BACHELOR THESIS

Balance control via fuzzy logic

FACHHOCHSCHULE TECHNIKUM WIEN
STUDIENGANG ELEKTRONIK

Stefan Leipold
Ameisgasse 63
1140 Wien
Matr.Nr. 0410254005

FH-Prof. Dipl.-Ing. Dr. Martin Horauer

Wien, 01.02.2006 ______________________________

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Ich erteile hiermit der Fachhochschule Technikum Wien die wissenschaftlichen Nutzungsrechte an der vorliegenden Arbeit.

Wien 01.02.2006 ____________________________

TABLE OF CONTENT

Abstract

Fuzzy controll is a very powerful tool to handle difficult control circuits. It is based on the mathematic set theory and the theory of fuzzy values.This paper explains the usage of linguistic statements and terms as well as the basic considerations needed to design a control ciricuit with fuzzy logic. Further more a discription , how to design a system for balance control is included.

This system is adepted for the C167, and controls the balance of an “one-axies-vehicle”.

1 Introduction

This thesis discribes how to use fuzzy logic to design simple and effective control circuits. It contains the basic mathematic background informations to solve simple tasks. For more detailed information should refer to [B1], [B2] and [B3]. For better understanding it's very usefull to be familiar with classic (Boolean) logic and the very basics of the mathematic set theory.

2 Historical developement

The first idea of fuzzy sets and fuzzy quantities was developed in 1965 by Lotfi Zadeh in the U.S.A. First reactions on the presentation of his publication were more then negative. The belief that every problem could be solved with exact mathematic formulas, if it is defined accurate enough, molded the development of systems. With this stance there was no place for non exact expressions and non exact calculations.
So the first usage of this technique was in Europe in 1974. A regulation system based on fuzzy logic was used to control a steam machine, because all other known control systems couldn't reach the wanted results. With this success fuzzy logic became more famous especially in the area of steam engines, cement production and military usage.
In 1988 Japan found the advantages of fuzzy control and started to improve and implement this technique in their products. While in Europe fuzzy logic was mainly used to solve very complex problems, Japanese technicians found out that fuzzy logic allows to attach technical know how and experience to simpler devices as well.
The usage of fuzzy-systems in Europe started rather slowly from 1991 awards when the first Japanese product came to the "westward world". However there was a strong position against Fuzzy logic, especially form automatic control engineers, because Fuzzy logic was always promoted as the new technique that will completely replace the classic control theory.
Nowadays it's a well known fact that Fuzzy can't replace the classical automatic control but is a very powerful tool to extend the classical logic for some applications. One of the best known examples is the process of producing cement. Longtime there was no chance to build a control system that fits the high demand of this process, since the quality of the cement depends not only on physical values as time, temperature, pressure but also on "human factors" like know-how and experience.

These days fuzzy control is used in nearly every product section. The earliest implementations are:

- Picture stabilization with a fuzzy gyroscope
- Fuzzy washing machine with a one-button-control
- Anti break system for vehicles (ABS)
- Automatic gear systems
- Automatic traction control systems (ASR)
- To control the drive dynamic of cars (INVEC)

3 Why Fuzzy logic?

The main advantages of fuzzy logic and fuzzy control are the possibility to attach experience and know-how to a system and to express very complex systems in an easy way:

- It's possible to solve technical tasks where the mathematic model construction is very complex or impossible in an efficient way.
- The description of technical correlations in spoken language allows the construction of a "self explaining" system.
- Existing functions could be improved by including process know-how into the system

.

Despite all these adavantages, fuzzy logic still isn't the perfect solution for every technical task. To decide wether to use a classic control system or a control system based on fuzzy logic, the following points should be considered:

Use fuzzy logic:

- When it's not possible to find an adequate mathematic model in an acceptable time.
- When the task includes either nonlinearities, hard to manage time constants, or a large number of input parameters.
- When the technical know-how about the process is available and it's useful to include this know-how in the system design.

Use classic logic:

- When the task is easy to solve with classical control systems such as PID or two point control.
- When an adequate and easy to solve mathematic model is available.

