PROGRAMMING THE ALGORITHM TO CONTROL AIRBAG ACTIVATION TIMING ON A PHYSICAL SIMULATION MODEL

Dam Van Trang*, Nguyen Phuoc Thanh
Faculty of Mechanical Engineering, District Twelve Technical Economic College.

Abstract:

The paper presents the results of programming an algorithm that adjusts the control timing for airbag deployment. The C++ programming language was selected to develop the control algorithm for the simulation model. The results show that, when applying this algorithm to an airbag control model, it is possible to determine the deployment timing of the airbag corresponding to the magnitude of the impact force. In other words, the greater the force acting on the sensor, the deployment time of the airbag changes accordingly. This programmed algorithm also allows parameter settings to be modified by users, enabling further optimization of airbag performance in the future.

Keywords: *Airbag; algorithm; Collision; Deployment time; Simulation.*

DOI:

Lập trình thuật toán điều khiển thời gian kích hoạt túi khí trên mô hình mô phỏng vật lý

Tóm tắt:

Bài báo trình bày kết quả lập trình thuật toán thay đổi thời gian điều khiển kích hoạt túi khí. Ngôn ngữ lập trình C++ đã được lựa chọn để viết lập trình thuật toán điều khiển mô hình. Kết quả nghiên cứu cho thấy, khi áp dụng thuật toán này lên mô hình điều khiển túi khí thì ta có thể xác định được tính năng thời gian túi khí được kích hoạt tương ứng với độ lớn của lực tác động. Tức là khi lực tác động vào cảm biến càng lớn thì thời gian kích hoạt túi khí sẽ thay đổi tương ứng. Thuật toán lập trình này chúng ta có thể thay đổi thiết lập số liệu để người sử dụng có thể thay đổi nhằm cải tiến để túi khí hoạt động tối ưu hơn sau này.

Từ khóa: *Túi khí; Thuật toán; Va chạm; Thời gian kích hoạt; Giả lập.*

1. Introduction

As automotive technology continues to advance, safety features in vehicles are increasingly emphasized. Among the current automotive safety systems, the airbag is an indispensable component for protecting occupants. Its primary purpose is to enhance passenger safety when deployed. The topic entitled “Programming the algorithm to control airbag activation timing on a physical simulation model” was selected to address this problem.

2. Airbag Operating Principle

This system operates through three main stages from the moment a collision occurs until the airbag deploys.

Stage 1: The control unit receives signals from collision, acceleration, and speed sensors to assess the severity of the impact. When this value exceeds a predefined threshold, the igniter in the

inflator is triggered (Toyota, 2004; Toyota, 2018).

Stage 2: The high temperature generated by the activated igniter immediately burns the initiator material and spreads to the gas-generating compound (NaN_3), producing a large volume of nitrogen gas. Under high pressure from the expanding gas, the airbag tears through the outer cover of the steering wheel and deploys. The electrical current used to activate the airbag igniter typically ranges from 1A to 3A within approximately 2 milliseconds to ignite the initiator and gas-generating material (Toyota, 2004; Toyota, 2018).

Stage 3: The airbag is fully inflated to reduce the impact force on the occupants, and the gas is then quickly released through vent holes at the rear of the airbag (Toyota, 2004; Toyota, 2018).

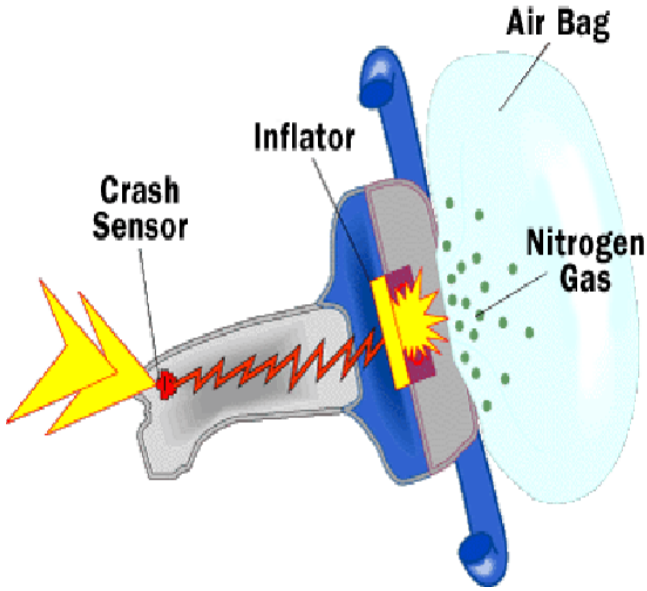

Figure 1. Airbag Operation

(Source: Adapted from the Toyota (2004) training manual)

3. Software Selection for Control Algorithm Programming

After studying programming methods, the C^{++} programming language combined with the Arduino IDE software was selected to program and execute the control algorithm for the airbag operation on the physical simulation model. The software is user-friendly, making it convenient to input data and adjust parameters based on the programmer's calculations. The software interface is shown in **Figure 2**.

