AutoRIO: An Indoor Testbed for Developing Autonomous Vehicles

Mohammad Loni, Masoud Daneshtalab, Mikael Sjödin
School of Innovation, Design and Engineering, Mälardalen University
Västerås, Sweden
{mohammad.loni, masoud.daneshtalab, mikael.sjodin}@mdh.se

Fadouao Hamouachy, Clémentine Casarrubios
Department of Microelectronics and automation, University of Montpellier
Montpellier, France
{fadoua.hamouachy, clementine.casarrubios}@etu.umontpellier.fr

***Abstract*— Autonomous vehicles have a great influence on our life. These vehicles are more convenient, more energy efficient providing higher safety level and cheaper driving solutions. In addition, decreasing the generation of CO_2, and the risk vehicular accidents are other benefits of autonomous vehicles. However, leveraging a full autonomous system is challenging and the proposed solutions are newfound. Providing a testbed for evaluating new algorithms is beneficial for researchers and hardware developers to verify the real impact of their solutions. The existence of testing environment is a low-cost infrastructure leading to increase the time-to-market of novel ideas. In this paper, we propose AutoRIO, a cutting-edge indoor testbed for developing autonomous vehicles.**

Keywords—Autonomous Vehicles; Embedded Systems; Near-Sensor Processing

I. INTRODUCTION

Autonomous vehicles are becoming prominent day-by-day. The main advantages of autonomous vehicles are extremely diminishing mobility costs, reducing traffic pressure, reducing the rate of car accidents and fatalities, increasing road utilization, decreasing the negative effect of CO_2 on the atmosphere, and efficient consuming of fuels.

To investigate and develop technologies for autonomous vehicles, we propose an indoor testbed, named AutoRIO, a realtime test environment to study autonomous vehicles. A comprehensive testbed requires considering factors of real environment which actively effect on operators manage processing hardware, navigation, control, and vehicle moving system. AutoRIO allows researchers to investigate their ideas by verifying the real impact of their algorithms associated with high-level tasks. Plus, AutoRIO provides features for testing low-level protocols and high-level path planning algorithms on autonomous vehicular platform. AutoRIO is also being used to implement modern deep learning algorithm on embedding system. Different situation could be considered to simulate a more realistic environment (for instance, vehicle failures, refueling, and foggy weather). Moreover, supporting modularity is essential for AutoRIO due to providing rapid prototyping of new vehicle configurations and algorithms without requiring a redesign of the vehicle hardware/software. In this article we first explain the needed components of AutoRIO, then we propose a general architecture for autonomous vehicles. Figure 1 illustrates the hardware frame of AutoRIO. AutoRIO consists of three motors including DC motor, servo motor, and stepper motor for moving vehicle forward/backward, rotating the wheels, and rotating laser sensors, respectively. The rest of the paper describes different essential components and the processing pipeline of AutoRIO including receiving data from sensors, raw data preprocessing, sensor data fusion, heavyweight processing unit for object and motion recognition, and path planning and controlling the vehicle.

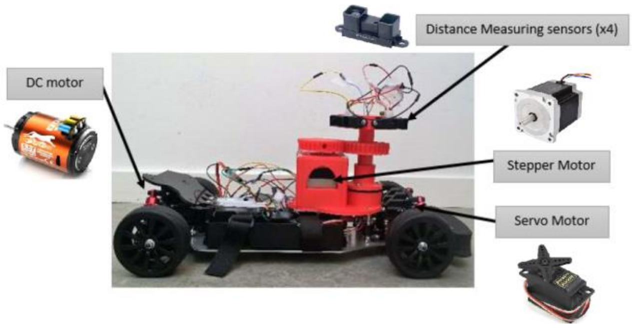

Figure 1. The hardware frame of AutoRIO

In nutshell, our main contributions in AutoRIO are threefold:

- Proposing a modular and scalable processing architecture for autonomous vehicles.
- Developing low-level interfaces in LABVIEW to communicate with various sensors and actuators.
- Integrating a deep learning accelerator, named ADONN [1], in the processing pipeline for object recognition.