4 Basic informations about fuzzy logic

The fuzzy logic is based on the mathematical set theory. An arbitrary number of states (=set) can result out of one statement. The biggest difference between classical logic and fuzzy logic is that fuzzy logic is developed out of linguistic descriptions. Building a control system with classical logic means to work with exact values and formulas. So it's necessary to have big mathematic know how and a mathematic model to calculate and to create the logic. In contrast fuzzy logic is the way to describe linguistic statements with mathematic expressions. Further more Fuzzy logic allows to handle abstract values e.g. "date", "temperature", "pressure" with adjectives like "begin of the week", "hot", "cold", "high", "low".

Every adjective is weighted by a number that declares the rate in which the adjective machtes the current state. Where "0" means the abstract value doesn't fit to the current state and "1" means the current state fits perfectly. This value could be compared with the "0" and "1" used in classic logic systems, with the difference that fuzzy logic also allows numbers in between "0" and "1".

4.1 The procedure of using fuzzy controls

The whole cycle of controlling systems with fuzzy logic can be split into five stages as shown in figure Figure 1.

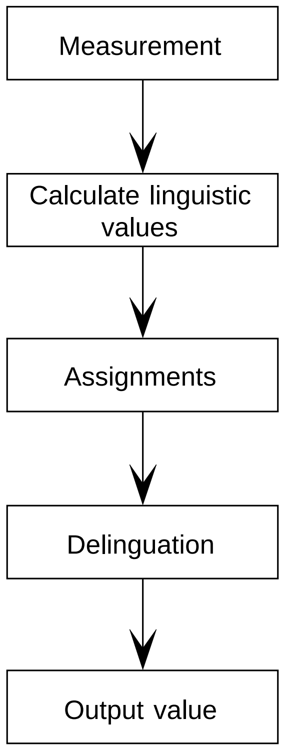

Figure 1: the control circle

Measurement:

The input values in technical systems are normally created by some kind of sensors. Sensors produce exact, measurable values as they are used in classical logic systems. In this step also some standardization operations are made, e.g. filtering, to make it easier and more comfortable to use the measured values for further calculations.

Calculate linguistic values:
In this step the physical input values, the base values of the fuzzy logic, are transformed into linguistic terms. This is done by the affiliation functions or affiliation lists (section 4.5).

Assignments:
Now the linguistic values are combined to calculate the output value(s). This is done with assignments, similar to the classic logic system, like AND, OR or negation operations. In this way the behavior of the control system is designed. The output values of this step are sets with the possible linguistic terms and their plausibility.
For a detailed description of this assignments refer to [B1].

Delinguation
The last step is necessary to build physical values out of the probability of the linguistic terms. Depending on the application and their properties, there are many different ways to calculate this physical value (section 4.6).

Output values
The calculated values can now be used to control engines, valves, etc.

4.2 Linguistic statements

Linguistic is the theory of spoken language. In contrast to mathematic statements, linguistic statements aren't exact at all. Verbalisms like "a little bit more", "nearly enough", "almost right" are often used to describe a situation. There is a lot of space to interpret the exact meaning of one statement. Fuzzy logic uses this margin to integrate technical know-how and experience of the system designer into the system itself. In other words the system starts to "think like humans". The following example shows some possible linguistic statements:

Statement 1: if the water is too cold.
Statement 2: if the distance is too far.
Statement 3: if there is too less power.

Nearly every human action is based on such statements, since it is difficult to describe physical procedures in mathematic language. The next example shows the way humans handle physical problems and tasks with linguistic descriptions and fuzzy values:

When humans try to throw a ball into a basket, they don't know anything about the physics behind the ball. They also don't need to calculate the trajectory as it would be done by a classical control system. After the first shot the result is analyzed with inexact phrases like "too strong", "too soft", "too far to the right". These "input values" are combined and as a result non exact decisions like "a little bit stronger", "more to the left" are made.
This example shows a very simple and effective way to solve problems with many input values. To get an adequate control system based on classical logic, it's necessary to consider distances, angles, as well as weight and rotation of the ball. Out of these values it's possible to calculate a trajectory with two angles, the power needed to throw the ball and the rotation the ball should have. Especially in microcontrollers such calculations are too complex and time-consuming.

4.3 The basic set

Every abstract value in a linguistic system has a set of concerted states. This set is called the "basic set" and could be countable, continuous or more dimensional. The elements of this set are the basic values. The specification of the basic values depends on the usage of the whole set and has to mind the rules of:

- Uniqueness
- Measurability

Uniqueness means that every abstract value has only one basic value at any time, and it's possible to measure basic value (at any time). Some Basic values with their base values are shown in the example below.

