ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/341760152

Real-Time Data Processing in Autonomous Vehicles Based on Distributed Architecture: A Case Study

Chapter · January 2020

DOI: 10.1007/978-981-15-5270-0_13

CITATIONS
0

READS
830

5 authors, including:

Yassine el Hafid
Hassania School of Public Works
6 PUBLICATIONS **4** CITATIONS

SEE PROFILE

El Rharras Abdessamad
Hassania School of Public Works
26 PUBLICATIONS **192** CITATIONS

SEE PROFILE

Abdellah Chehri
Royal Military College of Canada
211 PUBLICATIONS **1,659** CITATIONS

SEE PROFILE

Saadane Rachid
Hassania School of Public Works
189 PUBLICATIONS **1,264** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Optimized Edge IOT Solutions for Sustainable Smart Cities View project

Project Wireless Sensor Networks View project

All content following this page was uploaded by Saadane Rachid on 13 June 2020.

The user has requested enhancement of the downloaded file.

Real Time Data Processing in Autonomous Vehicles Based on Distributed Architecture: a Case Study

Yassine El Hafid [1], Abdessamad El Rharras [1], Rachid Saadane [1], Abdellah Chehri [2], Wahbi Mohammed [1]

[1] Laboratory Engineering system, SIRC/LAGeS-EHTP Hassania School of Public Works Casblanca, Morocco
el_hafid_yassine@ehtp.ac.ma, el_hafid_yassine@yahoo.fr, wahbi.mo@gmail.com, rachid.saadane@gmail.com
[2] Department of Applied Sciences University of Quebec in Chicoutimi Chicoutimi, QC, Canada, G7H 2B1
achehri@uqac.ca

Abstract. This work aims to evaluate the real-time processing system in the context of an autonomous vehicle with limited hardware and software capabilities. We elaborate algorithm implemented in 1/10 scale electric car using a line scan camera, speed sensors, and embedded electronic control system. The vehicle navigates in an arbitrary one-lane circuit using an edge detection algorithm. The challenge was to make a complete one loop of the arbitrary circuit in the shortest time with various lighting conditions. The experiments show that several issues were revealed in each step of data sensors processing and need a robust algorithm to handle exceptions caused by multiple disturbances. Furthermore, we paid particular attention to time constraints in embedded processor calculation and actuators response time to achieve reliable critical software control algorithms.

Keywords—Autonomous Vehicle, Real time data processing, embedded system.

1 Introduction

In the past few years, the new era of the autonomous vehicle has rinsing according to the development of new technologies like powerful embedded processor, Lidar, satellite positioning system, mems sensors and lightweight, high power density batteries. However, developing reliable, safe autonomous software need a strong understanding of the environment where the vehicle navigates, especially in the development of the next smart car transportation system where human life is the main concern. The primary issue is the unexpected events that can disturb the behavior of the autonomous.

The development of software capable of handling such complexed environments is possible with using novel methods like ANN [1], SVM [2], Deep learning [3], and others. Those methods have an expensive cost time processing and power consumption, especially in an embedded control system [4].

In this article, we expose our studies about a small autonomous vehicle in the scope of our participation in international smart car competition organized by the NXP Semiconductor company. The goal to achieve is to make one loop of the arbitrary circuit with the maximum speed and without leaving the circuit. Several challenges were placed in the road like hills, chicanes, crosses, and bumps. Furthermore, the lighting conditions were arbitrary by using a different kind of technology, controlling system, intensity, and some time with the parasitic source of light from the external environment.

The next part of this paper is organized as follows. Section II, presents car kit with different sensors, actuators, and electronic control boards. Section III exposes different problematic and issues encountered in the laboratory test. Section IV describes the proposed embedded software solution and his implementation with time constraints. And finally, a general conclusion is drawn in Section V.

2 Car system architecture

As we introduced in the previous paragraph, a normalized competition car kit was used for the principle of equal opportunities and fairness. However, the contestant can use additional sensors and actuators with limitations according to competition rules. Fig 1 presents the architecture of our test car used in competition.

