A motor controller for TORVEbot based on a System On Chip

Aristide Emanuele Casucci, Mohamed El Arayshi, Khaled Issa, Lorenzo Canese, Paolo Lucantonio, Mario Patetta
Department of Electronic Engineering
University of Rome "Tor Vergata", Rome, Italy 00133
Email: casucci.emyl@tiscali.it,elarayshi.mohamed@gmail.com,
khaledissa51@gmail.com, lorenzo@canese.it, paolo.lucantonio@outlook.it, mpatetta96@gmail.com

***Abstract*— This paper presents the implementation results of the motor controller for TORVEbot, a wheeled mobile robot designed to be used in cooperative teams/swarms. The controller has been implemented on the XILINX System on Chip Zynq.The Zynq SoC family integrates the software programmability of an ARM-based microprocessor with the hardware programmability of an FPGA, enabling key analytics and hardware acceleration while integrating CPU, DSP, ASSP, and mixed signal functionality on a single device.**

The controller has been implemented on the Programmable Logic and successively connected to the microprocessor using the AXI interface. Results in terms of area occupation have been presented. In order to verify the correct behavior of the controller, it has been integrated into TORVEbot.

I. Introduction

Swarm Robotics consists of the use of several autonomous robots able to cooperate with each other to accomplish several tasks. Swarm robotics is the study of how to coordinate large groups of relatively simple robots through the use of local rules.

Robot swarms find application in different fields as search and rescue, precision agriculture, military surveillance. In this scenario cooperating in the swarm must be able to take decisions autonomously considering its state and the external environment [1], [2]. In [3] the authors presented TORVEbot, a wheeled mobile robot designed to be used in cooperative teams/swarms. The main features of TORVEbot are:

- **Flexibility**: The robot it is able to operate in different environments and to accomplish different tasks
- **Experience sharing**: Robot units cooperating in the same swarm are able to communicate with each other. The capability to communicate with other robot units is very important. It can be used to share experience and speed-up the accomplishing of the task [4]

The flexibility is achieved providing the robot of reconfigurable electronic devices and of a common interface for all the sensors the robot needs. The use of a common interface for all the sensors, make possible to equip the robot of the appropriate sensors without the necessity to redesign the electronic board during updates or for different tasks. The experience sharing is obtained providing the robot of a Q-RTS module [4] and a wireless transceiver to communicate with the other robots in the swarm.

Q-learning Real-Time Swarm (Q-RTS) is an iteration-based reinforcement learning suitable for real-time systems. Reinforcement learning (RL) is an artificial intelligence technique to train an agent to perform a task by interacting with the environment. By trial-and-error, the agent adapts its behaviour through a rewarding mechanism, called reinforcement, which is a measure of its task-solving performance [5]

A crucial aspect of robots cooperating in a swarm is the capability to adapt itself to dynamic environments. Such capability is usually provided such robots also by Machine Learning (ML) algorithms. ML refers to the ability of computers to learning from data. In the last few years, ML gained an important role in several fields that as health, computer vision, and communications energy [7], [9], [11]. The availability of increasingly high computational power and the introduction of new technologies [12], [14], [13], [16], [17], [18], [19] have increased the interest in ML.

Thanks to ML robots are able to learn what to do when analyzing data coming from sensors and consequently, they are able to take decisions and be autonomous.

In this paper, we present the hardware implementation of TORVEbot motor control. The paper is organized as follow: In Sect. II the TORVEbot architecture is described, in Sect. III the motor control system is discussed, in Sect. IV the experimental results have been shown and finally, conclusions are discussed.

II. TORVEbot architecture

TORVEbot is a wheeled mobile robot designed to work in cooperative teams/swarms developed at the University of Rome Tor Vergata. The structure of the robot unit has been built using a 3D printer with PLA material and is equipped with the following devices:

- HC SR04 ultrasonic sensor with a range of 2 cm to 4 m and 1 cm resolution
- FS 90R servo motor

- A DILIGENT Pynq board that replaces the ARTY board used in the previous version of TORVEbot
- DHT11 digital temperature and humidity sensor
- GY-BMP280-3.3 pressure sensor
- NRF24L01+ wireless transceiver

