UNIVERSITY OF MISKOLC

FACULTY OF MECHANICAL ENGINEERING AND INFORMATICS

Fuzzy rule-interpolation inference and knowledge representation forms in embedded systems

Hatvany József Doctoral School of Information Sciences

Head of the Doctoral School:

Prof. Dr. Jenő Szigeti DSc

Thesis book

Supervisor: Dr. habil. József Vásárhelyi

Name: Roland Bartók

Neptun: AQCCRH

2023

Contents

1. Introduction

Adapting to a changing environment is one of the main problems of autonomous robots. Industrial collaborative robots and cobots (collaborative robot, cobot) [1] are working in a permanent environment, performing a specific task. Companion robots, on the other hand, operate in a changing environment, where unfamiliar situations may arise for them. Adaptation to various situations is facilitated by behavior-based control lean on ethological models. [2] and [3]

Ethological models were created based on the observation of the behavior of living beings. For example, the model is based on a dog's behavior [8]. The behaviors created based on the models are all valid in certain situations, but to different degrees. The description of behaviors is supported by FBDL (Fuzzy Behavior Description Language), which can be used to specify the given behavior pattern with rules [28] and [32]. The behavior described in FBDL can be calculated using the FIVE (Fuzzy Interpolation in Vague Environment) method. Taking advantage of the fuzzy nature, the fuzzy state machine creates the momentary movement and action of the robot by calculating the existing behaviors and combining them according to the environment.

The purpose of this dissertation is to examine the FIVE method applied to the description of different behaviors, in the control of mobile robots on embedded platforms. The purpose of the research is mainly to collect data to improve computing efficiency in robot control. In order to do this, I created the μFRI C language library, which is an implementation of the FIVE method intended for embedded systems.

I examined the parallelizability of the μFRI library on currently available and popular multi-processor "system on chip" solutions (SoC - System on Chip), from the point of view of optimal system utilization of descriptions containing many rule bases. I examined how the rule bases should be distributed in a given multi-processor environment in such a way that it would result in the greatest possible increase in calculation speed. Furthermore, the thesis shows a method for the hardware implementation of the FIVE method on FPGA (Field Programmable Gate Array) for applications requiring fast execution. The research includes the implementation of the calculations belonging to the FIVE method and a sample database used to store the knowledge base. Finally, I propose a solution in connection with the fact that the FIVE method is not only suitable for describing behaviors, but can also be used as a classification method to detect obstacles surrounding the robot by analyzing sensor data. The results are compared with the Naive-Bayes classifier using known conditional probability and a large amount of training samples.

2. Research background, the FIVE method

The autonomous robot makes its decisions and shapes its behavior based on information from multiple sensors and its own internal state. These input data form the observations and antecedents of the fuzzy rules. The outputs of the fuzzy rules are the consequent ones. A fuzzy rule can be written in the form of an implication in the form "IF … THEN …." The set of fuzzy rules affecting the same output is the fuzzy rule base. [23], [28], [32] and [36]

In the case of fuzzy, in order to ensure the stability of the system, the rules must be designed in such a way that there is a rule for every possible input combination, and the rules completely cover the state space. As the size of the task to be solved increases, this can even cause an exponentially increasing number of rules, which also increases the computational demand. The large number of rules also means an additional problem, namely partial knowledge of the task to be solved. In other words, the expert creating the rules may not be able to cover all possibilities due to the mathematical complexity of the problem or lack of information. For this reason, instead of a completely covering rule system, a rare rule base is formed. In the case of a rare rule base, there is an observation that does not belong to any rule, in which case the system output is uncertain. [23], [28], [32] and [36]

A sparse set of rules is more common than a full, covering set of rules. Fuzzy interpolation methods are a solution to the problem of sparse rule systems. The essence of interpolation methods is that the decision for the missing rules is calculated based on the existing rules. Thanks to this, it is enough for the expert to specify the most important rules for the operation of the system. This can reduce the number of rules, the complexity of the rule system and, depending on the interpolation method, the computation requirements. [23], [28], [32] and [36]

The FIVE method places the rules in a fuzzy space where the Euclidean distance between the rules can be interpreted. The rules can be distinguished from each other based on distance: a larger rule distance means a greater difference between two rules, while rules located at a smaller distance are more similar to each other. Using the weighted consequents by the rule distance, Shepard interpolation uses the inverse distances to create the decision of the rule base. The steps of FIVE method are shown on Fig 1. *Observation* is the information from the environment or internal state. This value to the observation is assigned by the *Linear Interpolation*. This value indicates in what extent of the observation is a member of the given fuzzy set. Its symbol is μ. The difference between the antecedent belonging to the rule and the μ value of the *Observation* gives the *Antecedents Distance*. The *Rule Distance* is the Euclidean distance of the Antecedents distances. The *Shepard Interpolation* uses the Rule distance to produce the *Decision*. [23], [28], [32] and [36]