II. BACKGROUND

AutoRIO needs to continuously monitor vehicle sensors since they are ears and eyes of the controlling system and AutoRIO operations and safety depend on the precision of the sensors. The sensors feed the vehicle control systems with information of the current state of the vehicle's surroundings. In this section, we briefly describe different common sensing ways used in modern autonomous vehicles. In addition, the specification of each sensor and the main processing unit of the AutoRIO will be explained.

A. *Vision Sensors*

Video images provide the most informative data for recognizing surrounding environment due to representing detailed outside environment of the vehicle. Plus, image sensors can present heterogeneous data including color, shape, and distance information. Generally, video cameras are divided into two categories including stereo-vision and mono cameras. AutoRIO supports both types of cameras. However, our main intention is to utilize stereo-vision based sensing modules due to providing the distance between the vehicle and detected objects. However, vision sensors have some limitations in which they are affected by the environmental degradation like dust, dirt, or high-moisture situations.

GIMME2 [2] is an embedded power efficient stereo-vision system based on the Xilinx Zynq 7020 System-on-Chip (SoC) FPGA that provides megapixel pictures and supports a high-speed image processing pipeline. Figure 2 illustrates the overview of GIMME2 architecture. Providing high resolution images, low power consumption, and standardized interface are the main properties of GIMME2, allowing vision-based sensing feasible. GIMME2 is equipped with two Aptina MT9J003 image sensors which are CMOS sensors with 15fps@10MP (3856 x 2764 bayer pattern array) or 60fps@1080p, with 12-bit color resolution. Each sensor produces 2.8 Gb/s data over four 240 MHz LVDS-lanes. A simple Zero-mean Sum of Absolute Differences (ZSAD) stereo-matching is utilized for full HD real-time depth estimation [3]. Frame rate could be varied for different applications, e.g. it can provide up to 240 frame/second for images with 640x400 pixels.

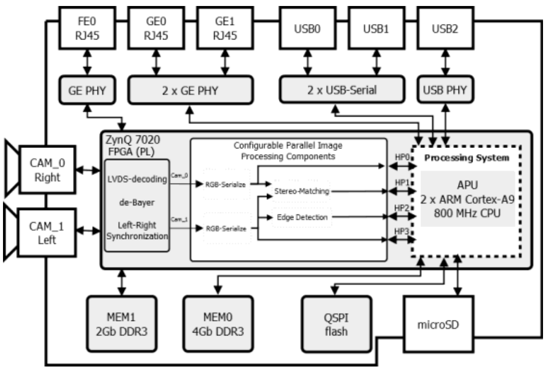

Figure 2. The overview of GIMME2 architecture

B. *LiDAR Sensors*

Light Detection and Ranging (LiDAR) is a laser-based system for moving object detection, distance calculation. It contains a laser transmitter and a highly sensitive receiver. In addition, LiDAR provides three-dimensional images of the detected objects since because of a complex mechanical mirror system with a 360° all-round visibility that capture spatial images of objects. However, LiDAR sensors are highly expensive that leading to be inadequate for cost effective solutions. AutoRIO is equipped with four rotating laser sensors to cover360° all-round visible environment, in order to simulate a semi LiDAR sensor (Figure 1).

C. *RADAR Sensors*

RAdio Detecting and Ranging (RADAR) is a system to detect objects and determine their range, angle, and/or velocity of objects. RADAR is based on sending out electromagnetic waves and analyzing received signals. RADAR systems provide highly reliable and low-cost sensors for safe autonomous vehicle operations. However, RADAR is not a power efficient solution. Current RADARs operate in the 77 GHz due to available wide bandwidth which improves accuracy and object resolution, the antenna can be small, and atmospheric absorption limits interference with other systems. AutoRIO does not support RADAR yet. Plus, a RADAR unlike LiDAR presents the velocity and bearing information of the object. Laser data is also more computationally intensive because a laser sends lots of data about each individual laser point of range data.

D. *Ultrasonic Sensors*

An ultrasonic sensor uses a single ultrasonic element for both emission and reception. In a reflective model ultrasonic sensor, a single oscillator emits and receives ultrasonic waves alternately. These sensors are not affected by color or transparency of objects, can be used in dark environments and are not highly affected by unclear environment. However, these sensors are not designed to be used in underwater and have a restricted detection range around maximum 10 meters. Today, ultrasonic sensors are used in auto-parking system to detect rear objects of the vehicle.