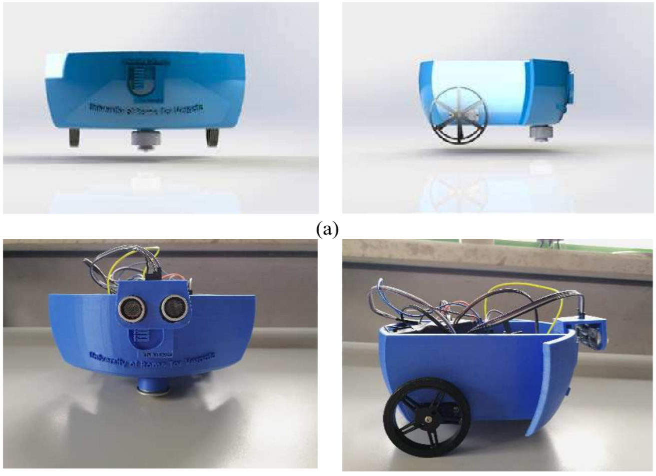
(a)
(b)

Fig. 1. The TORVEbot swarm robot unit: a) views of the CAD design; b) a built prototype

The use of reconfigurable electronic devices allows the possibility to reconfigure the robot for different tasks with fast easy operation with no necessary changes of the hardware and software as well as the corresponding mechanical design. The proposed TORVEbot robot is provided with XILINX PYNQ board (Fig. 2) equipped with a XILINX Zynq System On Chip (SOC) composed on an ARM Cortex A Microprocessor and Field Programmable Gate Array (FPGA). The main features of the board are:

- A 650MHz dual-core Cortex-A9 processor
- DDR3 memory controller with 8 DMA channels and 4 high performance AXI3 slave ports
- High-bandwidth peripheral controllers: 1G Ethernet, USB 2.0, SDIO
- Low-bandwidth peripheral controller: SPI, UART, CAN, I2C
- Programmable from JTAG, Quad-SPI flash, and microSD card
- Artix-7 family programmable logic

In addition to the reconfiguration capability, the FPGA offers the possibility to efficiently execute algorithms that are characterized by a considerable level of parallelism and the possibility to execute several algorithms in parallel. This solution allows the possibility to have a system composed of a Microprocessor for the general purpose operations, as for example, ADC and DAC interface and a Hardware Accelerator for all those algorithms requiring parallel computing. The use of mixed architecture composed of microprocessor and a hardware accelerator is a common solution in the literature [21], [20], [22], [23], [24], [25]. These approaches are very useful in the case of ML algorithms for three reasons, namely:

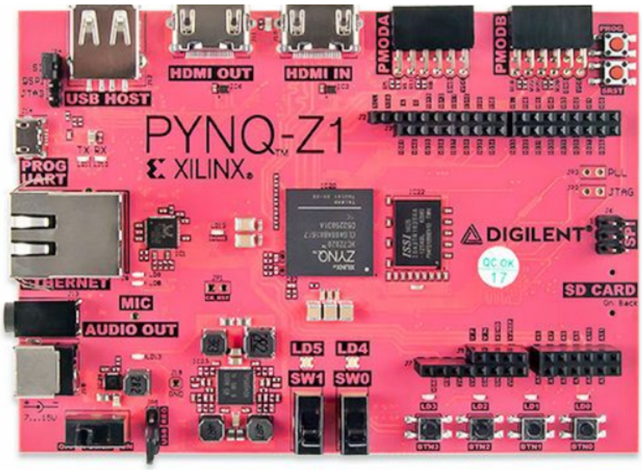

Fig. 2. The PYNQ board

1) FPGAs are able to execute matrix computation in a very efficient way. ML algorithms as for example CNN, SVM, SOM [8], [6], etc. are characterized by parallel operations as vector matrix multiplications.
2) FPGAs consents efficient implementation of Ensemble Machine Learning systems [10].
3) More ML algorithms can be executed in parallel being the FPGAs designed for parallel computing.

The Block Diagram of TORVEbot control unit is shown in Fig.3.

Such as control unit is composed of the ARM CORTEX A microprocessor and custom peripherals that are:

- A Machine Learning Accelerator.
- Two PWM controllers for the wheels
- Some Application Specific Sensors Interface ASSC. In our case, we have three interfaces for the HC SR04 , the DHT11 and the GY-BMP280-3.3.
- An interface for the wireless transceiver (Wireless Interface).

