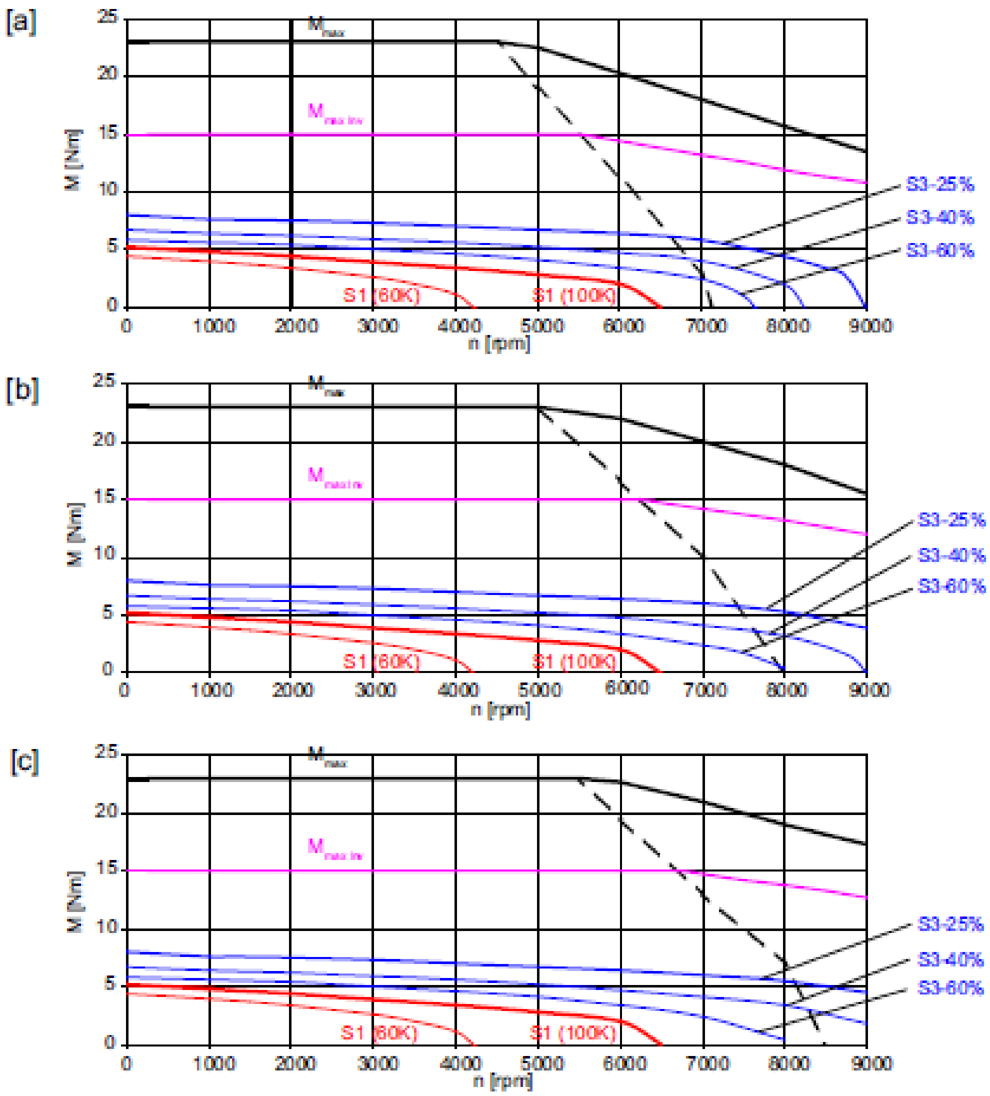



Figure 76: Example of torque (Nm) and speed (rpm) diagram with different curves depending on motor operation for a Siemens synchronous motor 1FT7044-AK7, a)Sinamics SLM 400V, b)Sinamics ALM 400V; c)Sinamics SLM 480V [SIEM07]

## 5.3 Summary

This chapter has been dedicated to evaluate the entire concept. Evaluation is usually divided into validation and verification but, as in this case the concept has not been implemented yet, only the former has been discussed and debated. It has been carried on in two parallel ways in order to validate two different aspects of the concept: on the one hand a walkthrough of the concept has been implemented to evaluate the assignment algorithm, on the other hand an experts dialog has been used to analyze characteristics choice.

Both of them showed relevant aspects that would be important to focus on: the experts dialog suggested that an analysis of the state of the drive should be led before drives classification; the walkthrough, instead, explained the concept with a real example of a possible application scenario.

As the concept has not been implemented yet, its verification is still needed; these considerations have been postponed to the chapter 6 “conclusion and outlook”. Indeed this chapter contains also reflections on the entire work and it proposes some ideas and advice for the future developments.