Basic set 1:
Date
Basic Values:
Mon, Tue, Wed, Thu, Fri, Sat and Sun
Mathematic description:
B1= {Mon, Tue, Wed, Thu, Fri, Sat; Sun}

Basic set 2:
Acceleration:
Basic Values:
1, 2, 3, 4, 5
Mathematic description
B1= {1, 2, 3, 4, 5}

Basic set 3:
Temperature:
Basic values:
from -10 to +10
Mathematic description
$B2 = \{x \in N | -10 < x < 10\}$

4.4 Linguistic terms

The basis of every linguistic system is the description of abstract values in colloquial linguistic terms. In contrast to the base value, every abstract value can have one or more linguistic values. So it's impossible to figure out the abstract value from its describing linguistic terms. The linguistic values are fuzzy. This means that the direct assignment of linguistic and abstract values is not possible, because it depends on many not assessable, external factors as well as on the imprecision of the spoken language. For an example: A high temperature is in terms of weather 30°C and in terms of soldering 400°C.

The allowed linguistic terms must be defined for every system in the same way as the basis values, however the linguistic terms don't have to be unique or measurable. Instead they should fit with the normal parlance. The next example shows possible linguistic terms for the basic sets out of the example in section 4.3.

Date:

L1= {week start, week middle, week end}

L2= {spring, summer, autumn, winter}

L2= Christmas, Easter, Pentecost, …}

Temperature:

L1= {high, normal, low}

L2= {nearly nothing, nearly 50°C, nearly 100°C, much more then 100°C}

4.5 The affiliation function

In every situation an abstract value has one exact base value and one or more fuzzy linguistic terms. The calculation from abstract into linguistic values is done by the affiliation function $\mu_{(b, l).}$ Where "b" stands for the base value and "l" for the linguistic term. The result of this function is a number between 0 and 1 and describes the plausibility that the linguistic term "l" describes the base value "b" probably. Where 0 means they don't fit and 1 means that they fit exactly. The next example describes one way of designing an affiliation function out of the examples in section 4.3 and 4.4.

Date:

Base set: B= {Mon, Tue, Wed, Thu, Fri, Sat; Sun}

Linguistic terms: L ={week start, week middle, week end}

Linguistic term	Week start	Week middle	Week end
Basic value			
Mon	1.0	0.0.	0.0
Tue	0.3	0.6	0.0
Wed	0.0	1.0	0.0
Thu	0.0	0.3	0.5
Fri	0.0	0.2	0.6
Sat	0.0	0.0	0.9
Sun	0.0	0.0	1.0

Table 1: Affiliation list exapmple

The function design depends strongly on the size of the base set. For very small base sets truth tables are the simplest solution, especially for creating source code, because truth tables can be easily programmed using “switch()” statements. In the case of big or continuous base sets every linguistic term could be seen as separate function.

Basic set
$B2 = \{x \in N | 0 < x < 100\}$

Linguistic terms:
L= {high, normal, low}

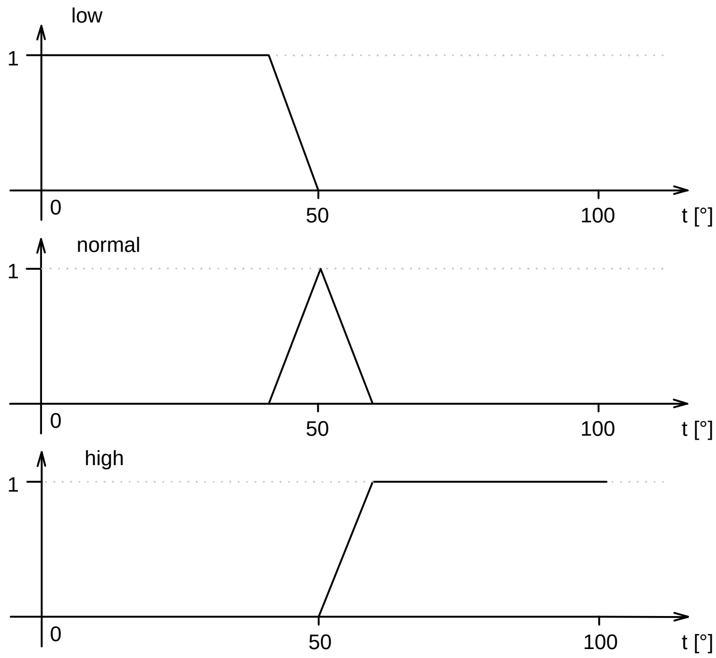

Figure 2: Affiliation function example a

In an often found illustration, all functions are painted into one chart. From now on this design is also used in this paper.