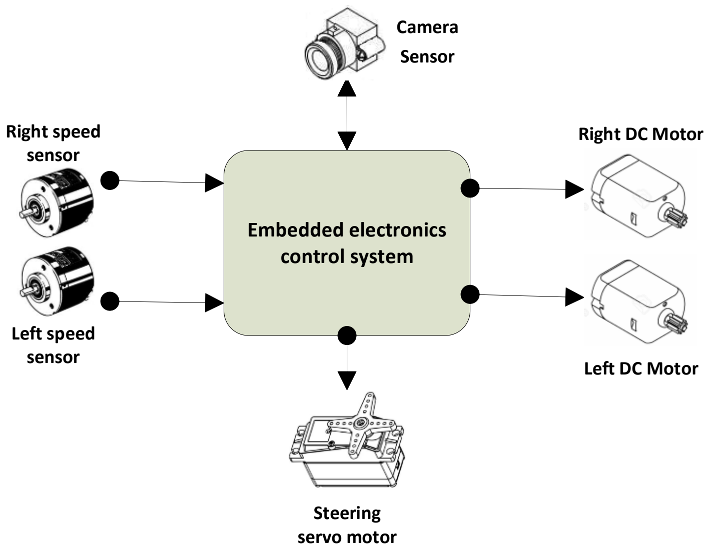

Fig 1. Car system architecture.

2.1 Car chassis and actuators

The car chassis Fig 2 is one-piece glass fiber chassis with tow DC motors controlling the rear wheels. For front steering wheels, we use a digital RC Servo motor having a torque of 11kg/cm, which connected by a metallic arm. The overall weight of the car is 750g (including the batteries).

Fig 2. Alamak Car model for NXP CUP competition.

2.2 Line scan Camera TSL1401

The primary sensor used was the line scan camera TSL1401 Fig 3. It's 128 pixels photodiodes array, which can be controlled by the embedded processor through tow signals CLK and SI (signal integration). The reading value is performed with the ADC module of the Microcontroller.

The camera was placed on Aluminium road at 30 cm from the floor. Furthermore, the reading point was à 60cm from the center of the front wheels. The lens camera angle is 120° with an infrared filter.

Fig 3: TSL1401 camera module.

However, the camera has an imperfection in detection. Indeed, we observe that when the camera reads on the white side of the track, especially in the cross. Different level of whites is detected like an arc, as shown in Fig 4.

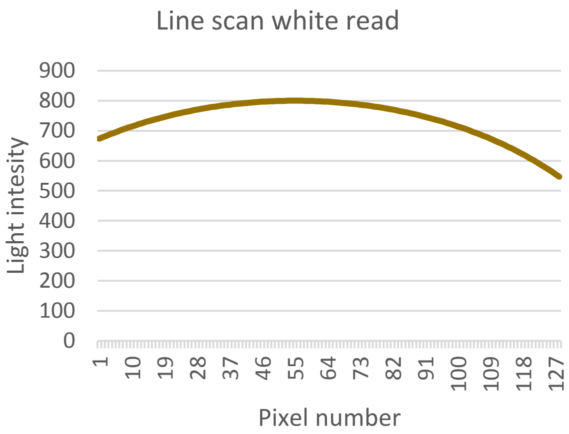

Fig 4. Line scan camera reading the white part of the track.

Another camera imperfection is the value of the edge on high light intensity. The pixels from 0 to 18 and from 109 to 127 no perform correctly as the pixels in the center of the sensor, so we ignore them in measurement.

2.3 Optical encoder

As we need to make a speed controller, we implement a system for measuring the speed. We adopted the Encoder E6A2-C (Incremental 25-mm-dia. Rotary Encoder) Fig. 5 for providing an electrical pulse signal corresponding to the rotational position and direction of the wheel. In every period T (T=10ms), we calculate the number of front edges in the entry of one of the pins of the microcontroller. We deduce the angular speed by dividing this number by the period T. To have the linear velocity, we multiply the angular velocity by the radius of the wheel and the transmission ratio of the gear system.

For our system, we have:

· Transmission ratio of the gear system = 1
· Output phases: Phases A and B
· Power supply voltage: 5 to 12 VDC
· Phase difference between outputs: Phase difference between phases A and B: 90°±45°.