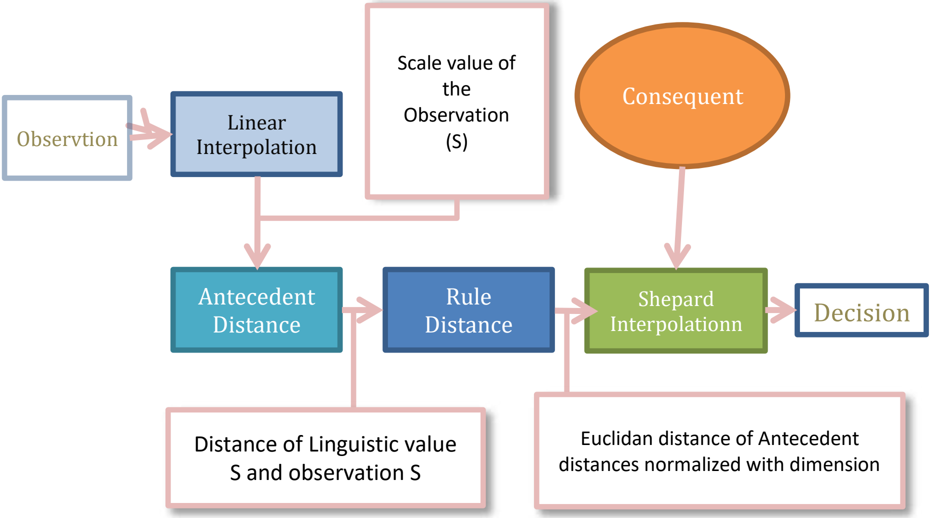


Fig 1. Computation steps of FIVE method

3. µFRI API

3.1. Objective

In addition to the already existing implementations of the FIVE method, I created a minimal implementation in C language, which takes into account the limited resources of embedded systems, performs the calculation steps of FIVE and stores parameters in a space-saving way. Until now, the implementation of the FIVE method exists in the following languages: C++, Python, JavaScript and Matlab. The Python and JavaScript versions also include the language interpreter, the other versions upload their knowledge base from a file. *The µFRI library, on the other hand, uses a structure that can be dynamically modified during runtime to store the parameters. This allows the user to perform the calculation at runtime with a completely new rule base.* Furthermore, the implementation in C makes it possible to use FIVE in Java, C#, and Python by loading it as an external library.

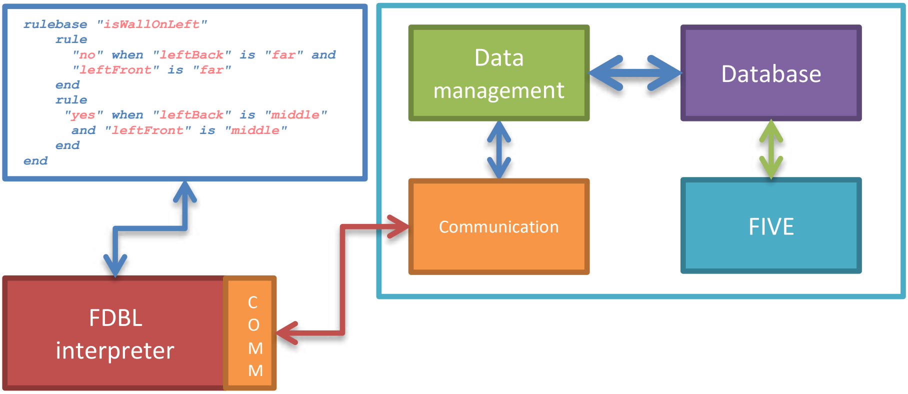
3.2. The parts of API

The µFRI (micro FRI, Fuzzy Rule Interpolation, further designation: uFRI) library is a lightweight, low-memory implementation of the FIVE method. The data structures of µFRI have been harmonized with the elements of the Fuzzy Behavior Description Language (FBDL). The implementation supports dynamic and static database uploading and rule base tuning by changing parameters or modifying the rule structure.

The µFRI parts are presented as follows (see Fig. 2.):

- *Database*: used to store the parameters of the universes and rule bases.

- *Data management*: uploading the database, adding and deleting universes and rule bases, and managing its parameters. In the case of systems that do not support dynamic memory allocation, this section can be turned off.
- FIVE: Computation parts based on FIVE.
- *Communication*: a binary and textual command interpreter that helps to modify the database, which is an external module. With its help, the knowledge base can be uploaded to the Database from FBDL or can be a tuning method via any communication channel. Furthermore, the knowledge base can be exported from the system. COMM and Communication indicate the same module.
- *FBDL interpreter*: interprets the descriptive language and generates the appropriate data for the communication block. The FBDL interpreter is not part of the library.



```
rulebase "isWallOnLeft"
    rule
      "no" when "leftBack" is "far" and
      "leftFront" is "far"
    end
    rule
     "yes" when "leftBack" is "middle"
      and "leftFront" is "middle"
    end
end
```

Fig. 2. Parts of μFRI API

3.3. New results

The μFRI library enables the use of FIVE-based behavior description in embedded systems so that the rule bases can be changed dynamically at runtime. With the μFRI library, it is possible to create systems that implement complex behavior with low energy requirements using FIVE.

The data structure of the μFRI library is based on the FBDL language. With the help of a compiler, the behavior of the system running μFRI can be changed or tuned in real time with the message pattern presented in [S8].

3.4. Practical application

The following devices were controlled using the μFRI library:

- Sphere bot with pendulum based movement control: [33],
- Inverted pendulum robot stabilization: [34],
- Wall detection for Micromouse robot: [S8] és [S10],
- Play with ball behavior for holonomic drive robot (KR1): [S5] és [S7].

3.5. Summary

In this chapter, I presented the version of FIVE with minimal memory requirements created in the C language, which I used to perform the tests carried out during the research.

The detailed operation and use are described in articles [S8] and [S10]. In relation to this topic, thesis I. was established.

4. Speeding up the parallel calculation speed of the FIVE method

The μFRI library is an implementation of the FIVE method with low computational and memory requirements. Based on the results of practical applications and Q-learning so far, the number of rule bases for each problem can be increased from 2-3 rule bases to 1000-2000 pieces. [31], [35] and [36].

4.1. Objective

The purpose of the investigation was to determine how the computational efficiency can be improved in the case of currently used, popular multi-core embedded systems, or to determine an optimal value for how many processor cores should be involved in the computation. For this purpose, the efficiency of computation in relation to the size of the system's cache memory was examined based on the library's memory requirements. Also, the degree of acceleration achieved compared to single-processor execution was determined from the calculation times based on Amdahl's relation.

Since the computation steps of μFRI themselves take a relatively short time, the algorithm is not further divided. A set of rules is treated as a unit during tests.

The result of increasing the calculation efficiency is that the time required for simulations and teaching procedures can be reduced, not only in the case of a small but also a large amount of rules.

4.2. Test methods

Table 1. Summary of test methods

Method name	Purpose	Method
Cache examination	What is the acceleration effect of the cache? At what amount of data does it reach 0 acceleration?	The ratio of the time of calculations performed with an empty cache and with cached data.
Compare of computation time of runs	Where is there a significant change in calculation time or their ratio?	Examination of total calculation times and their ratio in cached and non-cached cases.
Speed up examination	Speed up examination	Using Amdahl's formula. Speed up calculated compared to 1 processor. Based on the ratio of total calculation times, acceleration calculated compared to 1 processor in the case of 2 and 3 processors. Examination of the time spent by the processors on useful calculations.