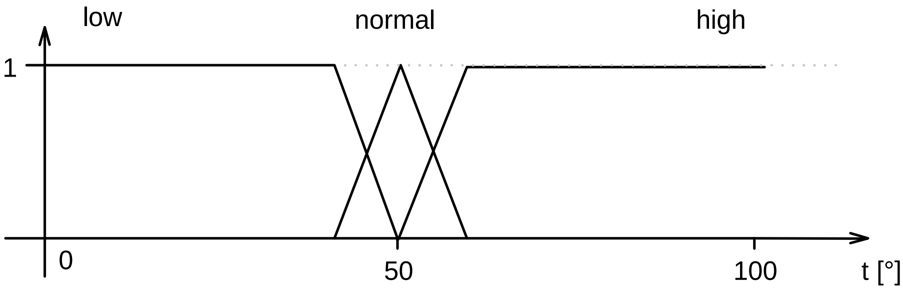

Figure 3: Affiliation function example b

4.6 Delinguation

The last step is necessary to build physical values out of the probability of the linguistic terms. Depending on the application and their properties, there are many different ways to calculate the physical value. In this section only the barycenter of the area is described in detail. For more information about other delinguation methods refer to [B1] and [B3].

The first step of every delinguation methode is to bind the maxima of the output affiliation function to the result of the output dataset (plausibility of the linguistic terms) as it is shown below:

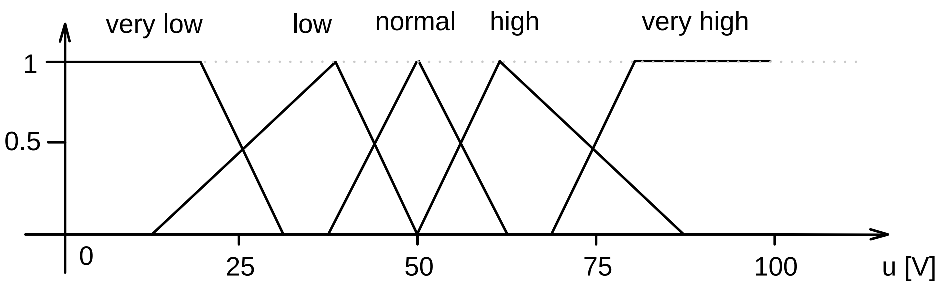

Figure 4: Delinguation example

The maxima of the original affiliation function are bound to the values of the result.

$\mu_{(\log ic)} = (veryLow, 0.0), (low, 0.2), (normal, 0.4), (high, 0.7), (veryHigh, 0.0)$

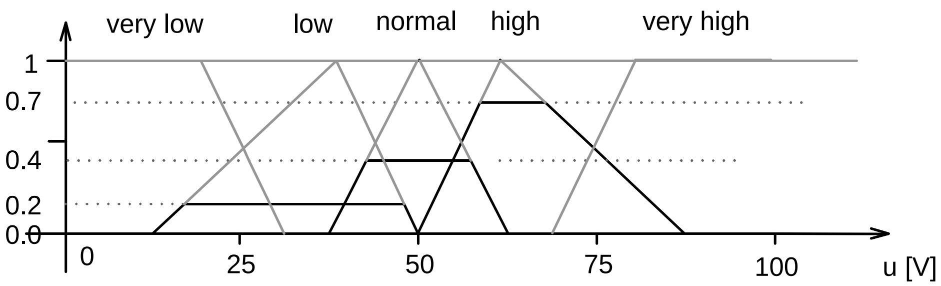

Figure 5: bound output affiliation function

Out of this new graph, the physical output value can be calculated. The way described in here is the barycenter of the area. This method calculates the barycenter of the area between the function and the abscissa. The abscissa value of the barycenter is the physical output value. Per definition the barycenter and therefore the current output value a_{cur}, is the quotient of the torque M and the area F.