Fig 5. Incremental 25-mm-dia. Rotary Encoder E6A2-C.

2.4 Electronic boards control

We use 3-layers stacked boards composed by KL25Z Microcontroller clocked at 48MHz with128KB Flash program data memory and 16KB SRAM as the mainboard. power board based on the BTN7960B motor circuit drivers and interface board for the switch, sensors, display, and Bluetooth communication.

Fig 6: KL25Z, power and interface boards.

2.5 Battery

According to rules of NXP CUP competition, only NiMH and Li-ion batteries are allowed for competition with the constraint of 7.2v as maximum nominal voltage rate and 300MAh as the maximum capacity. We choose a specific Li-ion battery among various type for the lightweight and higher discharge current needed in car acceleration. The NiMH batteries are two to three times heavier than Li-ION ones. Below you will find the characteristics of batteries used in our car model:

- Voltage: Nominal 3.7V,
- Maximum Voltage: 4.2V,
- Capacity: 2500mAh,
- Discharge max: 35A (pulse), Continuous discharge max: 20A,
- Battery LiNiCoAlO2 (NCA),
- Dimensions : L : 65 mm +/- 0.5mm / D : 18 mm +/- 0.5mm,

- Weight: 44g by a cell.

3 Car test environments and disturbances sources

The most challenging aspect of the autonomous vehicle is to be able to handle all fault conditions. So, the developer needs to understand all the possible sources of disturbances carefully. In general, several safety tests must be performed in different situations to trigger fault states and debug embedded software.

3.1 Track test

For the laboratory test as the NXP CUP competition, we use the same track specification according to rules as shown in Fig 7. the track was made with 3mm white plastic sheet with black edges.

The surface was rough enough to keep ground contact of the wheels. However, at high speed the car slipped and start drift causing error speed measurements. Another factor like dust, can increase this behaviour, so we need to keep car wheels and track clean. On another note, we observe in some cases that a reflective light can be detected by the camera causing the misreading black edges as white.

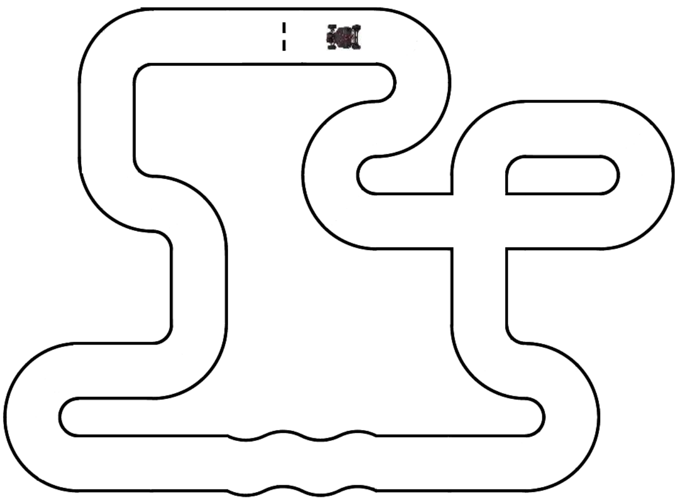

Fig 7. Example of car Track test.

3.2 Lightning system and parasitic light sources

Many lightning technologies were developed in the past few years. In our study, we focus on using the 50hz/60hz light system, which can cause reading fluctuations even if the car is on stop position. Indeed, the integration time of the camera sensor is 10ms, which is the deadline for our real-time car system. Thus, we developed an adaptative level threshold algorithm to binarize camera data flow. Others Technologies like Led or CFL with electronic ballast are easy to handle by the camera, which saw it like DC source of light.

Furthermore, in particular, the situation the car can experiment with glare conditions [5] can resolve by an adaptative exposition time of the photodiodes sensor. Sunlight, powerful projector, or flashlight can cause this issue. In our participation in NXPCUP 2019 final, we had the same problem resolved by putting a filter on the camera lens.