The microprocessor, communicate with the custom pheriperals using the AXI bus. AXI [30] is part of ARM AMBA, a family of micro controller buses introduced in 1996. The first version of AXI was first included in AMBA 3.0, released in 2003. AMBA 4.0, released in 2010, includes the second version of AXI, AXI4.

In the following, we focus on the motor control blocks (the two PWM controllers for the wheels)

III. Motor Control

The robot motion represents a crucial aspect in robot design and for this reason, this issue is very discussed in the literature [28], [29]. TORVEbot is provided of the motor FS90R (Fig. 4). The FS90R is a micro servo designed specifically for continuous rotation. This servo can work with both 5 V and

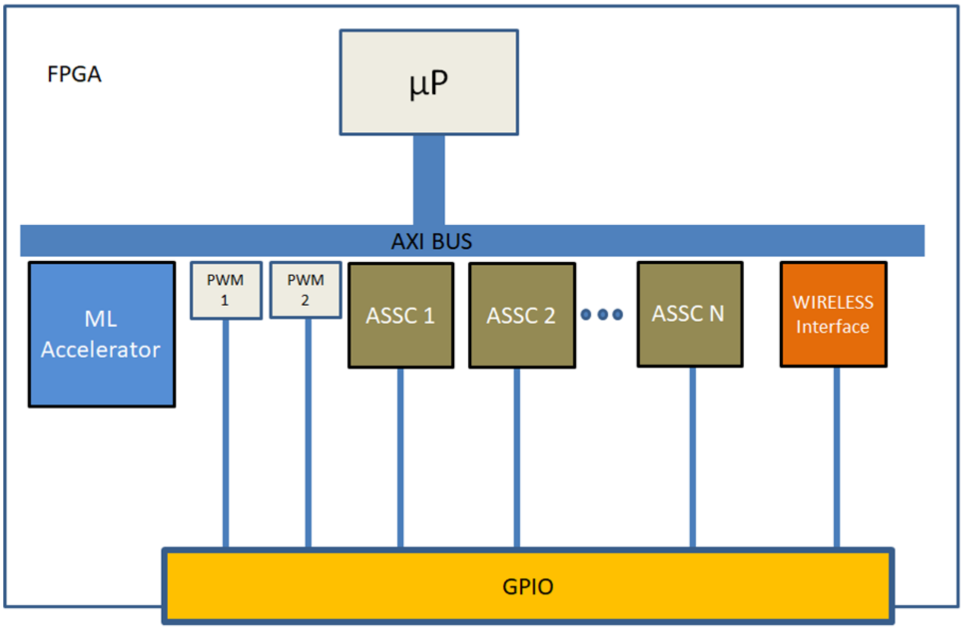

Fig. 3. TORVEbot controller block diagram

Fig. 4. The FS90R

3.3 V servo signals. Such a motor can be controlled using PWM signals having a control period of 20 ms.

The PWM controller block diagram is shown in Fig. 5 and it works in this way: The time is divided into temporal slots of 0.1 ms. At every 0.1 ms, the main counter increments its value from 0 to 200 (20 ms is the total duration of the control cycle of the motor). The output PWN waveform is obtained by a comparator that compares the counter value with a threshold value provided at the input of the controller. If the counter value is under the threshold the output is 1 otherwise it is 0. The threshold is provided to the controller by the ARM processor through the AXI lite interface.

The controller has been simulated in the MATLAB/SIMULINK environment (both floating point and fixed point simulation have been performed) and successively it was coded in VHDL at RTL level and implemented using the XILINX Vivado toolchain. The controller has been connected to the ARM processor (by the AXI LITE BUS) using the same toolchain.

IV. Experimental Results

After the VHDL description, synthesis and P&R have been performed using the XILINX VIVADO 2018 toolchain. Synthesis has been performed at 50 MHz (otiming constraint).

