

Research Article

Design and Implementation of Fuzzy Approximation PI Controller for Automatic Cruise Control System

Pallab Maji, Sarat Kumar Patra, and Kamalakanta Mahapatra

National Institute of Technology, Rourkela Department of Electronics and Communication Engineering, Rourkela, Odisha 769008, India

Correspondence should be addressed to Pallab Maji; pallab.vsp@gmail.com

Received 10 May 2015; Revised 5 October 2015; Accepted 12 October 2015

Academic Editor: Theo Van Der Weide

Copyright © 2015 Pallab Maji et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Fuzzy logic systems have been widely used for controlling nonlinear and complex dynamic systems by programming heuristic knowledge. But these systems are computationally complex and resource intensive. This paper presents a technique of development and porting of a fuzzy logic approximation PID controller (FLAC) in an automatic cruise control (ACC) system. ACC is a highly nonlinear process and its control is trivial due to the large change in parameters. Therefore, a suitable controller based on heuristic knowledge will be easy to develop and provide an effective solution. But the major problem with employing fuzzy logic controller (FLC) is its complexity. Moreover, the designing of Rulebase requires efficient heuristic knowledge about the system which is rarely found. Therefore, in this paper, a novel rule extraction process is used to derive a FLAC. This controller is then ported on a C6748 DSP hardware with timing and memory optimization. Later, it is seamlessly connected to a network to support remote reconfigurability. A performance analysis is drawn based on processor-in loop test with Simulink model of a cruise control system for vehicle.

1. Introduction

Fuzzy logic controllers (FLC) have gained high appreciation in nonlinear control [1, 2] although Proportional-Integral-Derivative (PID) controllers and model predictive controllers (MPC) are widely used in industrial processes [3–5]. This is mainly due to its aptitude which can counter nonlinear control problems by programming heuristic knowledge. These heuristics are derived from expert human knowledge and the input and output relationship of the process. However, it is vital that the heuristic knowledge is derived from plant experts. But it has been observed that skilled and expert operator for a process plant is a scarce resource. It is also a proven fact that human knowledge is often prone to error. Therefore, it becomes critical that correct system heuristics are used to derive the fuzzy control parameters. Many designs in the literature have used evolutionary techniques for fuzzy parameter (FCP) extraction [6–8]. However, these methods eliminate the human knowledge from these controllers, which is so imperative. This paper introduces a generic fuzzy logic controller system (G-FLCS) which extracts FCP from an

input-output relationship of a plant using genetic algorithm and provides a web based user interface (WebUI) for users to tweak these parameters for fine tuning of the fuzzy control system. This system provides a remote accessibility and control over systems. Internet-of-things (IOT) has already been a revolution in the automation industry. Fuzzy logic has been extensively used in ACC and collision avoidance system over the years [9–11]. However, this paper presents a system that controls an automatic cruise control (ACC) unit of a vehicle using fuzzy logic and connects it seamlessly to the Internet which will allow technical support engineers to remotely configure if it malfunctions. ACC has been used in this paper as an example subsystem of a vehicle, but this feature of remote reconfigurability can be extended to other modules of vehicle automation and advance driver assistance systems (ADAS). Figure 1 graphically describes the control strategy employed for the ACC.

This paper is organized as follows. In Section 2, mathematical modeling of ACC is described and is simulated in Section 3 for data collection. Fuzzy logic approximation control is explained in Section 4 and the proposed hardware

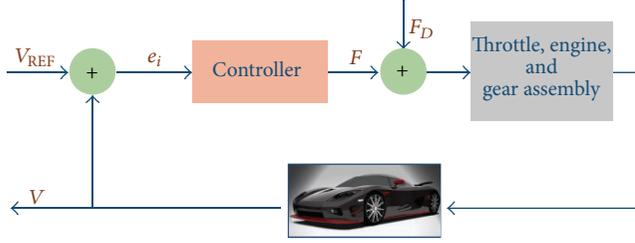


FIGURE 1: Block diagram of automatic cruise control system.