3.3 Mechanical vibration

Like all vehicles, the mechanical vibrations are generated basically by actuators or unbalanced wheels. In our case, there are two mains issues. The first one is camera misreading [6], which we resolved by detecting and avoiding the data infinite state machine algorithm implemented in our software. The second one is the parasitic values of speed measurements from the two optical encoders [7]. We used a digital low pass filter as a solution.

3.4 Hardware failure

Numerous hardware failures were observed caused by low-quality spares parts, no well-designed electronics systems, or degradation by car crashes. We took special attention to the wiring and connecting the system to avoid failures during competition.

4 Algorithm Implementation

In this part, we present the architecture of the algorithm with different stages. The data processing flow was achieved in the duration of 10ms for each camera reading and actuators updating values. We used several programming technics and dedicated modules in the KL25Z processor to reach our goals.

4.1 Software architecture

Fig 9 shows the global architecture of our real-time software:

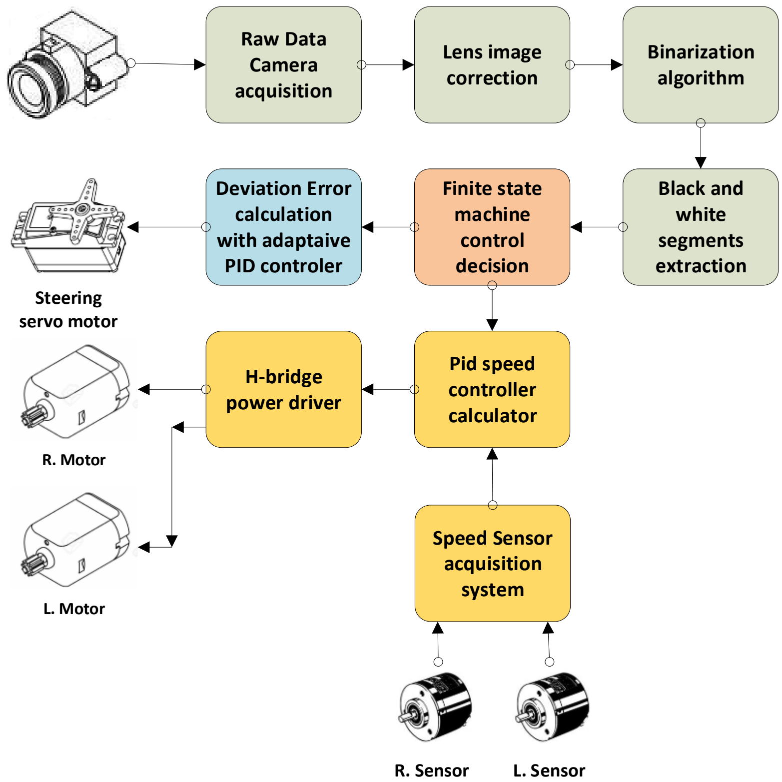

Fig 8. Architecture of real time smart car software controller.

4.2 Camera data flow processing

The first step is camera data acquisition. We use a timer, and ADC interrupts to trigger pixel reading from the camera. The raw data was stocked in an array for the next step processing. After that, we apply a correction function to raw data image, which was modelized as a reverse function of polynomial approximation of camera white reading. Fig 10 shows the curve before and after corrections. In the third step, we binarize data using a dynamic threshold calculated from the max and min of light intensity value at each data reading. After that, we extract the data black and white segment by recording a start pixel segment number and segment length.

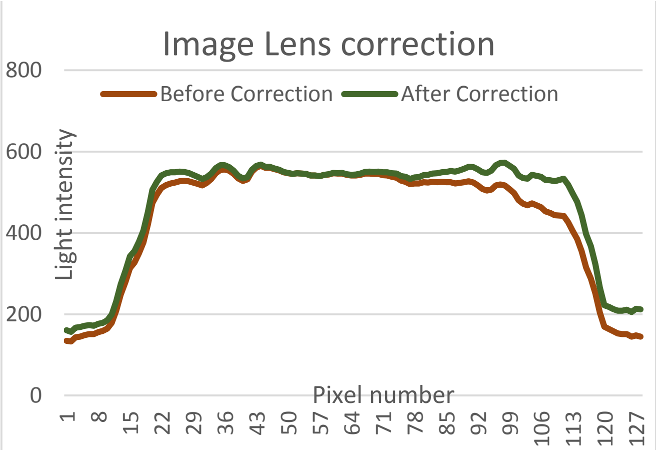

Fig 9. Raw data camera before and after correction.