4.3. New results

In this chapter, I examined the acceleration effect of the cache and the behavior of multi-core processor systems (ARM A53/A72). Compared to the amount of data envolved in the computation, using μFRI library. I analysed different storage requirements of the μFRI library.

During the tests, it was revealed that when running the μFRI library on several processor cores, due to its short calculation time, a decrease in speed can be observed, and there is no significant acceleration in the case of a tightly coupled multiprocessor system compared to the 1 working processor.

4.4. Practical application

The μFRI C language implementation of the FIVE method is designed for use on embedded systems. With the help of the results, the multiprocessor implementation can be saved in the case of a low number of rules, except when the fast response time of the system is required and is applied on a loosely coupled multiprocessor system.

4.5. Summary

I performed an examination of the calculation time of the μFRI library with the aim of determining what size of the rule base it is worth to envolve several processors in calculation, taking into account the cache size. The research was made for different data volumes (100, 1000 or 5000 rulebases). Based on the results, it was found that in the case of tightly coupled multiprocessor systems, the efficiency does not improve significantly if more processors perform the calculations. Also, the scheduler interrupts can cause significant (based on previous tests), increase of up to 30% in the calculation time, if the calculations take longer than the time slot allocated by the scheduler. In relation to this topic, thesis II. was established.

5. Hardware implementation of the FIVE method

The FIVE interpolation method is lightweight and can be easily used in an embedded system environment. In addition to reducing the time required for simulation and self-learning procedures, its further acceleration is justified due to the regulation of processes requiring rapid intervention. A possible solution to this is the hardware implementation of the FIVE method, which enables a simplified form of the calculation to be performed extremely quickly. The data structures designed for enable the adaptive (dynamic) configuration and tuning of hardware-implemented rule bases.

5.1. Objective

The hardware implementation of the computation steps of the FIVE method implemented in a hardware description language so that the parameters of the rule bases and universes can be changed during run-time without system shutdown.

Requirements for the circuit:

- can be tuned during operation, the parameters of the rule base can be changed,
- have simple interface,
- the data bus width is parametrizable ,
- portability, allows implementation on several types of FPGAs,
- the Intellectual Property (IP) allows module generation from FBDL.

5.2. Data stream of calculation

The FIVE method can produce the result with 4 calculation steps. The calculation sequence consists of 2 interpolation calculations and 2 distance calculations.

The representation of data stream processing according to Fig. 3. contains the calculation steps of a rule base. With the exception of the *Shepard interpolation* module in the figure, the number of individual modules depends on the number of rules, antecedents and observations contained in the rule base. The number of *Linear interpolation* calculations are determined by the number of observations and the number of *Antecedents distance* calculations, which varies based on the number of antecedents in each rule. Each rule contains one *Rule distance* calculation module. A rule base contains a single *Shepard interpolation* calculation.

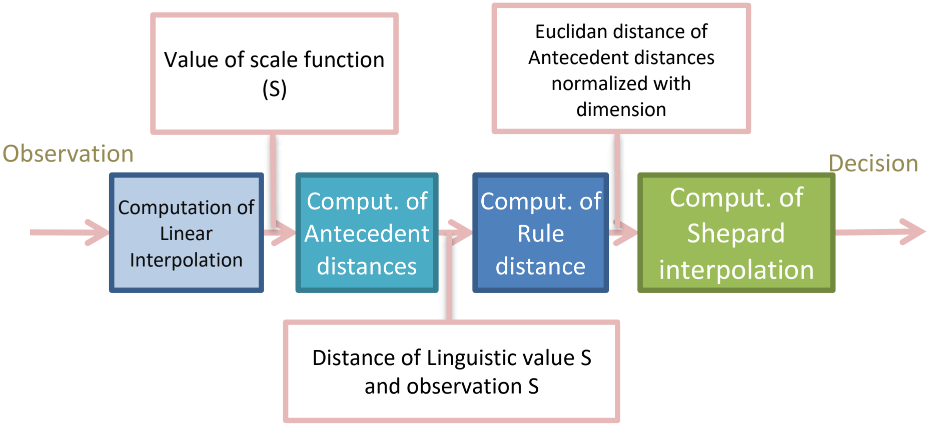


Fig. 3. Parts of FIVE IP module

Table 2. The connections of the FIVE FPGA modules

FBDL element	FIVE FPGA module	Transmitted parameters	Interface signals
universe element with linguistic values	Universe module	Number of linguistic values and parameters	The input of the block, towards the sensors. Output to the referenced AntecedentDistance
„antecedent0" is „p0" form of predicate	AntecedentDistance module	The value of the scale function of the linguistic value indicated in the predicate (second parameter)	Input to the associated Universe. Output to RuleDistance
The predicates that make up the rule	RuleDistance module	None	The inputs to the associated Antecedent Distance. Exit to Consequent.
Rulebase output universe and first parameter of the rule	Consequent module	Both parameters of the language elements of the universe specified in the rulebase	The inputs to the RuleDistance modules. The Output is to the system output

5.3. New results

I designed a framework in Verilog hardware description language for the use of the FIVE method on FPGA, which is suitable for generating a hardware IP based on FBDL (Fuzzy Behavior Description Language), which speeds up the computation of the FIVE method. The FIVE IP can be used independently or in association with a CPU as an auxiliary circuit for acceleration FPGA based embedded systems. A fuzzy behavior automaton that implements complex behavior can be created by connecting FRI_FIVE modules.

The IP structure of FIVE also enables implementation on the Xilinx Adaptive Platform as a hardware acceleration unit of a software environment. In this case, each computing unit or the entire FIVE IP is connected to the processor system via AXI or PCIe bus system and the software module performs the data collection and control tasks.

Currently, up to my knowledge there is no fuzzy interpolation method or adaptive fuzzy interpolation method implemented on FPGA. There are FPGA implementations of classical fuzzy and neuro-fuzzy methods.