G-FLCS is implemented in Section 5. Performance analysis of this controller with reference to PID controller and model predictive controller is drawn in Section 6. The paper is finally concluded in Section 7 and followed by the references used during this research work.

2. Mathematical Modeling Automatic Cruise Control System [12]

Consider a vehicle of mass m moving at a velocity v . A force F is generated from the engine while a disturbance force F_d is resisting motion of the vehicle [12, 13]. Therefore, the equation of motion of the vehicle is given by

$$m \frac{dv}{dt} = F - F_D. \quad (1)$$

The vehicle engine generates force F which is proportional to the rate of injected fuel in the engine. This phenomenon in turn controls the throttle of the vehicle. Torque produced T at engine speed ω can be mathematically represented as

$$T(\omega) = T_m \left(1 - \beta \left(\frac{\omega}{\omega_m} - 1 \right)^2 \right), \quad (2)$$

where T_m is the maximum torque generated by the engine at full throttle to attain ω_m and the maximum engine speed with torque coefficient β . For a gear ratio n and wheel radius r , current velocity can be related to the engine speed by the following equation:

$$\omega = \frac{n}{r} v = \alpha_n v. \quad (3)$$

Therefore, the driving force can be computed as

$$F = \frac{mu}{r} T(\omega) = \alpha_n u T(\alpha_n v). \quad (4)$$

Basically, there are three major disturbance forces working on the vehicle, namely, gravitational force (F_G), rolling friction of the road and vehicle tires (F_R), and aerodynamic drag due to the body of the vehicle (F_A)

$$F_D = F_G + F_R + F_A. \quad (5)$$

The gravitational force acting on the vehicle can be modeled based on the slope of the roads

$$F_G = mg \sin \theta, \quad (6)$$

$$F_R = mg C_r \operatorname{sgn}(v), \quad (7)$$

$$F_A = \frac{1}{2} \rho C_d A v^2. \quad (8)$$

Combining (5) and (6), (1) becomes

$$m \frac{dv}{dt} = \alpha_n u T(\alpha_n v) - mg \sin \theta - mg C_r \operatorname{sgn}(v) - \frac{1}{2} \rho C_d A v^2. \quad (9)$$

3. Simulation of Automatic Cruise Control System

The mathematical model of an ACC presented in Section 2 is simulated in MATLAB/Simulink with a PID controller. The Simulink model as presented in Figure 2 is simulated and the system response is observed and recorded. This recorded dataset is further used to extract fuzzy control parameter (FCP) which drives the G-FLCS. This method is carried out using PID control algorithm and, therefore, the G-FLCS eventually approximates the system response of the PID controller. However, since the control algorithm uses fuzzy logic in its core, a fast and easy tracking is observed.

3.1. Control Algorithms. There are various control algorithms in literature that have been effective in automatic cruise control systems [8, 9, 14, 15]. However, these control algorithms do not provide a system which can be remotely controlled. The system responses cannot be fine-tuned on-field. Moreover, control algorithms like model predictive control (MPC) and PID control are generally sluggish in their approach. In this paper, the system response with proposed G-FLCS control algorithm is compared to system responses with MPC and PID control as presented in Figure 3.

3.2. Data Collection for Proposed Controller Design. A system response with maximum of four inputs and one output is recorded. Once the dataset is derived, it is passed on to the system for FCP extraction. The proposed method for FCP extraction is based on genetic algorithm. In this algorithm, there are 40 system parameters that are varied according to genetic algorithm to achieve two objectives, namely,

- (i) quick settling,
- (ii) tracking reference.

The Simulink model with initial FCP is used as an objective function which returns the transient and settling time. The coordinates of the membership functions are nonlinear inequality constraints. Thus, varying these FCP will provide an absolute tracking of reference (the codes can be downloaded from <https://goo.gl/Zb17fZ>).