Implementation results are shown in Tab. I. Results refer to a couple of PWM controller, one for the right wheel and one for the left wheel. They have been used only 99 of the 53200 slice LUTs and only 231 of the 106400 slice registers. The few hardware resources used for the implementation of the motor controller allows the possibility to have lots of hardware resources available for the other block of the entire system shown in Fig 3. The total on-chip power is 1.4 W

TABLE I
Resource Utilization

Resource	Used	Available
SLICE LUTs	99	53200
SLICE REGISTERs	231	106400

The two implemented motor controllers have been connected to the microprocessor among the AXI-lite interface. Fig6 shows a screenshot of the entire project in the VIVADO tool suite. The Figure shows the main block of the systems that are the ARM microprocessor, the system reset, the AXI interface and the motor controller

After the integration of the motor controller with the ARM processor, the controller has been tested in TORVEbot. Exper-

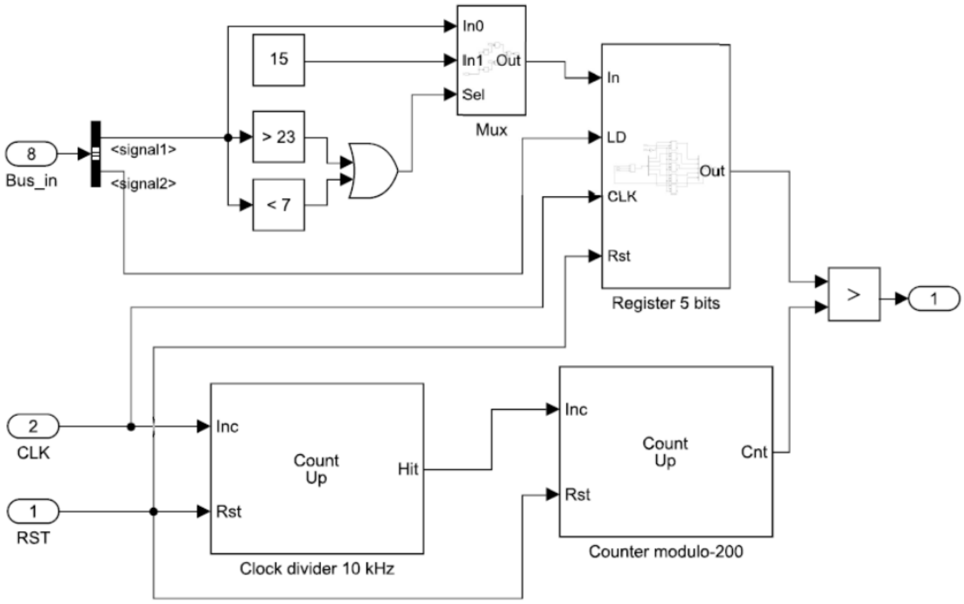

Fig. 5. Controller Simulink Block Diagram

iments have been performed programming the robot to execute some specific trajectories. The software for the controller has been developed in C language in the XILINX VIVADO SDK environment. When the robot has to perform some specific movements, the microprocessor controls the custom peripheral implemented in the programmable logic using the AXI-LITE interface.

V. Conclusion

In this paper we present the implementation results of the motor controller for the TORVEbot, The controller has been implemented on the XILINX Zynq, composed of an ARM microcontroller and a Field Programmable Gate Array. Implementation results show a very limited resources utilization (only 99 of the 53200 slice LUTs and only 231 of the 106400 slice registers) and a total on-chip power about 1.4 W.

Other custom peripherals for the TORVEbot are under development. In particular, we are working on the wireless interface that will provide the robot of the NRF24L01+ digital transceiver. Such transceiver will be used by TORVEbot to communicate with the other robots in the swarm and with a data collection control unit. This unit will permit the remote monitoring of the robot.

VI. Acknowledgments

The authors would like to thank Xilinx Inc, for providing FPGA hardware and software tools by Xilinx University Program.

References

[1] Seo, S., Yang, H., and Sim, K. ”Behavior learning and evolution of swarm robot system for cooperative behavior” (2009). IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM, 673-678. doi:10.1109/AIM.2009.5229933.

[2] Iaki Navarro and Fernando Mata ”An Introduction to Swarm Robotics” ISRN Robotics Volume 2013, Article ID 608164, 10 pages

[3] Di Nunzio L. Cardarilli G.C., Ceccarelli M. Fazzolari R. ”Design and requirements for a mobile robot for team cooperation” 6th IFToMM International Symposium on Robotics and Mechatronics (ISRM 2019) ARTICLE IN PRESS

[4] Matta, M., Cardarilli, G.C., Di Nunzio, L., Fazzolari, R., Giardino, D., Re, M., Silvestri, F., Span, S. ”Q-RTS: A real-time swarm intelligence based on multi-agent Q-learning” (2019) Electronics Letters, 55 (10), pp. 589-591.