In the end some critical aspects should be introduced: some of them are embodied by the pitfalls and difficulties of fuzzy logic use as it is explained in paragraph 2.3.2, like e.g. membership functions selection and definition and fuzzy inference rules determination. Nevertheless there is another relevant aspect that should be analyzed: once that component X has been assigned to diagram Y, there is still a critical issue that remains, that is understanding between captured and stored diagrams which of these is the one that describes component X’s behaviour and trend better, in other words which is, between available diagrams, diagram Y. This further assignment can be done before or after fuzzy logic step, but it still represents a complex stage.

A possible solution, which should be studied and analyzed in more detail, could be employing an inverse reasoning: captured diagrams are counted and a numerical label will be assigned to each diagram depending on its behaviour. Just now components will be divided into performance classes and fuzzy membership function’s number will be determined in relations to the number of diagrams presented in database.

# 6 Conclusion and outlook

This chapter contains an overview of the entire work with its related conclusions. It is summarized why an assignment of components to drive maps is required, what it can lead at and some ideas for future works are proposed.

## 6.1 Conclusion

Within this work a concept was presented, to assign machine tool components to drive maps that trace their behaviour in order to calculate their energy consumption. This approach comes to help every time that it is not possible building a diagram related to each component as new developed components have never been in industrial use before, and so it is impossible capturing empirical information within an application scenario. Therefore through this approach during the design phase predicting the energy consumption of new planned machine tools and their components within the lifecycle is allowed.

This concept can be divided into two different part. First of all it must be said that the energy consumption of each machine tool depends on the application scenario. So the fist part of the approach has to inspect all the components belonging to the application scenario: drives and their heat production, ball-screws, cutting and cooling systems. The components' analysis should figure out which technical and architectural characteristics have to be considered to make the assignment for each component. This first part has been examined and then validated through an expert dialog.

The second part of the concept concerns the successive fuzzy logic based assignment. For each determined characteristic in the previous analysis, the range of values that can assume has to be studied and calculated. This range will be then expressed into membership functions through the use of fuzzy linguistic variables that get the advantage of being very similar to human way of thinking. After the fuzzification process, inference rules, that assign a different diagram for each set of membership functions, are built. Through the defuzzification process the assigned diagram for that given application scenario is determined. These drive maps can be used in order to optimize new designed machine tools within an assigned application scenario, in terms of the energy consumption before the machine is assembled and operating. Thus, following the explanation of this concept, working out and assigning to each different application scenario a related energy consumption is possible

## 6.2 Future Developments

Many reflections and observations can be done about future works, but first of all it should talk about the implementation of the concept. As it is explained in the chapter 5, the concept has not been implemented yet and the consequence is that only a logical validation is possible. So, in the further developments, the concept should be implemented through several tests as it can make possible also a verification of its well-functioning.

Verification could discover and highlight situations where the concept can be particularly useful or lead to take wrong decisions. There are many ways to verify a process, in this case an evaluation method that can be used is the $X^2$test (chi-squared test) or any of its variants. Another way that could worth to verify concept functioning is to implement it to some components where it is possible to trace their behaviour and energy consumption. Including this diagrams in the database the concept should assign precisely them or another very similar (if present) to each component.

If the implementation shows that the concept presents a well-functioning, this concept, after having captured a sufficient number of diagrams, might be used to increase the forecast's efficiency and speed through a direct prevision of the future energy consumption based only on technical and architectural data, without getting other diagrams from running production.

As it is explained in the evaluation's chapter and in the conclusion, the walk-through of the concept showed that sometimes it would be better combining more diagrams to increase the accuracy. Despite an example of how proceeding it is presented, more studies and researches should be done to express precisely the methodology and a combination of rules to formalize this process.

In the end, some reflections about the defuzzification step could be done: as it is explained in the paragraph 2.2, there is not a defuzzification method better than others, but each method has different advantaged and disadvantages depending on its application. For example a study on a new defuzzification method could help in combining more diagrams: an idea could be studying which is the threshold value of the difference between the truth degree of the rule first place and runner-up’s one. If the difference between these two truth degree returned a value lesser than the threshold value, combining the two diagrams specified by the two rules would be the best solutions, otherwise if the returned value was superior than the threshold value the assigned diagram would be the one indicated by first place rule. This study also should lead to figure out how making those processes automated.