E. *Wireless Communication Devices*

Vehicle-to-vehicle communication is a modern solution which lets cars broadcast their position, speed, brake status, and other data to other close vehicles. The other cars can use that information to better picture what's unfolding around them, revealing catastrophic occurrences. In addition, the range of sensors integrated on each car is limited to a few car lengths, and they cannot see past the nearest obstruction. Therefore providing a wireless connection for vehicle-to-vehicle communication is essential for the AutoRIO life-saving goals.

Table 1. NI myRIO specifications

Reconfigurable Processor	XILINX Zynq-7010
Realtime Processor	ARM® Cortex™-A9 dual-core
Processor Frequency	667MHz
Non-volatile Memory	256 MB
DDR3 Bus Width	16 bit
WLAN	2.4 GHz, 802.11
Other Supported Features	UART, Accelerometer, Analog/digital I/O, USB

F. *Processing Board*

We used National Instruments (NI) myRIO-1900 embedded processing board in this project for managing near-sensor processes and sensor communication tasks. NI myRIO totally supports LABVIW [4] which is a system-design platform and development environment for a visual programming language. The main characteristics of myRIO board is shown in Table 1.

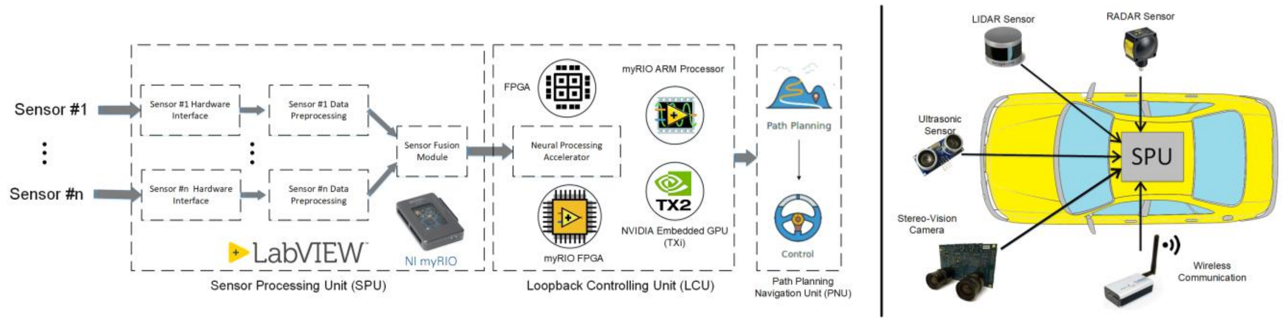

Figure 3. An overview of the proposed AutoRIO system architecture

III. System Overviw

Figure 3 illustrates the proposed processing architecture of AutoRIO that is composed of three components including Sensor Processing Unit (SPU), Loopback Controlling Unit (LCU), and Path-planning and Navigation Unit (PNU), respectively. The main aim of this architecture is to satisfy the requirements of a realistic testbed including high modularity, ease of development, and easy debugging. Moreover, the architecture should support the necessities of an autonomous vehicle such as receiving data from different sensor roots, real-time processing of data sensors and providing comprehensive information about the surrounding happenings. In this section we explain each part separately.

A. Sensor Processing Unit (SPU)

SPU is mainly responsible for near-sensors signal processing and relaxing the high-performance processor from managing I/O interrupts and low-level communications. In addition, the drivers of vehicle motors including stepper motors, servo motor and DC motor are implemented in SPU. Since the AutoRIO sensors are inherently heterogeneous, we need to implement different hardware interfaces for each communication protocol. Data preprocessing module is the next module which oversees data compression, dropping redundant data and converting raw analog data to adequate format. For example, the raw received data from GIMME2 image lenses is analog on four LVDS-lanes and suitable format for the input images is RGB format, thus converting the analog image signals to RGB images is the responsibility of data preprocessing module.