5.4. Practical application

Using FIVE IP, behavior-based control can be implemented at the circuit level using FPGA. Thanks to the simple interface, it can be connected to an embedded microcontroller or to an application processor via AXI bus. Direct use is also possible by scaling the signals appropriately. FIVE IP enables high-speed control tasks and reduces tuning time by performing iterations faster compared to software implementations, such as behavioral based robot control. However, the number of rules in FIVE IP cannot be increased arbitrarily only within the constraints/limits specified by the hardware. The parameters (values of universes, values of observations and consequents) can be rewritten without limits. The update of the parameters is done by one clock period time. At the next clock signal the result is computed with the new parameters, and appears at the output.

5.5. Summary

I did the implementation of FIVE IP in Verilog. FIVE IP structure enables the tuning of the rule base in real time. In relation to this topic, thesis III. was established. For more information see: [S1], [S2], [S8] and [S13].

6. Application of the FIVE method as a classification procedure

The classification and labeling of the data from the observations is important for their further processing. The extracted information can be used to generate a decision tree in the case of mobile robot control or artificial intelligence applications. Classification procedures are supervised sorting processes that use a statistical or expert database to label observations. Conventional classification procedures usually sort the input data into crisp sets. However, the fuzzy classifier can also be used to specify the degree to which each data belongs to each class. In this way, the output of a classifier operating on the basis of an expert database can be used by several fuzzy-based behavior descriptions. In practice, this will create a smoother transition when switching between behaviors.

The fuzzy classification method can be used as a pattern recognition procedure, which also provides an indication of which patterns the incoming data series matches in addition to the best-fitting pattern.

6.1. Objective

Application of the FIVE procedure as a rule-based classifier using the μFRI library. I compare the achieved results with the Naive Bayes classifier.

In the study of the classifier, for a robot traveling in a maze, it calculates where the walls are located at the current location of the robot.

6.2. Results

The two classification procedures created based on the description correctly recognized the shape of the walls surrounding the robot at different points of the labyrinth used for testing. During the tests, I checked that the robot did measurements mainly in places where data was not collected for the trainig samples. The results of the two classifiers were consistent and correct. There was only one difference in the decisions, where the robot met with the position shown in Fig. 4., where the FIVE-based classifier did false perception of a maze.

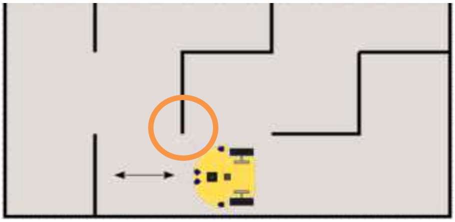


Fig. 4. False corridor detection in the FIVE-based classifier

6.3. New results

A fuzzy classifier can be created using the FIVE fuzzy interpolation method. The advantage of a classifier based on fuzzy logic is that the number of dimensions of the classifier can be reduced, as was also seen in the described example. Fuzzy classifiers created using the interpolation method can be found in the following articles: [70], [71], [72] and [73], which are based on the KH (Kóczy-Hirota) method.

The advantage of the classifier created using the FIVE method compared to the Bayesian classifier can be seen in the size of the training sample. Instead of occurrence probabilities calculated from a sample of a few thousand, expert (human) experience can be used and a functioning rule system can be created from a few measured data sets. The rule system sometimes requires refinement, but even this can be done by an expert. With FBDL, the rule system can be easily created.

Both the μFRI library and the FIVE IP FPGA implementation support the rule base created by machine learning or the tuning of an existing rule base. For details see [31], [35], [36] and [74].

In the case of the FIVE method, data clustering can be omitted. However, clustering was necessary in the case of the Bayesian classifier in order to compress the training sample.

A classification procedure with the use of μFRI and FIVE IP that also can be used on embedded systems.

6.4. Practical application

Rule based fast classifier that is implemented as software on a microcontroller with μFRI and as hardware on FPGA with FIVE IP. The advantage of the rule based solution is that the classifier can also be created based on an expert knowledge base. Other possible applications are in IoT (Internet of Things) systems in the recognition of behaviors and patterns and early fault detection or maintenance prediction. With an FPGA implementation, high-speed applications such as real-time traffic monitoring of a computer network are also possible.

6.5. Summary

I compared the results under the same conditions of the rule-based classifier using the FIVE method with the results of the popular Naive Bayes classifier. Based on the results, the FIVE method is also suitable to function as a classification procedure. Its advantage compared to the Naive Bayes classifier is the smaller amount of data to be collected and the applicability of the expert knowledge base. In relation to this topic, thesis IV. was established. For more information see: [S5], [S7], [S8] and [S10].

7. Theses

I. The μFRI library is suitable for implementing FIVE FRI on microcontrollers. The knowledge base provides the possibility to dynamically change the parameters without system shutdown. The library is suitable for developing adaptive applications in embedded systems.

Corresponding papers: [S5], [S7], [S8] and [S10].

II. In the case of the implementation of the μFRI library on the closely coupled ARM A72 parallel architecture, the dynamically changing distributed fuzzy knowledge base can be efficiently allocated up to a total memory requirement of 1720 kB on 1 processor core.

The memory requirement of an arbitrary rule base can be calculated using the following formula:

$$T = 8N_{UE} + 12N_U + 12N_A + 14N_R + 12N_{RB}$$

where the N_{UE} is the total number of universe elements, the N_U is the total number of universes, the N_A is the total number of the antecedents in the rules, the N_R is the total number of rules, the N_{RB} is the total number of rulebases, and the T is the total memory requirement in bytes. The minimum allocated memory demand is 58 byte, when $N_{UE}=N_U=N_A=N_R=N_{RB}=1$.

Corresponding papers: [S3], [S4] and [S6].

III. The modules of the FIVE IP library are suitable for the FPGA resource efficient implementation of the FIVE FRI method on the FPGA. The system which is based on the FIVE IP library is able to dynamically change the knowledge base without system shutdown, and to create adaptive applications in embedded systems.
Corresponding papers: [S1], [S2], [S8] and [S13].

IV. Using the created μFRI library or the FIVE IP modules, an efficient classification algorithm based on the FIVE FRI method can be developed. The resulting classification algorithm is suitable for direct use in embedded systems.
Corresponding papers: [S5], [S7], [S8] and [S10].

8. Summary

Ethological behaviour models and behaviour based control can be implemented wtih the FIVE method. The operation of behaviour based system is described by a set of rules. The FIVE method is a fuzzy interpolation method that places fuzzy sets in a multidimensional, so-called fuzzy environment for interpolation. The FIVE method is a lightweight interpolation that is also suitable for use on embedded systems. The FBDL language helps to describe the behaviours and rules.