[5] Sutton, R.S., and Barto, A.G.: ”Reinforcement learning: an introduction” (MIT Press, Cambridge, MA, USA, 2018)

[6] Cardarilli, G.C., Di Nunzio, L., Fazzolari, R., Re, M., Spano, S. ”AW-SOM, an Algorithm for High-speed Learning in Hardware Self-Organizing Maps” (2019) IEEE Transactions on Circuits and Systems II: Express Briefs.

[7] Capizzi, G., and G. Tina. ”Long-term operation optimization of integrated generation systems by fuzzy logic-based management.” Energy 32.7 (2007): 1047-1054

[8] Giardino, D., Matta, M., Silvestri, F., Span, S., Trobiani, V. ”FPGA implementation of hand-written number recognition based on CNN” (2019) International Journal on Advanced Science, Engineering and Information Technology, 9 (1), pp. 167-171.

[9] Capizzi, Giacomo, et al. ”Cascade feed forward neural network-based model for air pollutants evaluation of single monitoring stations in urban areas.” International Journal of Electronics and Telecommunications 61.4 (2015): 327-332.

[10] Cardarilli, G.C., Di Nunzio, L., Fazzolari, R., Giardino, D., Matta, M., Re, M., Silvestri, F.S. ”Span Efficient Ensemble Machine Learning implementation on FPGA using Partial Reconfiguration” (2019) Lecture Notes in Electrical Engineering Springer, 2019

[11] Beritelli, Francesco, et al. ”Automatic heart activity diagnosis based on Gram polynomials and probabilistic neural networks.” Biomedical engineering letters 8.1 (2018): 77-85.

[12] Susi, G., Toro, L.A., Canuet, L., Lpez, M.E., Maest, F., Mirasso, C.R., Pereda, E. ”A neuro-inspired system for online learning and recognition of parallel spike trains, based on spike latency, and heterosynaptic STDP” (2018) Frontiers in Neuroscience, 12, art. no. 780, .

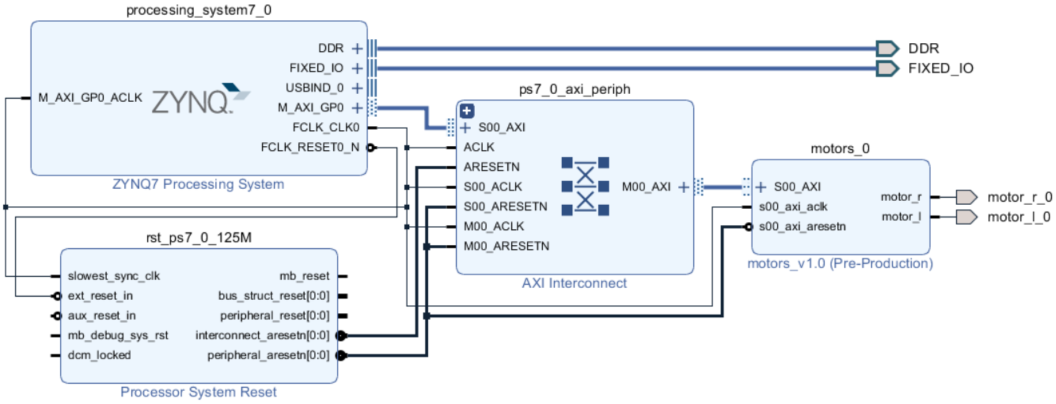

Fig. 6. VIVADO block diagram

[13] Salerno, M., Susi, G., Cristini, A. "Accurate latency characterization for very large asynchronous spiking neural networks" (2011) BIOINFORMATICS 2011 - Proceedings of the International Conference on Bioinformatics Models, Methods and Algorithms, pp. 116-124.

[14] Yassin, I.M., Abdul Khalid, M.F., Herman, S.H., Pasya, I., Ab Wahab, N., Awang, Z. Multi-Layer Perceptron (MLP)-based Nonlinear Auto-Regressive with Exogenous Inputs (NARX) stock forecasting model (2017) International Journal on Advanced Science, Engineering and Information Technology, 7 (3), pp. 1098-1103.