Why do we need several sensors? We need to find the answer of the question beforehand for explaining the necessity of the sensor fusion module. This is because, each sensor provides different types of information about the outside environment with different accuracies especially in different weather conditions, e.g. a LiDAR based sensor can provide good resolution about the position but suffers for accuracy in poor weather, but, the spatial resolution of a RADAR sensor is relatively poor compared to laser but provides better accuracy in poor weather.

The main idea of sensor fusion is to take the inputs of different sensors and sensor types and use the merged information to perceive the environment more accurately. Plus, a certain level of redundancy will be required when a sensor fails. For example, imagine our RADAR tells us that the vehicle ahead of us is moving at 12m/s, but our LiDAR tells us that it is moving at 10m/s. With sensor fusion techniques we can provide a pretty good guess about what speed the vehicle ahead of us is moving. A particular mathematical algorithm called a Kalman filter [5] that is often used to combine this merge. Kalman filters rely on probability and a "measurement-update" cycle to put together a probabilistic understanding of the world. Kalman filter is fully supported in AutoRIO and could be implemented using LABVIEW. Finally, the sensor fusion module is centralized in our design.

Besides precise implementation of low-level protocols (hardware side), the realtime-ness consideration of each module is essential since we have a hard deadline to finish each task in autonomous vehicles as a mission critical application. The NI myRIO processor provides a realtime solution, but it cannot guarantee to meet any timing deadline for any desired design. FPGAs could be beneficial to tackle this challenge since FPGAs support implementation of any type of parallelism and highly efficient designs. The NI totally support FPGA implementations using LABVIEW on myRIO board. The AutoRIO did not use FPGA for implementing SPU since the realtime processor could afford to meet required timing boundaries. As mentioned earlier, the moving drivers of AutoRIO is implemented as a part of SPU. Figure 4 shows LABVIEW implementation of each motor. Due the sake of brevity, we just illustrate the implementation of motors.

B. Loopback Controlling Unit (LCU)

Compare to different sensing ways such as LiDAR, RADAR, and cameras, stereo-vision cameras are more attractive because of providing heterogeneous information simultaneously. AutoRIO can benefit from stereo-vision systems to move on the right path, detect dynamic objects such as pedestrians and other vehicles, and estimating the distance between the vehicle and recognized objects. Nowadays, deep Convolutional Neural Networks (CNNs) are popular processing models in visual recognition, decision making and prediction algorithms because of providing higher accuracy and more flexibility compared to old-fashioned solutions. However, CNNs are complex and containing up to billions of operations which makes their realtime implementation challenging especially for embedded realtime systems.

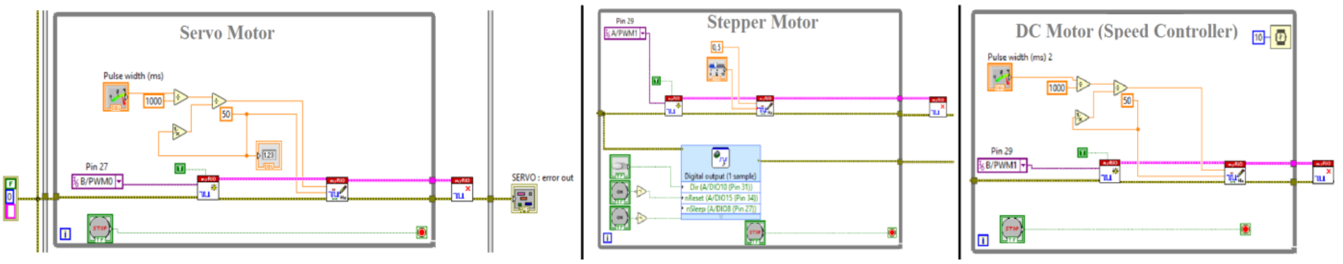

Figure 4. LABVIEW implementation of AutoRIO moving interfaces

Table 2. Comparing the inference time for different optimized networks designed by ADONN framework