In the dissertation I created a version of FIVE method which is optimized for embedded systems in languge C. Due to its low memory requirements, it can also be used successfully on microcontrollers. The μFRI library is also the basis for the implementation of the FIVE method FPGA. Various robots were developed based on behaviour based control during the research period. The behaviour implementations are based on μFRI library for ARM microcontrollers and application processor systems. The data structure of the μFRI library uses the structure of the FBDL language to help with easier transcription. The μFRI library allows changing parameters dynamically during operation for systems that support dynamic memory allocation. The entire rulebase can be changed at runtime.

I examined how the μFRI library behaves on a multiprocessor system. I investigated the computation times with non-cached and cached data on a tightly and loosely coupled system with 1, 2, and 3 processor cores. Although the μFRI library allows each rule base to be computed independently, it works more efficiently between 1 and 10,000 number of rulebases (20,000 rules) due to the short computation time and low memory requirements when one processor core performs the calculation. I examined that how much faster execution is done on multiple processor cores compared to single-processor calculations. Based on the ratio of the computational times and I used Amdahl's formula. For the studies, I used the popular ARM A53 and ARM A72 architectures during the research.

I created a hardware implementation for FPGA in the Verilog hardware description language to further accelerate the FIVE method. The FIVE IP is operating with 8 bits integers by default at maximum 4 MHz frequency but the bit witdh is paramemterizable. I compared the hardware requirements of my implementation with a tool called Xilinx High Level Systhesis, which can translate from specially modified C code into a descriptive language. I managed to make my own implementation use less hardware resources on the FPGA. The hardware implementation I created allows me to change the parameters dynamically during operation.

I used the FIVE method – implemented with μFRI library – as a classification procedure. By processing the sensor data of a real robot moving in a maze, I determined by rules what wall pattern is in the given square maze cell around the robot. The robot was equipped with 4 simple infrared sensors and used them to detect walls and obstacles. I compared the FIVE-based classifier with the classic Naiv-Bayes classifier, and in the end, the two solutions hit the position of the walls around the robot with equal probability.

Further steps in the research include the generation of a set of rules based on the knowledge gained in Chapter 7, using statistical methods to help develop machine learning in addition to existing Q-learning. The goal is to generate rule bases from the behavior presented by an expert that can be tuned by a Q-learning method. Furthermore, the goal is to further improve the dynamically reconfigurable hardware implementation by adapting it to modern solutions provided by Xilinx.

During the research, I have created solutions that allow the efficient use of the FIVE method on embedded systems on both processor and hardware systems, as well as the dynamic parameterization and tuning of rulebases.

9. References

[1] Szilveszter Kovács, Dávid Vincze, Márta Gácsi, Ádám Miklósi, and Péter Korondi, "Ethologically inspired robot behavior implementation," in *2011 4th International Conference on Human System Interactions, HSI 2011*, May 2011, pp. 64–69.

[2] Michiel P. De Looze, Tim Bosch, Frank Krause, Konrad S. Stadler, and Leonard W. O'sullivan, "Exoskeletons for industrial application and their potential effects on physical work load," *Ergonomics*, vol. 59, pp. 671–681, 2016.

[3] Universal Robots, "www.universal-robots.com," Tech. rep. 2020.

[4] Stuart Shepherd and Alois Buchstab, "Kuka robots on-site," in *Robotic Fabrication in Architecture, Art and Design 2014*.: Springer, 2014, pp. 373–380.

[5] Simon Blackmore, Bill Stout, Maohua Wang, Boris Runov, and others, "Robotic agriculture–the future of agricultural mechanisation," in *Proceedings of the 5th European conference on precision agriculture*, 2005, pp. 621–628.

[6] Richard Bloss, "Mobile hospital robots cure numerous logistic needs," *Industrial Robot: An International Journal*, 2011.

[7] Bence Kovács et al., "Ethologically Inspired Robot Design," in *The 2nd International Conference on Cognitive Infocommunications (CogInfoCom2011)* , 2011, pp. 7-9.

[8] Péter Korondi, Ádám Miklósi, Szilveszter Kovács, and Antal Dóka, "Ethologically inspired models for human-robot interaction-state of the art," 2009.

[9] Dávid Vincze and Szilveszter Kovács, "Using fuzzy rule interpolation based automata for controlling navigation and collision avoidance behaviour of a robot," in *2008 IEEE International Conference on Computational Cybernetics*, 2008, pp. 79–84.

[10] Hongwei Mo, Qirong Tang, and Longlong Meng, "Behavior-based fuzzy control for mobile robot navigation," *Mathematical problems in engineering*, vol. 2013, 2013.

[11] Paolo Pirjanian, "Multiple objective behavior-based control," *Robotics and Autonomous Systems*, vol. 31, pp. 53–60, 2000.

[12] Michael Dooley, Nikolai Romanov, Paolo Pirjanian, Lihu Chiu, and Enrico Di Bernardo, Robotic game systems and methods, January 2014, US Patent 8,632,376.

[13] Lotfi A. Zadeh, "Information and control," *Fuzzy sets*, vol. 8, pp. 338–353, 1965.

[14] Johanyák Zsolt Csaba and Dr. Tóth Tibor, "Fuzzy szabály-interpolációs módszerek és mintaadatok alapján történő automatikus rendszergenerálás," Ph. D. értekezés, Miskolci Egyetem, Hatvany József Informatikai Tudományok, Ph.D. dissertation 2007.

[15] Kovács Szilveszter, "Fuzzy logic control," Technical University of Budapest, Faculty of Informatics and Electrical Engineering, Master's thesis 1993.

[16] Szilveszter Kovács, "Extending the fuzzy rule interpolation" FIVE" by fuzzy observation," in *Computational Intelligence, Theory and Applications*.: Springer, 2006, pp. 485–497.

[17] LászlóT Kóczy and Kaoru Hirota, "Approximate reasoning by linear rule interpolation and general approximation," *International Journal of Approximate Reasoning*, vol. 9, pp. 197–225, 1993.

