The Design of Embedded Fuzzy Logic Controller for Autonomous Mobile Robots

Youngwoo Jeong, Won Sik Jeong, Jin Young Shin, and Seung Eun Lee
Dept. of Electronic Engineering
Seoul National University of Science and Technology
Seoul, Republic of Korea
{jeongyoungwoo, jeongwonsik, shinjinyoung, *seung.lee}@seoultech.ac.kr

Abstract— **In this paper, we propose an embedded fuzzy logic controller (EFLC) with a focus on addressing the challenges arising from the increasing number of fuzzy rules in autonomous mobile robots (AMRs). The Cortex-M0 is selected as the main processor, and the entire logic was implemented utilizing Verilog HDL. In order to evaluate performance, we analyzed the execution time difference between a software-only implementation of a fuzzy logic controller (FLC) on Cortex-M0 and a hardware-accelerated implementation with the proposed EFLC. The effectiveness and feasibility of the proposed design were demonstrated through simulations and field-programmable gate array (FPGA) implementation.**

Keywords; formatting; Fuzzy Logic Controller, Embedded System, Dedicated Circuit, Autonomous Mobile Robots

I. INTRODUCTION

Fuzzy logic has gained popularity as a powerful tool for effectively addressing complex and uncertain systems through the utilization of approximate reasoning [1]. This logic provides a framework for dealing with imprecise information and making decisions based on fuzzy rules that model human-like reasoning.

In applications such as autonomous mobile robots (AMRs), the ability to perceive and understand the surrounding environment is crucial for safe and efficient operation [2]. AMRs are equipped with a variety of sensors, including cameras, LiDAR, radar, and ultrasonic sensors, to gather information about their surroundings. As the number of sensors increases and the data they generate grows, the complexity of fuzzy rules also increases [3]. Efficient and timely decision-making plays a critical role in dynamic environments, particularly in tasks such as obstacle avoidance, optimal path planning, and making well-informed decisions [4]. However, the utilization of conventional methods to design a fuzzy logic controller (FLC) for AMRs introduces delays that impede the attainment of real-time capabilities in the embedded system.

For the purpose of overcoming the computational challenges posed by complex fuzzy rule sets and ensuring real-time decision-making capabilities in AMRs, one of the alternative approaches is to design dedicated hardware circuits specifically for performing fuzzy logic operations. These circuits efficiently handle the growing complexity of fuzzy rules and enable prompt responses to sensory inputs by parallelizing frequently used function blocks.

In this paper, we propose an embedded fuzzy logic controller (EFLC) for AMRs. For catering to the requirements of embedded systems, we adopt the ARM Cortex-M0 core and fixed-point arithmetic unit, both of which are compatible with the embedded systems. We validated the feasibility of the proposed design through simulation and a field-programmable gate array (FPGA) implementation.

II. ARCHITECTURE

Figure 1 represents the architecture of the proposed EFLC. The Cortex-M0 and EFLC are integrated through the advanced high-performance bus (AHB), enabling access to the EFLC via the AHB slave interface bus controller. The EFLC consists of blocks for the fuzzifier, rule evaluator, aggregator, and defuzzifier, which are essential components of fuzzy logic. All blocks are implemented based on a fixed-point arithmetic unit with a 22-bit integer part and a 10-bit fractional part.

The fuzzifier is a block in fuzzy logic that converts crisp values to fuzzy values, incorporating membership functions such as R-function, L-function, and triangular function. Through the Cortex-M0, criteria for each membership function are determined flexibly. The fuzzy rule evaluator is a module that encompasses various operations for fuzzy inference in the FLC.

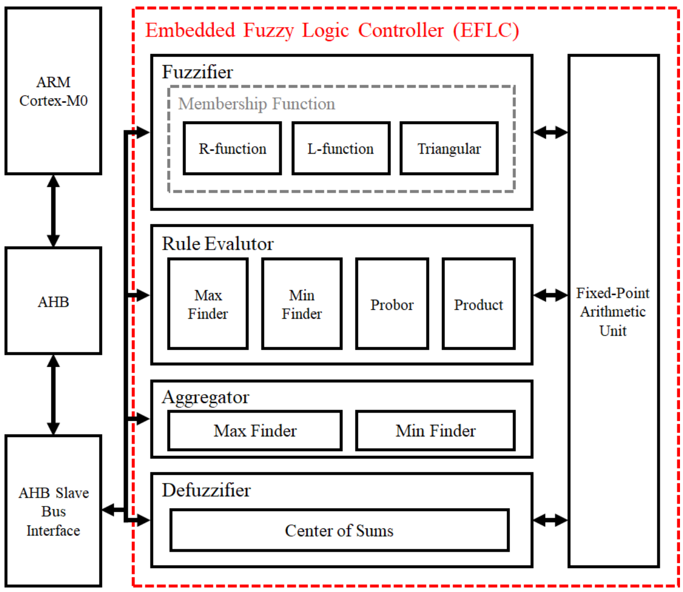

Figure 1. Architecture of the proposed EFLC with Cortex-M0.