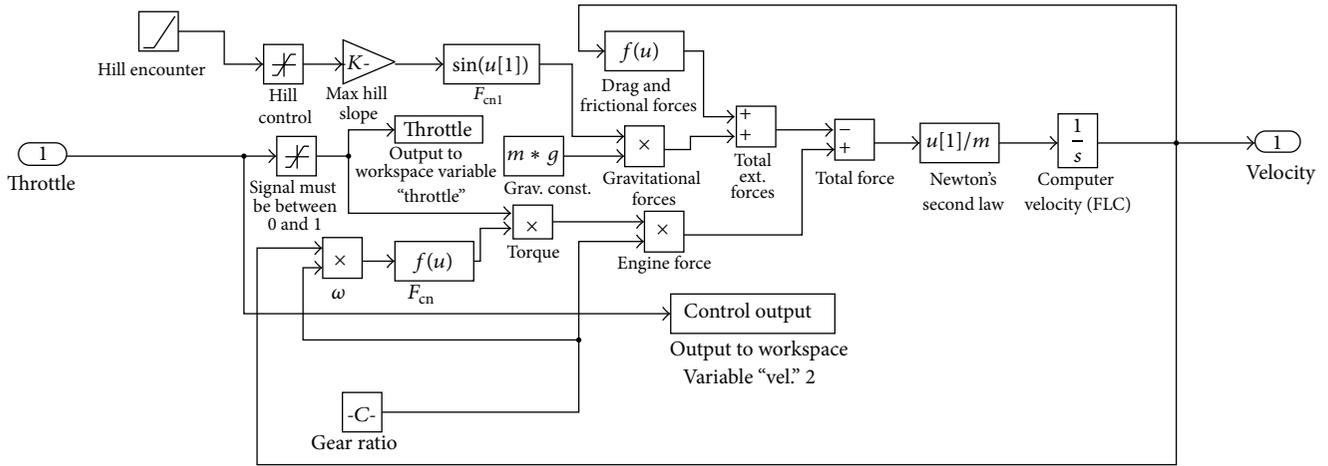


FIGURE 2: Simulink model of automatic cruise control.