[15] Nasution, F.B.B., Bazin, N.E.N., Daliyusmanto, Zulfikar, A. Big data's tools for internet data analytics: Modelling of system dynamics (2017) International Journal on Advanced Science, Engineering and Information Technology, 7 (3), pp. 745-753.

[16] Susi, G., Cristini, A., Salerno, M. "Path multimodality in a feedforward snn module, using lif with latency model" (2016) Neural Network World, 26 (4), pp. 363-376.

[17] Coco, S., Laudani, A., Fulginei, F. R., Salvini, A., "Bacterial chemotaxis shape optimization of electromagnetic devices" (2014), Inverse Problems in Science and Engineering, 22 (6), pp. 910-923, Taylor & Francis.

[18] Bonanno, F., Capizzi, G., Coco, S., Laudani, A., Lo Sciuto, G., "A coupled design optimization methodology for Li-ion batteries in electric vehicle applications based on FEM and neural networks", (2014) IEEE International Symposium on Power Electronics, Electrical Drives, Automation and Motion, pp. 146-153.

[19] Bonanno, F., Capizzi, G., Coco, S., Napoli, C., Laudani, A., Sciuto, G. L., "Optimal thicknesses determination in a multilayer structure to improve the SPP efficiency for photovoltaic devices by an hybrid FEMcascade neural network based approach", (2014) IEEE International Symposium on Power Electronics, Electrical Drives, Automation and Motion, pp. 355-362.

[20] Hauck, S., Fry, T.W., Hosler, M.M., Kao, J.P. The Chimaera Reconfigurable Functional Unit (2004) IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 12 (2), pp. 206-217.

[21] Cardarilli, G.C., Di Nunzio, L., Fazzolari, R., Re, M. "Fine-grain reconfigurable functional unit for embedded processors" (2011) Conference Record - Asilomar Conference on Signals, Systems and Computers, art. no. 6190048, pp. 488-492.

[22] Schmit, H., Whelihan, D., Tsai, A., Moe, M., Levine, B., Taylor, R.R. "PipeRench: A virtualized programmable datapath in 0.18 Micron technology" (2002) Proceedings of the Custom Integrated Circuits Conference, pp. 63-66.

[23] Cardarilli, G.C., Di Nunzio, L., Fazzolari, R., Re, M. Cardarilli, G.C., Di Nunzio, L., Fazzolari, R., Re, M. "TDES cryptography algorithm acceleration using a reconfigurable functional unit" (2015) 2014 21st IEEE International Conference on Electronics, Circuits and Systems, ICECS 2014, art. no. 7050011, pp. 419-422.

[24] Razdan, Rahul, Brace, Karl, Smith, Michael D. "PRISC software acceleration techniques" (1994) Proceedings - IEEE International Conference on Computer Design: VLSI in Computers and Processors, pp. 145-149

[25] Ammendola, Roberto, and Pierpaolo Loreti. "Design and Evaluation of a Scalable Engine for 3D-FFT Computation in an FPGA Cluster." International Journal on Advanced Science, Engineering and Information Technology 9.2 (2019): 677-684

[26] Bracciale, L., Catini, A., Gentile, G., Loreti, P. "Delay tolerant wireless sensor network for animal monitoring: The Pink Iguana case" (2017) Lecture Notes in Electrical Engineering, 429, pp. 18-26.

[27] Loreti, P., Catini, A., De Luca, M., Bracciale, L., Gentile, G. and Di Natale, C., 2019. The Design of an Energy Harvesting Wireless Sensor Node for Tracking Pink Iguanas. Sensors, 19(5), p.985.

[28] Dewangga, S.A., Tjandrasa, H., Herumurti, D. "Robot motion control using the emotiv EPOC EEG system" (2018) Bulletin of Electrical Engineering and Informatics, 7 (2), pp. 279-285.

[29] Bracciale, L., Loreti, P., Detti, A., Paolillo, R., Melazzi, N. B. (2019). Lightweight Named Object: an ICN-based Abstraction for IoT Device Programming and Management. IEEE Internet of Things Journal.

[30] XILINX, IP documentation: AXI Reference guide, (2019).