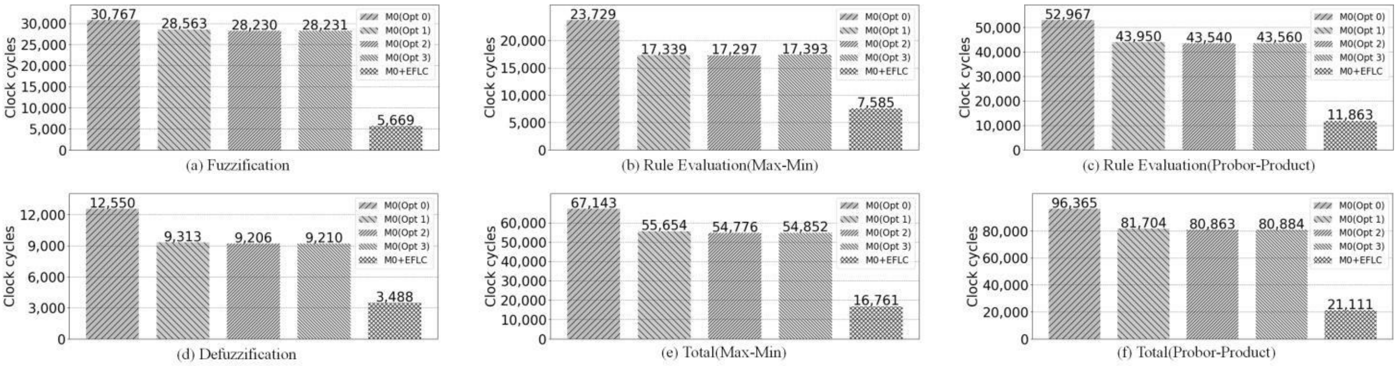

Figure 3. Clock cycles for each step in the simulations.

This module includes max finder (indicating "OR"), probor (representing probabilistic "OR"), min finder (indicating "AND"), and product (representing probabilistic "AND") blocks. These components execute operations on two inputs using comparators, adders, and multipliers to generate the intended output. The aggregator is a module which is responsible for integrating the outputs of rule evaluator. Lastly, the defuzzifier consists of the Center of Sum module, which is one of the various defuzzification methods. The Center of Sum method combines these outputs to obtain the final crisp value. By leveraging the parallel processing of dedicated hardware circuits, the proposed EFLC efficiently manages a larger number of fuzzy rules while preserving real-time performance.

III. Experiment

The experiments were carried out based on the previously designed collision avoidance algorithm for AMRs [5]. A 1-channel LiDAR sensor capable of measuring distances up to 10 meters was employed and 23 number of data points were obtained from the LiDAR sensor. Figure 2 represents the membership functions of the input and output. For these values, 53 number of Mamdani method-based fuzzy logic were implemented. Figure 3 depicts an analysis of the clock cycles needed for fuzzy logic calculations specifically for a motor of 4-wheel drive AMRs. The required clock cycles for the aggregator were measured as part of defuzzification. A comparison was made between the performance of FLC implemented solely on the Cortex-M0 processor with various optimization levels (opt 0-3) and the hardware-accelerated implementation with the EFLC. The Max-Min method demonstrated a maximum execution time gain of 4x, and the Probor-Product method achieved a maximum execution time gain of 4.56x.

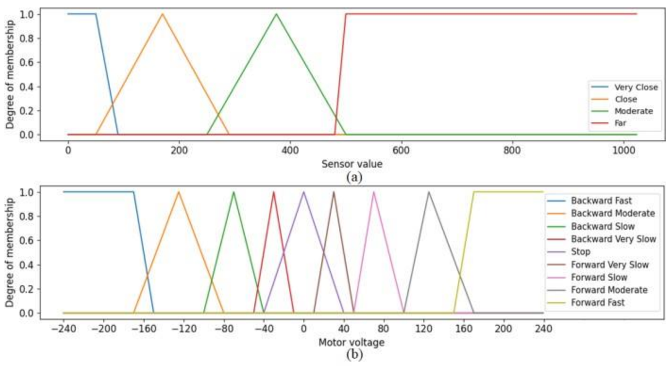

Figure 2. Membership functions (a) One of the inputs (b) Output

TABLE I. Absolute Error of the EFLC

	Max-Min	*Probor-Product*
Mean	0.08	0.274
Maximum	0.467	2.459

Table 1 presents the results of the analysis regarding the absolute error incurred by utilizing a fixed-point arithmetic unit. The results indicate a maximum error of 2.459 for the output values, which has a negligible impact on the motor voltage.

IV. Conclusion

In this paper, we proposed the EFLC for handling large number of fuzzy rules in the embedded systems. In the experiment, we compared the execution time between the software-only and the hardware-accelerated implementation for fuzzy logic through simulations. As a result, the proposed EFLC achieved a maximum execution time gain of 4.56x.

Acknowledgment

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2023-RS2022-00156295) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation) and by the Technology Innovation Program (RS-2022-00154902) funded By the Ministry of Trade, Industry & Energy(MOTIE, Korea).

References

[1] R. Das, S. Sen, and U. Maulik, “A Survey on Fuzzy Deep Neural Networks,” ACM Comput. Surv., vol. 53, no. 3, May 2020.

[2] Md. A. K. Niloy et al., “Critical Design and Control Issues of Indoor Autonomous Mobile Robots: A Review,” IEEE Access, vol. 9, pp. 35338–35370, 2021.

[3] X. Gu, J. Han, Q. Shen, and P. P. Angelov, “Autonomous learning for fuzzy systems: a review,” Artificial Intelligence Review, pp. 1–47, 2022.

[4] O. Varlamov, “‘Brains’ for Robots: Application of the Mivar Expert Systems for Implementation of Autonomous Intelligent Robots,” Big Data Research, vol. 25, p. 100241, 2021.

[5] Y. W. Jeong, K. H. Go and S. E. Lee, "Robot-on-Chip: Computing on a Single Chip for an Autonomous Robot," 2022 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, pp. 1-3, 2022.