[18] LászlóT Kóczy and Kaoru Hirota, "Interpolative reasoning with insufficient evidence in sparse fuzzy rule bases," *Information Sciences*, vol. 71, pp. 169–201, 1993.

[19] Domonkos Tikk et al., "Stability of interpolative fuzzy KH controllers," *Fuzzy Sets and Systems*, vol. 125, pp. 105–119, 2002.

[20] Gy Vass, L. Kalmár, and L. T. Kóczy, "Extension of the fuzzy rule interpolation method," in *Proc. Int. Conf. Fuzzy Sets Theory Applications*, 1992, pp. 1–6.

[21] Péter Baranyi, László T. Kóczy, and Tamás D. Gedeon, "A generalized concept for fuzzy rule interpolation," *IEEE Transactions on Fuzzy Systems*, vol. 12, pp. 820–837, 2004.

[22] Suman Das, Debjani Chakraborty, and László T. Kóczy, "Linear fuzzy rule base interpolation using fuzzy geometry," *International Journal of Approximate Reasoning*, vol. 112, pp. 105–118, 2019.

[23] Szilveszter Kovács, Márta Gácsi, Dávid Vincze, Péter Korondi, and Ádám Miklósi, "A novel, ethologically inspired HRI model implementation: Simulating dog-human attachment," in *2nd International Conference on Cognitive Infocommunications*, 2011, pp. 1–4.

[24] Szilveszter Kovács and László T. Kóczy, "Application of interpolation-based fuzzy logic reasoning in behaviour-based control structures," in *2004 IEEE international conference on fuzzy systems (IEEE Cat. No. 04CH37542)*, vol. 3, 2004, pp. 1543–1548.

[25] Szilveszter Kovács and László T. Kóczy, "Application of the approximate fuzzy reasoning based on interpolation in the vague environment of the fuzzy rulebase in the fuzzy logic controlled path tracking strategy of differential steered AGVs," in *International Conference on Computational Intelligence*, 1997, pp. 456–467.

[26] Szilveszter Kovacs and Laszlo T. Koczy, "Approximate fuzzy reasoning based on interpolation in the vague environment of the fuzzy rulebase," in *Proceedings of IEEE International Conference on Intelligent Engineering Systems*, 1997, pp. 63–68.

[27] Szilveszter Kovács, "New aspects of interpolative reasoning," in *Proceedings of the 6th. International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Granada, Spain*, 1996, pp. 477–482.

[28] Imre Piller, Dávid Vincze, and Szilveszter Kovács, "Declarative language for behaviour description," in *Emergent trends in robotics and intelligent systems*.: Springer, 2015, pp. 103–112.

[29] Dávid Vincze and Szilveszter Kovács, "Incremental rule base creation with fuzzy rule interpolation-based Q-learning," in *Computational Intelligence in Engineering*.: Springer, 2010, pp. 191–203.

[30] Dávid Vincze and Szilveszter Kovács, "Fuzzy rule interpolation-based Q-learning," in *2009 5th International Symposium on Applied Computational Intelligence and Informatics*, 2009, pp55–60

[31] Tamás Tompa and Szilveszter Kovács, "Applying expert heuristic as an a priori knowledge for FRIQ-learning," *Acta Polytechnica Hungarica*, vol. 17, pp. 27–45, 2020.

[32] Imre Piller and Szilveszter Kovács, "FBDL: A Declarative Language for Interpolative Fuzzy Behavior Modeling," in *2019 IEEE 23rd International Conference on Intelligent Engineering Systems (INES)*, 2019, pp. 000295–000300.

[33] Balázs Pónya and László Czap, "Gömbrobot stabilizálása viselkedés leíró nyelvvel," in *ENELKO 2017 XVIII. Nemzetközi Energetika-Elektrotechnika Konferencia, SzámOkt 2017 XXVII. Nemzetközi Számítástechnika és Oktatás Konferencia, Kolozsvár 2017. október 12-15.*, Cluj-Napoca, RomĂˇnia, 2017.

[34] Gergő Lengyel, "Egyensúlyozó robot megvalósítása Fuzzy szabály interpolációval (FRI)," Miskolci Egyetem, Master's thesis 2017.

[35] Tamás Tompa and Szilveszter Kovács, "Clustering-based fuzzy knowledgebase reduction in the FRIQ-learning," in *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 2017, pp. 000197–000200.

[36] Tamás Tompa, Szilveszter Kovács, Dávid Vincze, and Mihoko Niitsuma, "Demonstration of expert knowledge injection in Fuzzy Rule Interpolation based Q-learning," in *2021 IEEE/SICE International Symposium on System Integration (SII)*, 2021, pp. 843-844.

[37] Gene M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," in *Proceedings of the April 18-20, 1967, spring joint computer conference*, 1967, pp. 483–485.

[38] Cypress. (2020) PSoC Products. [Online]. https://www.cypress.com/products/microcontrollers-mcus

[39] Xilinx. (2020) Products. [Online]. https://www.xilinx.com/products/silicon-devices/soc.html

[40] S. T. Microelectronics. (2020) Products. [Online]. https://www.st.com/en/microcontrollers-microprocessors/stm32h7-series.html

[41] Atul P. Godse and Deepali A. Godse, *Microprocessors & Microcontrollers.*: Technical publications, 2021.

[42] János Végh, Péter Molnár, and József Vásárhelyi, "A figure of merit for describing the performance of scaling of parallelization," *arXiv preprint arXiv:1606.02686*, 2016.

[43] János Végh, József Vásárhelyi, and Dániel Drótos, "The performance wall of large parallel computing systems," in *International Conference on Reliability and Statistics in Transportation and Communication*, 2018, pp. 224–237.

[44] STMicroelectronics, "STM32F4 datasheet," p. 80, Tech. rep.. [Online]. https://www.st.com/resource/en/reference_manual/dm00031020-stm32f405-415-stm32f407-417-stm32f427-437-and-stm32f429-439-advanced-arm-based-32-bit-mcus-stmicroelectronics.pdf

[45] Dávid Vincze and Szilveszter Kovács, "Performance Optimization of the Fuzzy Rule Interpolation Method" FIVE".," *J. Adv. Comput. Intell. Intell. Informatics*, vol. 15, pp. 313–320, 2011.