4.3 Machine state control decision

In this stage, we took the position and length of the segments to calculate two kinds of parameters. The first one is car position error on the track with fault state avoiding. The adaptative PID steering control will use the positioning error. The second one is speed references for each left and right motors used in the PID speed controller. The stage can detect pattern sequences to start/stop or speed changing according to specials marks on the track.

4.4 PID controllers for DC Motors speed and adaptative steering

The PID command consists of three terms P, I, and D, hence the 'P' corresponds to the proportional term, 'I' for the integral, and 'D' for the term derived from the command. The PID controller is a continuous error calculating the feedback approach. The error calculated as the difference between the reference value and the measured one. It can eliminate the compensation for the state of equilibrium through integral action, and it can anticipate the future through a derivative action.

Those stages aim to implement two PID commands for both DC motors and steering systems with separate excitation (to obtain an adequate response from the process and regulation and have a precise, fast, stable and robust system). The adjustment of the coefficients (parameters) of the PID command is based on the empirical method of "Ziegler & Nichols" [8].

As the first step, we use the empirical method to specify the coefficient for each DC Motor even if they seem the same. We use a wireless data acquisition for recording and get the right factors, which generally depend on the vehicle weight. The steering system is much more complicated because that car behavior depends on the global speed and steering angle. For what we make an array of PID parameters depending on car speed. Thus, the car behaves correctly in curves and straights.

4.5 Speeds measurement

To achieve speed measurement, we use the sequent measure of each sensor channel by the one-timer module in 5ms for each sensor. An interrupt, help the processor to switch between channels. After that, we apply a digital low pass filter to avoid parasitic values.

5 Conclusion

In this work, we achieve several improvements in our software system on each stage. We used several methods and technics to solve different issues caused by a disturbed environment. Excellent skills in hardware and software programming are needed to optimize code implementation. We project for future works to add more sensors like accelerometer and gyroscope to merge data for better control software algorithms, especially for drifting cases. Otherwise, we preconize to use a novel parallel platform like GPU for an embedded system; it's seams that will be the better hardware platform for the autonomous vehicles development.

References

[1] Katare, D., El-Sharkawy, M.: Embedded System Enabled Vehicle Collision Detection: An ANN Classifier. In: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC) 2019, pp. 0284-0289. IEEE

[2] Gao, T., Lai, Z., Mei, Z., Wu, Q.: Hybrid SVM-CNN Classification Technique for Moving Targets in Automotive FMCW Radar System. In: 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP) 2019, pp. 1-6. IEEE

[3] Ren, J., Ren, R., Green, M., Huang, X.: A deep learning method for fault detection of autonomous vehicles. In: 2019 14th International Conference on Computer Science & Education (ICCSE) 2019, pp. 749-754. IEEE

[4] Essaid, M., Idoumghar, L., Lepagnot, J., Brévilliers, M.: GPU parallelization strategies for metaheuristics: a survey. International Journal of Parallel, Emergent and Distributed Systems **34**(5), 497-522 (2019).

[5] Hamm, M., Huhn, W.: Glare Investigations and Safety Research on Digital Light Technologies. In. SAE Technical Paper, (2019)

[6] Verma, M., Collette, C.: Active vibration isolation system for drone cameras. In: Proceedings of International Conference on Vibration Problems: ICOVP 2019

[7] Alejandre, I., Artés, M.: Method for the evaluation of optical encoders performance under vibration. Precision engineering **31**(2), 114-121 (2007).

[8] Shariff, H.M., Rahiman, M.H.F., Adnan, R., Marzaki, M.H., Tajjudin, M., Jalil, M.H.A.: The PID Integrated Anti-Windup Scheme by Ziegler-Nichols Tuning for Small-Scale Steam Distillation Process. In: 2019 IEEE 9th International Conference on System Engineering and Technology (ICSET) 2019, pp. 391-395.

View publication stats