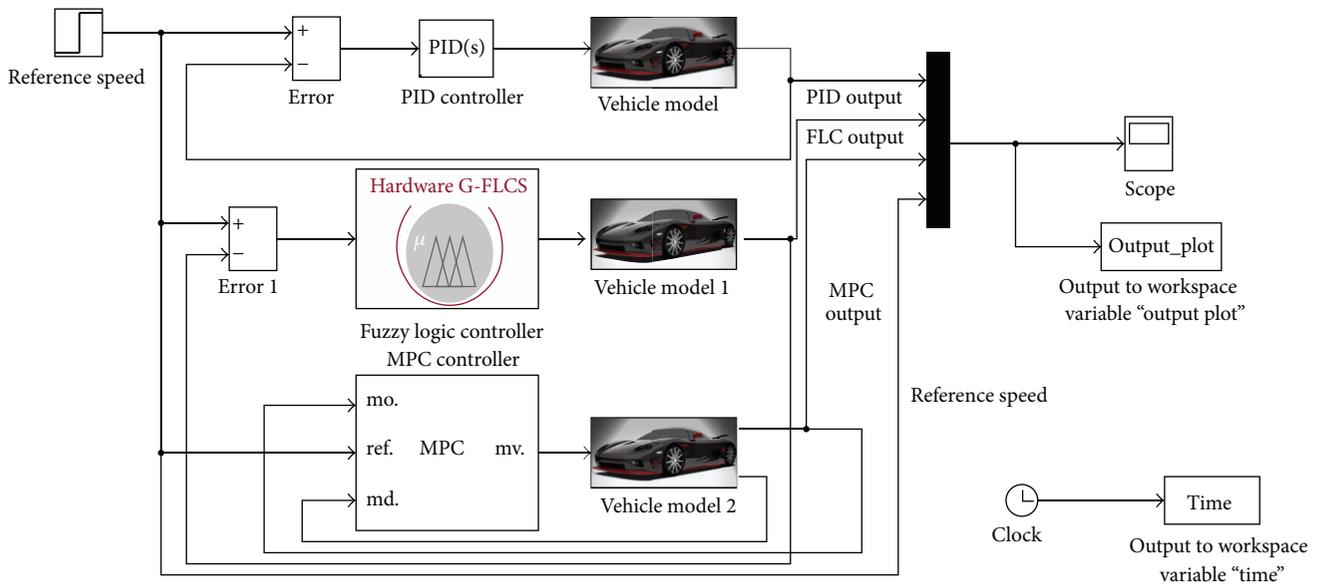


FIGURE 3: Simulink model of automatic cruise control system with MPC, PID, and proposed G-FLCS on C6748 DSP hardware.

Throughout this implementation process, data synchronization and communication between user and the hardware G-FLCS are critical as it can be observed in Figure 5. This figure depicts how data communication can be achieved with the help of various control signals between client-server and server-hardware G-FLCS. A web application provides a systematic user interface which collects data from authenticated users. The application waits for a new connection request on startup. On successful login, a user is presented with a web page containing four different tab-windows, each for basic parameters and information about inputs, outputs, and Rule-base. These windows will be preloaded with extracted parameters as shown in Figure 4 and can provide a way for fine-tuning the control parameters by an operator. In Figure 5, operations like connections, login, parameter collection, and communication with H-FLC through a server program are performed by the web application and are shaded in *Light*

Grey. When a user provides the fuzzy control parameters (FCP) data, the web application validates the entered data based on the following protocols.

- (i) All MFs have properly defined coordinates within specified range of operation.
- (ii) Number of inputs and outputs is correctly defined.
- (iii) Rules are validated according to Mamdani model.

After validation of new parameters, the server application connects serially to the hardware G-FLCS. Serial communication protocol is used in this paper for the ease of implementation. This data communication can be easily extended to industry standard controller area network (CAN) protocol. G-FLCS completes current execution and generates control signal to the system. Thereafter, it acknowledges any incoming serial communication request and starts receiving and

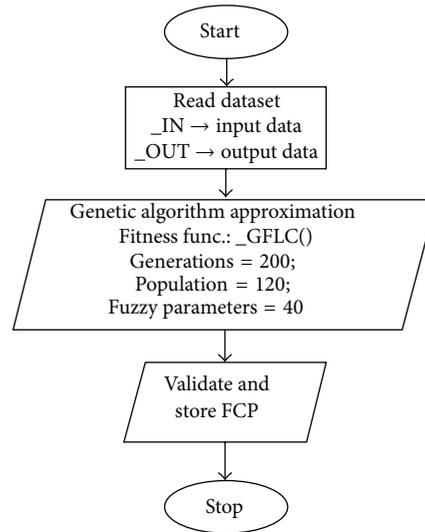


FIGURE 4: Fuzzy control parameter (FCP) extraction using genetic algorithm.