[46] Ulrich Drepper, "What every programmer should know about memory," *Red Hat, Inc*, vol. 11, p. 2007, 2007.

[47] Haohuan Fu and Robert G. Clapp, "Eliminating the memory bottleneck: an FPGA-based solution for 3D reverse time migration," in *Proceedings of the 19th ACM/SIGDA international symposium on Field programmable gate arrays*, 2011, pp. 65–74.

[48] Andreas Nowatzyk, Fong Pong, and Ashley Saulsbury, "Missing the memory wall: The case for processor/memory integration," in *23rd Annual International Symposium on Computer Architecture (ISCA'96)*, 1996, pp. 90–90.

[49] Digilent Inc. (2017) Zybo reference manual. [Online]. https://reference.digilentinc.com/_media/reference/programmable-logic/zybo/zybo_rm.pdf

[50] RaspberryPi.org. (2020) Raspberry Pi 4 B. [Online]. https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/

[51] Broadcom. (2021) BCM5871 adatlap. [Online]. https://docs.broadcom.com/doc/1211168571391

[52] Kernel.org. (2020) Scheduler. [Online]. https://docs.kernel.org/scheduler/

[53] József Vásárhelyi and János Végh, "Clock Around Embedded Systems and Reconfigurable Systems," 2013.

[54] Xilinx. (2022) Xilinx Unified Software Platform. [Online]. https://www.xilinx.com/products/design-tools/vitis.html

[55] Kovács Szilveszter Tompa Tamás, "Heuristically accelerated FRIQ-learning," *IEEE 20th Jubilee International Symposium on Intelligent Systems and Informatics (SISY 2022), Szabadka, Szerbia : IEEE (2022)*, p. 6, 2022.

[56] Daniele Bagni, A. Di Fresco, J. Noguera, and F. M. Vallina, "A zynq accelerator for floating point matrix multiplication designed with vivado hls," *Application note, January*, 2016.

[57] Xilinx. (2013) Xilinx, Introduction to High-Level Synthesis with Vivado HLS. [Online]. www.xilinx.com

[58] Xilinx, "UltraFast High-Level Productivity Desing Methodology Guide, UG1197," , 2016.

[59] Xilinx, "Vivado Design Suite - High Level Synthesis, UG902," , 2012, 07. 25.

[60] Xilinx, "High-Level Synthesis with Vivado HLS, heterogen_xilinx_hls.pdf," , 2012.

[61] Daniel T. Larose, *Data mining and predictive analytics*.: John Wiley & Sons, 2015.

[62] Irina Rish and others, "An empirical study of the naive Bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, 2001, pp. 41–46.

[63] Farzad Zafarani and Chris Clifton, "Differentially private naive bayes classifier using smooth sensitivity," *arXiv preprint arXiv:2003.13955*, 2020.

[64] Manuel Baltieri, "A Bayesian perspective on classical control," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–8.

[65] Baosheng He, *New Bayesian methods for quality control applications*.: The University of Iowa, 2018.

[66] Rainer Deventer, Joachim Denzler, and Heinrich Niemann, "Bayesian control of dynamic systems," in *Innovations in Intelligent Systems.*: Springer, 2004, pp. 21–50.

[67] Pedro A. Ortega. Bayesian Control Rule. [Online]. https://www.adaptiveagents.org/lib/exe/fetch.php?media=bayesiancontrolrule.pdf

[68] Pedro Alejandro Ortega, Daniel Alexander Braun, and Simon Godsill, "Reinforcement learning and the Bayesian control rule," in *International Conference on Artificial General Intelligence*, 2011, pp. 281–285.

[69] Hably Alexandra, "Szériában gyártható half-size MicroMouse robot tervezése," in *Konzulens: Szayer Géza, Tajti Ferenc*, BME-MOGI, Budapest TDK dolgozat, 2014, p. 35.

[70] Ferenc Lilik, Szilvia Nagy, and László T. Kóczy, "Wavelet based fuzzy rule bases in pre-qualification of access networks' wire pairs," in *AFRICON 2015*, 2015, pp. 1–5.

[71] Ferenc Lilik, Szilvia Nagy, Melinda Kovács, Szonja Krisztina Szujó, and László T. Kóczy, "Interpolative decisions in the fuzzy signature based image classification for liver CT," in *2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2021, pp. 1–6.

[72] Szilvia Nagy, Ferenc Lilik, and Laszló T. Kóczy, "Entropy based fuzzy classification and detection aid for colorectal polyps," in *2017 IEEE AFRICON*, 2017, pp. 78–82.

[73] Szilvia Nagy, Ferenc Lilik, and László T. Kóczy, "Applicability of various wavelet families in fuzzy classification of access networks' telecommunication lines," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2017, pp. 1–6.

[74] Tamás Tompa and Szilveszter Kovács, "Determining the minimally allowed rule-distance for the incremental rule-base contruction phase of the FRIQ-learning," in *2018 19th International Carpathian Control Conference (ICCC)*, 2018, pp. 480–483.

[75] Xilinx, "High-Level Systhesis with Vivado HLS," 2012, heterogen_xilinx_hls.pdf.

[76] Paolo Pirjanian, "Behavior coordination mechanisms-state-of-the-art," Technical report, University of Southern California, Institute for Robotics â€¦, Tech. rep. 1999.

[77] Pedro A. Ortega and Daniel A. Braun, "A bayesian rule for adaptive control based on causal interventions," *arXiv preprint arXiv:0911.5104*, 2009.

[78] Andreas Olofsson, Tomas Nordström, and Zain Ul-Abdin, "Kickstarting high-performance energy-efficient manycore architectures with epiphany," in *2014 48th Asilomar Conference on Signals, Systems and Computers*, 2014, pp. 1719–1726.

[79] Amin Shaka Aunali and D. Venkatesan, "Bayesian Approach in Control Charts Techniques," *Int. J. Sci. Res. in Mathematical and Statistical Sciences Vol*, vol. 6, p. 2, 2019.

[80] Kovács Szilveszter. FIVE notes.