$$a_{cur} = \frac{M}{F}$$

For M and F the following expressions are used.

$$M = \int_{a_{\min}}^{a_{\max}} a * f_{(a)}, da \qquad F = \int_{a_{\min}}^{a_{\max}} f_{(a)}, da$$

This is one of the most exact methods to calculate the physical value. To use it effectivly the affiliation function of the output value must be integrable or at least terminateable. If this is not possible or the base set consists of singletons this approach should be exact enough.

$$a_{cur} = \frac{\sum a_i * f_{(a_i)}}{\sum f_{(a_i)}}$$

The big disadvantage behind this solution is the enormous performance required to calculate the area. If it's possible geometric methods could be used to minimize this complexity.

5 Task: Balance

Balance is a one axis vehicle with two wheels. It shall be able to hold itself in balance by moving forward or backward with a specific velocity and a specific acceleration. But also "normal" movement, this means forward, backward, left and right, should be possible. The control commands are sent via Bluetooth to the Balance, received with a Bluetooth dongle and handled by the onboard USART of the C167 microcontroller. The C167 also does the whole balance control as well as the handling of the received data. The activation of the engines (PID-controller) is implemented on an external FPGA. This project is a four people work. The following paper only describes the balance control. For further information about the other parts see [BT1] and [BT2].

6 Basic considerations:

6.1 Why fuzzy control?

To realize this problem two ways are possible: classical logic or fuzzy control. To design such a control system with classic logic means to know the exact behavior of the robot, starting with weight, position of the center of mass as well as a complete description about the engines. All this constants and all input values are combined into one expression to calculate the output speed for the engines. As described in section 2 this system works with exact input and output values and includes complex calculation operations. On the other hand a controller design with fuzzy control would be an alternative. Such a system reacts more "instinctive". In contrast to classic logic, it's hardly necessary to know constants like weight, speed and so on. Also the output values for the engines are calculated with a simple logic assignment of input values in analogy to that used by humans instead of one big and complicated expression. E.g. balancing a wooden bar on the palm. This task could be solved by every human after a couple of tries. To solve this task it is not necessary to know the physical background or the physical behavior of the bar. The reaction comes as an instinctive answer relted to the movement of the bar. Is the bar tipping to the right, the hand is moved to the right, is the bar tipping to the left, the hand must move to the left; and so on. Fuzzy logic works in the same way and this allows disregarding physical parameters in the control circuit and implementing human experience instead.

6.2 Input values

The next point is thinking about which input values are needed for controlling Balance and in which way Balance should react on certain inputs. Remembering the example above, one input value is the angle α of the longitudinal axle of the robot to the horizontal (Figure 6:

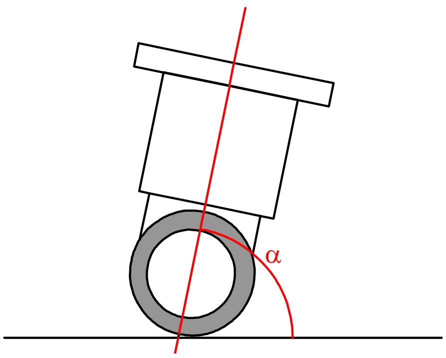

Figure 6: Balance input values

Balance input values). Is the angle zero, balance doesn't have to react. Is the Angle negative balance has to move backward; otherwise it has to move forward. In particular, not only is the sign of the angle relevant also the amount has to be considered. If a big error angle is measured the reaction has to be fast, if the error angle is low a small reaction is in order. Otherwise, the change of the angle would be too great so that the robot cants to the other side. This leads to the next point: the ideal state is reached with an error angle and an angle speed of zero. Therefore also the speed of the change of the angle (further called "angle speed") must be involved into the control circuit. Similar to the amount of the error angle, the angle speed determines the acceleration of the engines.

6.3 Measurement

As mentioned in section 6.2, the physical values "error angle, and the "angle speed", have to be measured. Both tasks can be implemented by using a gyroscope. This component simply returns a voltage proportional to the speed of the angle movement. Out of this voltage it's no problem to calculate an absolute angle, as long as the time between the measurements is constant. However, this solution has two disadvantages.

- there is no specified point zero. Because the gyroscope can't measure absolute angles, the angle zero is equal to the angle at power up,
- the accuracy of the calculated absolute angle. Due to measurement errors, inexact calibration and too fast movements, the absolute angle drifts and the balance control gets difficult.