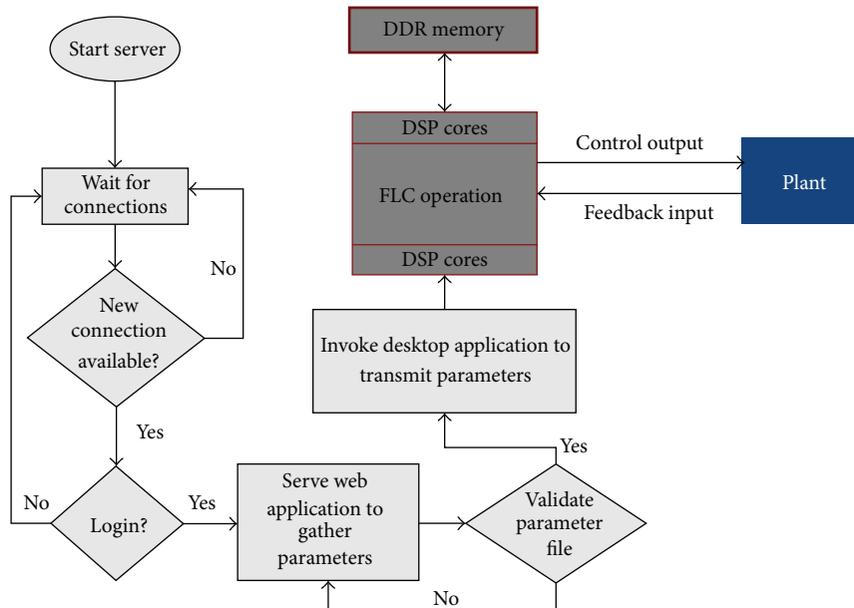


FIGURE 5: Dataflow of the proposed G-FLCS system.

storing fuzzy parameters in the data memory. Sys/BIOS is a widely used real time operating system for TI DSPs and has been used in this work to take care of the multitasking of fuzzy processes.

4. Fuzzy Logic Approximation Control (FLAC)

The proposed system architecture involves hardware-software codesign to present a complete reconfigurable G-FLC as shown in Figures 6(a) and 6(b). The WebUI in client-server model represents the software and the driver layer to interface the H-FLC through serial port. The DSP hardware receives FCP data serially and stores them. These parameters are segregated into two categories, namely, *Setup* and *Rulebase*

data. The driver layer in the hardware G-FLCS receives and acknowledges the data transmission.

5. Hardware Realization on TI C6748 DSP

5.1. Code Optimization. Code written in assembly (ASM) is processor-specific C code that can readily be ported from one platform to another. However, optimized ASM code runs faster than C and requires less memory space. Before optimizing, make sure that the code is functional and yields correct results. After optimizing, the code can be so reorganized and resequenced that the optimization process makes it difficult to follow. One needs to realize that if a C-coded algorithm is functional and its execution speed is satisfactory, there is

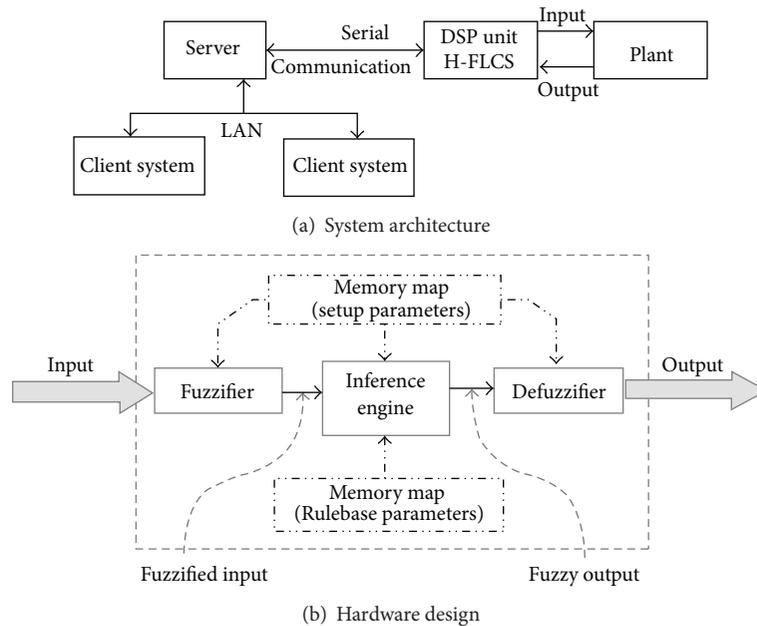


FIGURE 6: Proposed M-FRHC based H-FLC design architecture.

no need to optimize further. But this motivates us to stretch the optimization such that we can extract the best possible efficiency from the system [16].

Step 1. G-FLC is programmed in C language without using any compiler optimization levels.