# 7 References


[ABEL11] Abele, E.; Sielaff, T.; Schiffler, A.; Rothenbucher, S.: Analyzing energy consumption of machine tool spindle units and Identification of potential for improvements of efficiency, Proceedings of the 18th CIRP international conference on life cycle engineering, Braunschweig, Germany, 2nd – 4th May 2011, pp. 280-285

[ALTI11] Altintas, Y.; Verl, A.; Brecher, C.; Uriarte, L.; Pritschow, G.: Machine tool feed drives, CIRP annals – Manufacturing technology, 60 (2011) 2, pp. 779-796

[BIG28] Big28, December 2013, http://www.ambrosimeccanica.com/affilatura.asp

[BIOL13] Biolchi, V.: Energia termica e processi industriali, Tempco srl, August 2013

[BOSC09] Bosch Rexroth AG: Unità viti a sfere R310IT, Linear motion and assembly technologies, Rexroth Bosch group, August 09, www.boschrexroth.com/brl

[BOSC11] Bosch Rexroth AG: Instructions ball screw assembles, Linear motion and assembly technologies, Rexroth Bosch group, June 2011, www.boschrexroth.com

[BREC12] Brecher, C.; Baumler, S.; Jasper, D.; Triebs, J. : Energy efficient cooling systems for machine tools, Proceedings of the 19th CIRP conference on life cycle engineering, Berkeley, USA, 23rd – 25th May 2012, pp. 239-244

[BRUS13] Traxxas Velineon 3500 Brushless RC Motor - New Release, December 2013, www.rcslot.com/blog/traxxas-velineon-3500-brushless-rc-motor-new-release.html

[CELI 09] Celikyilmaz, A.; Türksen, I. B.: Modelling uncertainty with fuzzy logic, studies in fuzziness and soft computing, 240 (2009)

[CHIN98] Chinelatto, E.: Gestione intelligent di scorte con logica fuzzy, “master thesis”, università degli studi di Padova, 1998

[DESI04] De Simone, M.: La tecnologia fuzzy, “doctoral thesis”, università degli studi di Milano, 2004

[DORE13] Doreth, K.; Henjes, J.; Kröning, S.: Approach to forecast energy consumption of machine tools within the design phase, “article”, University of Hannover, 2013

[GOLD96] Golding, R.a.; Rosenbloom, P. S.: Improving accuracy by combining rule-based and case-based reasoning, Artificial Intelligence 87 (1996), 215-254

[HYDA07] Hydac international, Kühlsysteme D/E/F 5.700.5, Hydac cooling system, March 07, www.hydac.com/de-en/coolingsystems

[KOLO92] Kolodner, J. L.: An introduction to Case Based Reasoning, Artificial Intelligence Review, 6 (1992), pp. 3-34

[LELI09] Lelikov, O. P.: Springer handbook of mechanical engineering, California Institute of Technology, 2009, pp. 327-522

[LOPE01] Lopez de Mantaras, R.: Case-Based Reasoning, Machine Learning and Its Applications, 2049 (2001), pp. 127-145

[LOPE09] Lopez de Lacalle, L. N.; Lamikiz A. : Machine tools for high performance machining, Springer Verlag, London, 2009

[MITSa13] Mitsubishi materials, Frese integrali, Catalogo generale, Informazione sui prodotti, 2013, www.mitsubishicarbide.com

[MITSb13] Mitsubishi materials,Dati tecnici, Catalogo generale, Informazione sui prodotti, 2013, www.mitsubishicarbide.com

[RACK13 ] Rack and pinion drive - max. 850 Nm, 2 - 4 mm – VOGEL, December 2013, www.directindustry.com/prod/vogel/rack-pinion-drives-9352-932811

[SAND13] Sandvik Coromant: the reaming operation, December 2013, http://www.sandvik.coromant.com/es-es/knowledge/reaming/pages/default.aspx

[SCRE13] Screenshot-1, October 2013, http://ferasferas.wordpress.com/2010/10/16/the-cbr-cycle/

[SHIH11] Shih, W. L.; Shih, C. C.: Parameter tuning, feature selection and weight assignment of features for case-based reasoning by artificial immune system, Applied soft computing 11 (2011), pp. 5042-5052

[SIEM07] Siemens Sinamics S120 1FT7 Synchronous motor catalogue, Siemens servomotors, 04/2007, www.automation.siemens.com

[SIEM12] Siemens technical data 1PH8 main motor, Siemens Asynchronous motors, 2012, http://www.automation.siemens.com/

[STEP13] Stepper-Motor-Architecture_0, December 2013, www.engineersgarage.com/articles/stepper-motors

[TEMP14] Tempco Blog, www.tempcoblog.it, January 2014

[WIKI14] Wikipedia, Verification and validation, www.wikipedia.org, April 2014

[WYGR13] Wygralak, M.: Intelligent counting under information imprecision, studies in fuzziness and soft computing, 292 (2013), pp. 111-137