A different way to solve this problem is to use a pair of distance sensors,to measure the distance between a point at a specific height at the robot, down to the ground (Figure 7: distance sensors).

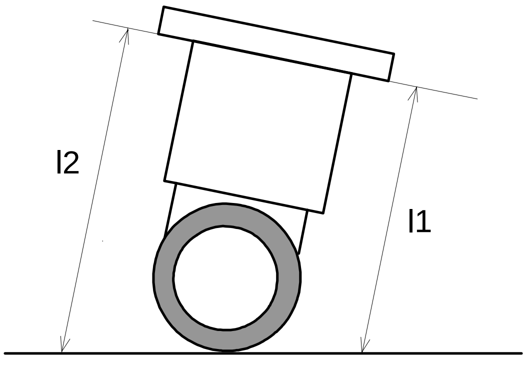

Figure 7: distance sensors

This leads to two different ways to get the exact absolute angle: Either the absolute angle is calculated out of the distances towards the ground or the absolute angle is calculated out of the results from the gyroscope and is set to zero every time the measured distances of both sensors are equal.

7 System design

7.1 Base values and linguistic terms

Based on the considerations of section 3 the system includes three input values "error angle" and "angle speed", as well as one output value, the speed of the engines. The first step in system design is to fix the base value and the linguistic terms which are allowed in the system. All physical values in the system are continuous and could be either negative or positive. Furthermore, six different linguistic terms are allowed:

- NH negative high
- NM negative medium
- NL negative low
- ZR zero
- PL positive low
- PM positive medium
- PH positive high

Because the base values are continuous the use of affiliation lists is senseless, so affiliation functions are used. For every physical value, seven carts are used to translate the physical value into a linguistic term, expect "angle speed" and "error position" which are described by five carts.