Network Architecture	Accuracy	#Trainable Parameters	GPU (ms)	CPU (ms)	ARM (ms)	FPGA (ms)
CNN-3	86%	0.14×10^6	2.82	14.9	81.3	14.66
CNN-2	91.3%	0.49×10^6	4.82	41.47	175.2	57.73
CNN-1	93.1%	1×10^6	7.51	76.88	292.58	111.72

In a forthcoming accepted publication [1], we propose an acceleration framework for CNNs which tries to optimize the network architecture since the choice of the architecture strongly effects on the inference time, memory usage, hardware footprint, the accuracy level, and the prediction quality of CNNs. The proposed framework is evolutionary-based accelerator which is also compatible with GIMME2. Our proposed solution tries to design a near-optimal CNN for image classification algorithms in term of network size while guaranteeing acceptable level of accuracy by employing a multi-objective Genetic Programming (GP) based method. Our framework then maps the generated network to a multi/many core System-on-Chip and/or FPGA. The FPGA of myRIO is not suitable for implementing CNNs since we cannot always put elephant in fridge. In the other words, the Zynq-7010 is not a powerful reconfigurable processor and is used inherently for light designs. Thus, we must use an alternative processor which could be myRIO realtime processor or other embedded platforms such as NVIDIA Tegra TX2, an ARM processor or another FPGA. Depending on the complexity of the CNN and response time limitations, our framework generates a robust CNN architecture. Table 2 presents the experimental results for a popular standard object recognition dataset, CIFAR-10 [6], on four prevalent hardware platforms including XILINX UltraScale plus FPGA, NVIDIA Tesla M60 GPU, Intel Core i7-7820, and ARM Cortex-A15. The total execution time is considered as the evaluation metric since mission critical applications are mainly latency-oriented. CIFAR-10 is a complex colorful benchmark dataset of natural images, each with 32x32 pixels which is mainly used for object recognition. As inferred from the values of the Table, the image processing rate for a complex dataset is 12 frame/second on the ARM processor as the slowest platform.

C. Path Planing and Navigation Unit (PNU)

The PNU is the final stage in the proposed architecture. The raw data received from sensors after preprocessing (by SPU) and detecting objects and recognizing motions by applying deep learning algorithms (by LCU) will become knowledge. The main intention of PNU is to leverage the extracted knowledge from the environment to make the best decisions and finding the safest, most convenient, and most economically navigation for the routes from point A to point B. Therefore, the system should generate corresponding controlling commands for the wheels, brake system, the motor engine etc. There have been proposed different path planning algorithms. For ease of implementation, we just consider a situation when the vehicle detects a close obstacle such as a pedestrian and should immediately stop the car. Therefore, we need to generate stop signals and send them to the DC motor.

IV. Conclusion

This paper proposes AutoRIO, a realtime testbed for autonomous vehicles. AutoRIO could be beneficial in research project for developing and testing novel ideas. High modularity, ease of development, and supporting modern sensors are the key advantages of AutoRIO.

References

[1] M. Loni, M. Daneshtalab, and M. Sjödin, ADONN: Adaptive Design of Optimized Deep Neural Networks for Embedded Systems, Euromicro Conference on Digital System Design (DSD), Prague, Czech, 2018.

[2] C. Ahlberg, F. Ekstrand, M. Ekstrom, G. Spampinato, and L. Asplund, GIMME2 - An embedded system for stereo vision and processing of megapixel images with FPGA-acceleration, in 2015 International Conference on ReConFigurable Computing and FPGAs, ReConFig 2015, 2016.

[3] P. Greisen, S. Heinzle, M. Gross, and A. Burg, An FPGA-based processing pipeline for high-definition stereo video, EURASIP Journal on Image and Video Processing, vol. 2011, no. 1, 2011. [Online]. Available: http://dx.doi.org/10.1186/1687-5281-2011-18

[4] Johnson, G.W., 1997. LabVIEW graphical programming. Tata McGraw-Hill Education.

[5] Sasiadek, J.Z. and Wang, Q., 1999. Sensor fusion based on fuzzy Kalman filtering for autonomous robot vehicle. In Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on (Vol. 4, pp. 2970-2975). IEEE.

[6] A. Krizhevsky and G. Hinton. Cifar-10 dataset. https://www.cs.toronto.edu/kriz/cifar.htm