Step 2. Intrinsic functions are used along with various optimization levels. Functions like minimum (min), maximum (max), product (prod), and division (div) are written as intrinsic functions and called necessary by the modules and submodules of G-FLCS.

Step 3. Thereafter, the profiler is used to determine and identify the functions and submodules that may need further optimization. Later, these functions are converted to linear ASM.

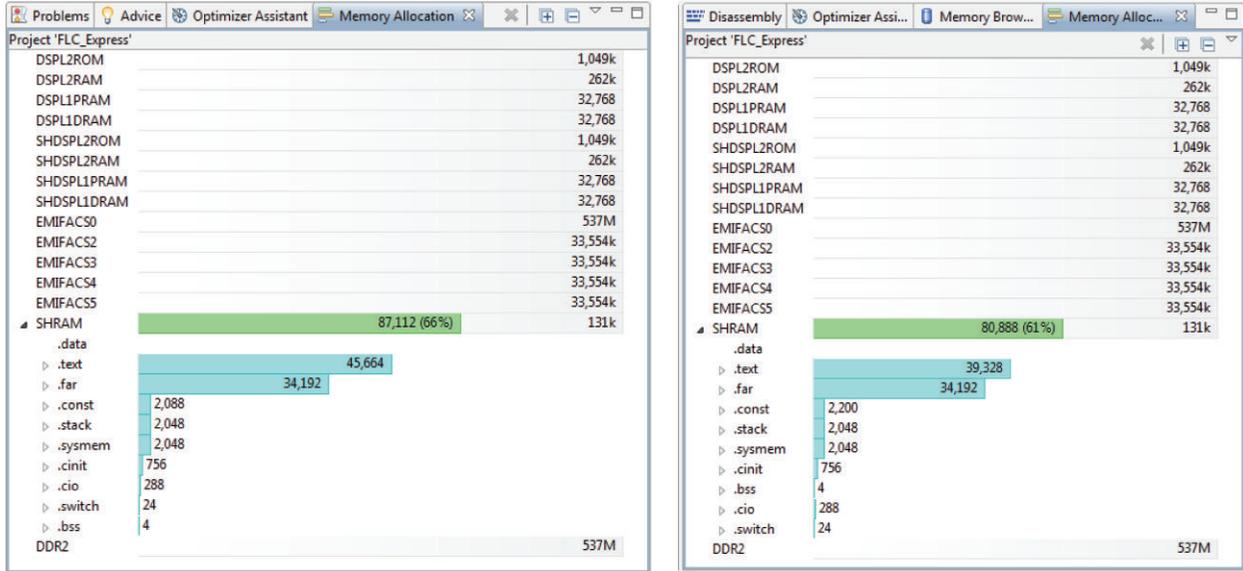
The compiler can insert calls to special functions in the run-time support library (RTS) to support operations that are not natively supported by the ISA. For example, the compiler calls `...c6xabi.divi()` (`_divi()` in COFF) function to perform 32-bit integer divide operation. Such functions are called compiler helper functions and result in a function call within the loop body. For example, in G-FLC, the compiler accomplishes the division operation by calling the compiler helper function “`_divi`” in fuzzification and defuzzification modules [16–18].

5.2. Code Implementation. Code Composer Studio (CCS) is a proprietary integrated development environment developed by TI for programming DSP and ARM processors. The design

is programmed in C language and optimized as described in Section 5. Thereafter, it is cross compiled using CCS v5.5 compiler and implemented on TMS320C6748 as a target DSP processor. This system represents the hardware G-FLCS as discussed previously. G-FLCS is connected to a server PC using on-board UART and provides a platform which is capable of accepting FCP file to operate as a standalone tunable G-FLCS. Detailed memory utilization of the built code is shown in Figure 7. It can be observed that the code size without optimization acquires 66% of the SHRAM and has a code size of 88 K in comparison to 61% of the SHRAM and code size of 80 K with code optimization technique described in Section 5.1.