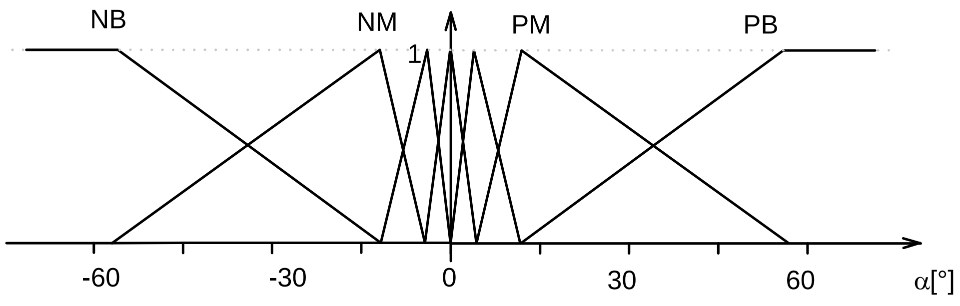

Figure 8: Affiliation function for the error angle

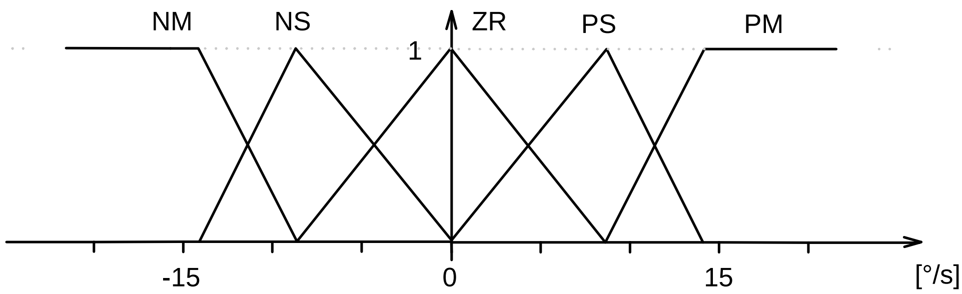

Figure 9: Affiliation function for the angle speed

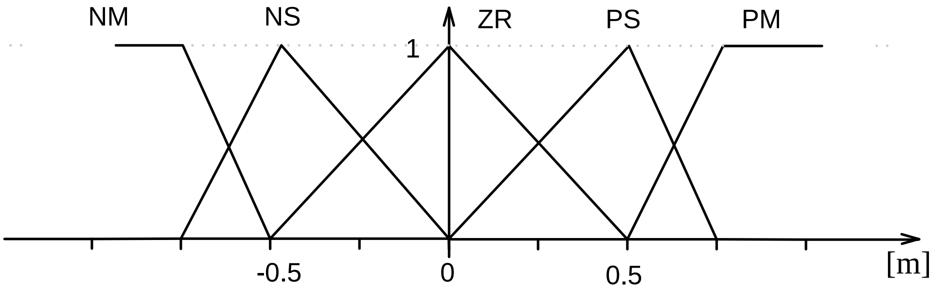

Figure 10:Affiliation function for the error possition

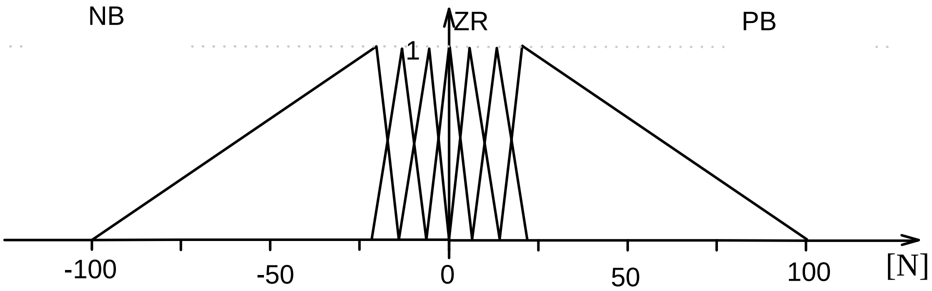

Figure 11: Affiliation function for the output value

7.2 Assignment of the input values

Humans balance a bar by instinct. The movement of the hand simply dependends on the assignment of the “error angle” and how fast this angle changes (= “angle speed”). Based on simple logic, the following rules can be developed:

- If the error angle is very big, the bar will definitely tip, so the reaction must be very big.
- If the error angle is medium and the angle speed is also medium, the error angle soon will be very big. To beware of this the reaction must be very big too.
- If the error angle is medium and the angle speed is low or the error angle is low and the angle speed is medium, the bar tips slowly in that moment but reaches more and more speed. Therefore there must be a medium reaction against this movement.
- If the error angle is zero but the error speed is medium, the bar stands good at the moument, but will start to tip because of the angle speed. So a medium correction is needed.
- If the error angle and angle speed is small also the reaction has to be very small, because the bar stands nearly perfect.
- If the error angle is medium and there is also a medium angle speed in the opposite direction, it’s assumed that the bar stands still or will soon stand still and there no correction is needed at all.

This logic rules count for humans as well as for this fuzzy control. With the inclusion of the error position, the following assignment table for the system can be designed.

if	E = NM E = NB	and	D = NM	THEN	K = NB
	E = NM E = NS E = ZR	and and and	D = NS D = NM D = NM		K = NM
	E = NS	and	D = NS		K = NS
	E = PM E = ZR E = NM	and and and	D = NM D = ZR D = PM		K = ZR
	E = PM E = PB	and	D = PM		K = PB
	E = PM E = PS E = ZR	and and and	D = PS D = PM D = PM		K = PM
	E = PS	and	D = PS		K = PS

Table 2: Assignments used for balance control

7.3 Calculation of the output value

For the first tests the calculation of the output values is done via the barycentre of the area, because the affiliation functions of the output value (engine power) is finite, therefore integralable. Also the C167 chip has enough resources to calculate also difficult periodic operations.

8 Software design

8.1 Gyroscop

The gyroscope is used to measure the angle speed. The output of this device is an analog voltage, which is proportional to the rotation speed. The maximum values of this voltage is bound between 0V and 5V. An output voltage of 2.5V means there is no rotation. A voltage higher then 2.5V describes a positive rotation, where 5V is the maximum positive rotation speed. A voltage less then 2.5V describes a negative rotation, where 0V is the maximum rotation speed.
The resolution of the measurement is bound to 5mV/°/sec.
The output voltage of the gyroscope is measured by the C167 internal ADC with a duration of 36.1µs (27.7kHz) between each measurement. The ADC works with a

resolution of 10 bit. With a minimum voltage of 0V and a Maximum of 5V the input is measured in 5mV/bit or in other words 1°/sec/bit. Multiplied with the reciprocal of the time between the measurements an absolute angle can be calculated.

$$\alpha = \sum(\alpha_{cur} - offset) * 0.0005$$

To minimize the consequences of wrong measurements, the median value of 10 measurements is taken to calculate α_{cur}.