10. Author's publications

[S1] *Bartók, Roland; Vásárhelyi, József*: **Design of a FPGA Accelerator for the FIVE Fuzzy Interpolation Method**, INTERNATIONAL JOURNAL OF COMPUTER APPLICATIONS IN TECHNOLOGY, 68: 4 pp. 321-331. Paper: 11 , 11 p. (2022)

[S2] *Bartók, Roland*: **Fuzzy szabály interpolációs eljárás HLS-el történő hardveres megvalósíthatóságának vizsgálata**: Analysis of hardware implemented fuzzy rule interpolation, implemented with HLS In: Dr. Sebestyén-Pál, György; Dr. Szabó, Loránd; Dr. Biró, Károly-Ágoston (szerk.) ENELKO 2019 SzámOkt 2019 : XX. Nemzetközi Energetika-Elektrotechnika Konferencia, XXIX. Nemzetközi Számítástechnika és Oktatás Konferencia Kolozsvár, Románia : Erdélyi Magyar Műszaki Tudományos Társaság (EMT), (2019) pp. 115-118. , 4 p.

[S3] *Bartók, Roland ; Vásárhelyi, József*: **Examining Cache Handling of the FIVE Method on Multicore Systems** In: Szakál, Anikó (szerk.) SAMI 2019 : IEEE 17th World Symposium on Applied Machine Intelligence and Informatics Herlany, Szlovákia : IEEE, (2019) pp. 141-146. Paper: 24_sami2019 , 5 p.

[S4] *Bartók, Roland ; Vásárhelyi, József*: **A FIVE módszer párhuzamosíthatóságának vizsgálata Parallella mikroszámítógépen**: Examining the parallelism of FIVE method on Parallella microcomputer In: Bíró-Károly, Ágoston; Sebestyén-Pál, György; Szabó, Lóránd (szerk.) ENELKO 2018 XIX. Nemzetközi Energetika-Elektrotechnika Konferencia SzámOkt 2018 XXVIII. Nemzetközi Számítástechnika és Oktatás Konferencia Erdélyi Magyar Műszaki Tudományos Társaság (EMT), (2018) p. 134 , 6 p.

[S5] *Bartók, Roland ; Vásárhelyi, József*: **Fuzzy Rule Interpolation Based Object Tracking and Navigation for Social Robot** Lecture Notes In Mechanical Engineering 2018 pp. 370-375. , 6 p. (2018)

[S6] *Bartók, Roland ; Vásárhelyi, József*: **Parallelization of FIVE method on multicore embedded system** In: Drótos, Dániel; Vásárhelyi, József; Czap, László; Ivo, Petráš (szerk.) Proceedings of the 19th International Carpathian Control Conference (ICCC 2018) Piscataway (NJ), Amerikai Egyesült Államok : IEEE, (2018) pp. 400-403., 4 p.

[S7] *Bartók, Roland ; Korcsok, Erika ; Vásárhelyi, József:* **Fuzzy szabály interpoláció alapú objektumkövetés megvalósítása Robot Operációs Rendszerben**: Implementation of Fuzzy Rule Interpolation Based Object Tracking in Robot Operating System In: Bíró, Károly-Ágoston; Sebestyén-Pál, György (szerk.) ENELKO 2017 : XVIII. Energetika-Elektrotechnika Konferencia : SzámOkt 2017 : XVII. Nemzetközi Számítástechnika és Oktatás Konferencia Kolozsvár, Románia : Erdélyi Magyar Tudományos Társaság, (2017) pp. 123-126. , 4 p.

[S8] *Bartók, Roland ; Vásárhelyi, József*: **Two Methods for Autonomous Robot Obstacle Sensing and Application Programming Interface for Fuzzy Rule Interpolation** In: Dan, Popescu; Dorin, Şendrescu; Monica, Roman; Elvira, Popescu; Lucian, Bărbulescu (szerk.) 2017 18th International Carpathian Control Conference (ICCC) Craiova, Románia: IEEE, (2017) pp. 87-92. Paper: 7970376 , 6 p.

[S9] *Bartók, Roland ; Pintér, Judit Mária*: **Beszéd alapú ember-gép interfész fejlesztése szociális robothoz: Development of Speech-based Human-machine Interface for Social Robot** In: OGÉT 2017: XXV. Nemzetközi Gépészeti Konferencia : 25th International Conference on Mechanical Engineering Kolozsvár, Románia : Erdélyi Magyar Műszaki Tudományos Társaság (EMT), (2017) pp. 67-70., 4 p.

[S10] *Bartók, Roland ; Ahmed, Bouzid ; Vásárhelyi, József ; L Kiss, Márton*: **Wall and Object Detection with FRI and Bayes-Classifier for Autonomous Robot** Lecture Notes In Mechanical Engineering F12 pp. 383-389. , 7 p. (2017)

[S11] *Bartók, Roland ; L, Kiss Márton ; Dr. Vásárhelyi, József ; Dr. Kovács, Szilveszter ; Ahmed, Bouzid*: **Embedded behavioral model implementation** In: Ivo, Petráš; Igor, Podlubny; Ján, Kačur (szerk.) Proceedings of the 2016 17th International Carpathian Control Conference (ICCC) [s. l.] - Nemzetközi, Nemzetközi : IEEE, (2016) pp. 35-40., 6 p.

[S12] *Bartók, Roland ; L, Kiss Márton ; Vásárhelyi, József ; Ferenczi, István*: **Holonomikus hajtású robot vezérlése Fuzzy alapú viselkedés leírással** In: Biró, Károly-Ágoston; Sebestyén-Pál, György (szerk.) ENELKO 2015 XVI Nemzetközi Energetika-Elektrotechnika konferencia; SzámOkt 2015 XXV. Nemzetközi Számítástechnika és Oktatás Konferencia Arad, Románia : Erdélyi Magyar Műszaki Tudományos Társaság (EMT), (2015) pp. 185-188. , 4 p.

[S13] *Bartók, Roland ; Vásárhelyi, József*: ***A fuzzy rule interpolation base algorithm implementation on different platforms*** In: Ivo, Petras; Igor, Podlubny; Jan, Kacur; Vásárhelyi, József (szerk.) Proceedings of the 16th International Carpathian Control Conference Miskolc, Magyarország : IEEE IAS/IES/PELS, (2015) pp. 37-40. , 4 p.