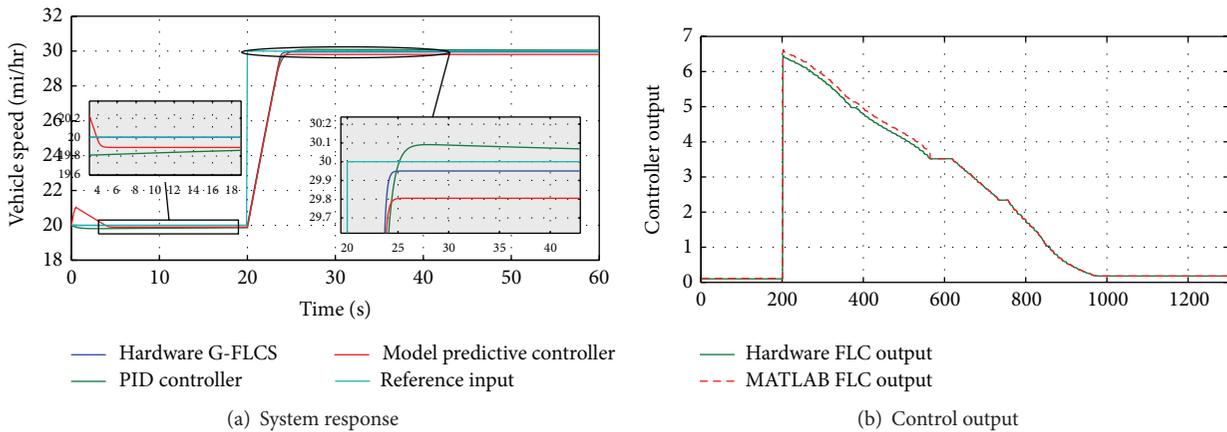
6. Performance Analysis

A performance analysis between MPC, PID, and proposed hardware G-FLCS is drawn after a hardware-in-loop simulation of Simulink model presented in Figure 3 is completed. The ACC system response under the influence of individual controllers is plotted in Figure 8(a). It is evident from comparing responses of these controllers that the fuzzy approximated PI controller has a very less settling time and a fast transient response. However, the fuzzy controller structure provides an option to fine-tune the control parameters on run-time. Moreover, mathematical model of the ACC plant was also not in the vicinity while formulation of fuzzy parameters was undertaken and Figure 4 abundantly explains this, hereby reducing the computational complexity which exponentially increases with the increase in input variables. The control output from the proposed hardware G-FLCS closely matches the control output from the Fuzzy Toolbox



(a) Without optimization (b) Optimization

FIGURE 7: Memory utilization of proposed system realized in TI C6748 DSP.



(a) System response (b) Control output

FIGURE 8: (a) Response of ACC under influence of various controllers, (b) control output of hardware G-FLCS and MATLAB Fuzzy Toolbox with same FCP.

of MATLAB. A mean square error of 0.13 was recorded. Figure 8(b) shows the control output of the two fuzzy logic controllers when operated on the ACC plant model.

7. Conclusion

It is evident from the responses of the three controllers presented in Figure 8(a) that proposed FLAC has a very less settling time and a fast transient response compared to others. Mathematical model of the plant was not in the vicinity while formulation of fuzzy parameters was undertaken. A parameter extraction technique inspired with genetic algorithm is implemented on an ACC process to provide the applicability of the proposed idea. The resultant hardware FLAC is found to have a low computational complexity and high code density and connects seamlessly to the local area network (LAN).

Appendix

Fuzzy Parameter File for ACDC Motor Control

Fuzzy PID approximation to automatic cruise control system is found at <https://goo.gl/zJhn5n>.