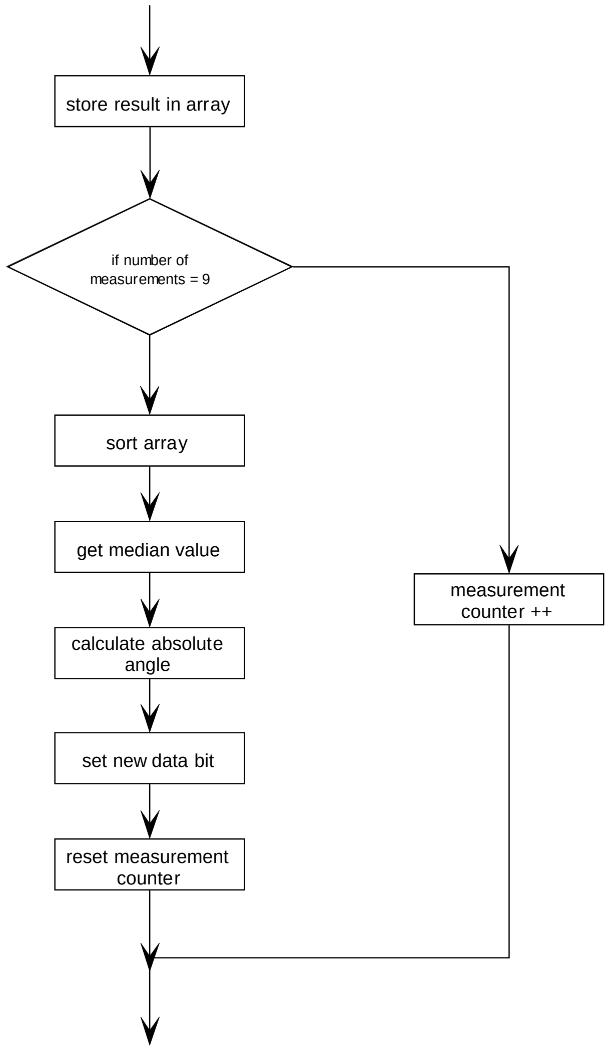

Figure 12: flow chart of gyro-program

8.2 Affiliation function

The files "affiliaitionfunction.c" and "affiliationfunction.h" include everything to calculate linguistic terms out of physical values. Also the behaviour of the system can be changed in these files, because in here also the constant values, that are used to construct the affiliation functions, are stored. Since the robot should react in both, negative and positive sides in the same way, the affiliation function just has to be designed for one side. The fuzzy sets are defined as an array out of seven float values in the order:

0. ZR
1. NB
2. NM
3. NS
4. PB
5. PM
6. PS

In this case all affiliation functions are symmetric around the ordinate. So only the half of the values have to be calculated, because the others are zero anyway.

8.3 Fuzzylink

The files “fuzzylink.h” and “fuzzylink.c” include several functions to assign input and output values, as well as a function to calculate the output value with the method of the bary center of the area. The function assignments are based on the S-norm and T-norm. Basicaly this means an OR gate corresponds to the calculation of the maximum of two values and an AND gate to the calculation of the minimum of two values. ([B1])

8.4 Assignment

The assignment of the input values and the calculation of the output value is done in the file main.c. These lines are executed with a duration of 36.1µs (27.7kHz). The expressions and descriptions of these assignments can be found in section 7.2 and table 2.

9 Appendix

9.1 Literature:

[BT1] Bachelor thesis Sams Harald, “Messen, Steuern und Regeln mit FPGA”; FH Technikum-Wien; 2006

[BT2] Bachelor thesis Böck Stefan „Remote control of a robot via bluetooth“, FH Technikum-Wien, 2006

[B1] Constantin von Altrock „Fuzzy Logic“, Oldenbourg, 1993,
ISBN 3-486-22673-8

[B2] Börcsök „Fuzzy Control – Theorie und Industrieeinsatz“, Verlag Technik, 2000
ISBN -3-341-01237-0

[B3]Alfred Friedrich „Logik und Fuzzy-Logik“, Expert Verlag, 1997,
ISBN 3-8169-1405-5