Nomenclature

- T : Engine torque constant
- β : Torque coefficient
- ω : Present engine speed
- ω_m : Maximum engine speed
- C_r : Coefficient of rolling friction
- C_d : Drag coefficient
- ρ : Density of air

A: Area of the vehicle
 g: Gravitational constant.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] T. Ross, *Fuzzy Logic with Engineering Applications*, John Wiley & Sons, 3rd edition, 2010.
- [2] J. M. Mendel and G. C. Mouzouris, "Designing fuzzy logic systems," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 11, pp. 885–895, 1997.
- [3] J. A. Laghari, H. Mokhlis, A. H. Abu Bakar, and H. Mohamad, "A fuzzy based load frequency control for distribution network connected with mini hydro power plant," *Journal of Intelligent and Fuzzy Systems*, vol. 26, no. 3, pp. 1301–1310, 2014.
- [4] M. J. Nalley and M. B. Trabia, "Control of overhead cranes using a fuzzy logic controller," *Journal of Intelligent and Fuzzy Systems*, vol. 8, no. 1, pp. 1–18, 2000.
- [5] E. A. Ramadan, M. El-Bardini, and M. A. Fkirin, "Design and FPGA-implementation of an improved adaptive fuzzy logic controller for DC motor speed control," *Ain Shams Engineering Journal*, vol. 5, no. 3, pp. 803–816, 2014.
- [6] A. H. Zavala and O. C. Nieto, "Fuzzy hardware: a retrospective and analysis," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 4, pp. 623–635, 2012.
- [7] G. Bosque, I. Del Campo, and J. Echanobe, "Fuzzy systems, neural networks and neuro-fuzzy systems: a vision on their hardware implementation and platforms over two decades," *Engineering Applications of Artificial Intelligence*, vol. 32, pp. 283–331, 2014.
- [8] S. Paul Sathiyar, S. Suresh Kumar, and A. Immanuel Selvakumar, "Particle swarm optimization technique for rule base optimization of FLC for low speed ACC vehicle," *ARPJ Journal of Engineering and Applied Sciences*, vol. 9, no. 6, pp. 981–987, 2014.
- [9] A. Hassan and G. Collier, "Adaptive cruise control for a robotic vehicle using fuzzy logic," in *Mechatronics 2013: Recent Technological and Scientific Advances*, pp. 535–542, Springer, Berlin, Germany, 2014.
- [10] A. Merah and K. Hartani, "A robust longitudinal control strategy for safer and comfortable automotive driving," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 7, no. 23, pp. 5026–5033, 2014.
- [11] A. Vahidi and A. Eskandarian, "Research advances in intelligent collision avoidance and adaptive cruise control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 3, pp. 132–153, 2003.
- [12] K. J. Aström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*, Princeton University Press, Princeton, NJ, USA, 2nd edition, 2009.
- [13] Z. Koruba, *Classical Mechanics: Applied Mechanics and Mechatronics*, Springer, New York, NY, USA, 2012.
- [14] T. Schwickart, H. Voos, J.-R. Hadji-Minaglou, M. Darouach, and A. Rosich, "Design and simulation of a real-time implementable energy-efficient model-predictive cruise controller for electric vehicles," *Journal of the Franklin Institute*, vol. 352, no. 2, pp. 603–625, 2015.
- [15] M. C. Martinez-Rodriguez, P. Brox, and I. Baturone, "Digital VLSI implementation of piecewise-affine controllers based on lattice approach," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 3, pp. 842–854, 2015.
- [16] Texas Instruments, "TMS320C6000 optimizing compiler v7.6," Tech. Rep., Texas Instruments, Dallas, Tex, USA, 2014, <http://www.ti.com/lit/ug/spru187u/spru187u.pdf>.
- [17] R. Chassaing, *Digital Signal Processing and Applications with the C6713 and C6416 DSK*, John Wiley & Sons, 1st edition, 2005.
- [18] D. Batten, S. Jinturkar, J. Glossner, M. Schulte, and P. D'Arcy, "A new approach to DSP intrinsic functions," in *Proceedings of the Hawaii International Conference on System Sciences (HICSS '00)*, p. 214, IEEE, Maui, Hawaii, USA, January 2000.




Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