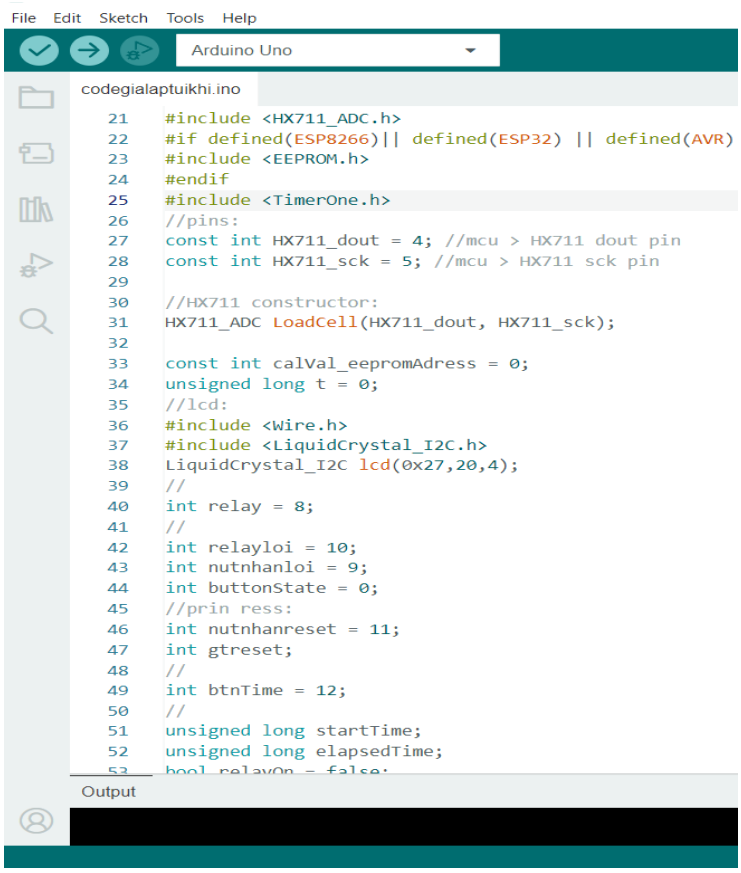

Figure 2. Arduino IDE Software

(Source: Prepared by the authors)

4. Programming the Control Algorithm

To control the operating capability of the model, the selected input data was the mass applied to the sensor to trigger the airbag deployment. The programmed algorithm also allows this input data to be modified, thereby supporting the future development and expansion of the model. The algorithm defines a minimum threshold weight required to activate the airbag model. If the applied weight is below this threshold, the airbag model will not deploy.

At different weight levels, the airbag deployment time varies correspondingly. All these parameters can be easily reconfigured to refine the control system for more advanced research, aiming to determine the most optimal operating conditions.

Mass was selected as the reference unit for determining the airbag activation timing because based on the formula, the magnitude of the impact force can be converted into mass. This approach simplifies the design while improving the safety and durability of the model.

```
283         }
284         lcd.setCursor(9, 0);
285         lcd.print(String(hours) + ":" + String(minutes) + ":" + (seconds < 10 ? "0" : "") + String(seconds) + ":" + String(milliseconds) );
286     }
287     currrentValue = integerValue;
288     //dk:
289     if(integerValue < 3000  {
290       digitalWrite(giaiapprelay,LOW);
291       digitalWrite(relay,LOW);
292       lcd.setCursor(12,2);
293       lcd.print("OFF");
294     }
295     //DK1
296     if (integerValue >= 3000 && integerValue < 5000 && M1==0)
297     {
298       M1=1;
299       if(getMaxValue==1)
300       {
301         Serial.println("DK1");
302         lcd.setCursor(12, 2);
303         lcd.print("ON ");
304         digitalWrite(relay, HIGH);
305         milliseconds=0;
306         while(milliseconds<30)
307         {
308           hours = 0;//elapsed_time / 3600;
309           minutes = 0;// round(elapsed_time / 60) % 60;
310           seconds =0;// (int)elapsed_time % 60;
311           milliseconds++;// (int)((elapsed_time - (int)elapsed_time) * 1000);
312           lcd.setCursor(9, 0);
313           lcd.print(String(hours) + ":" + String(minutes) + ":" + (seconds < 10 ? "0" : "") + String(seconds) + ":" + String(milliseconds) );
314           delay(1);
315         }
```

Figure 3. Algorithm Programming Data for Airbag Model Activation

(Source: Prepared by the authors)

The overall operation of this model can be generalized through the logic block diagram shown in **Figure 4** below. At the same time, the performance and responsiveness of the airbag system in this model are verified through the graph presented in **Figure 6**.

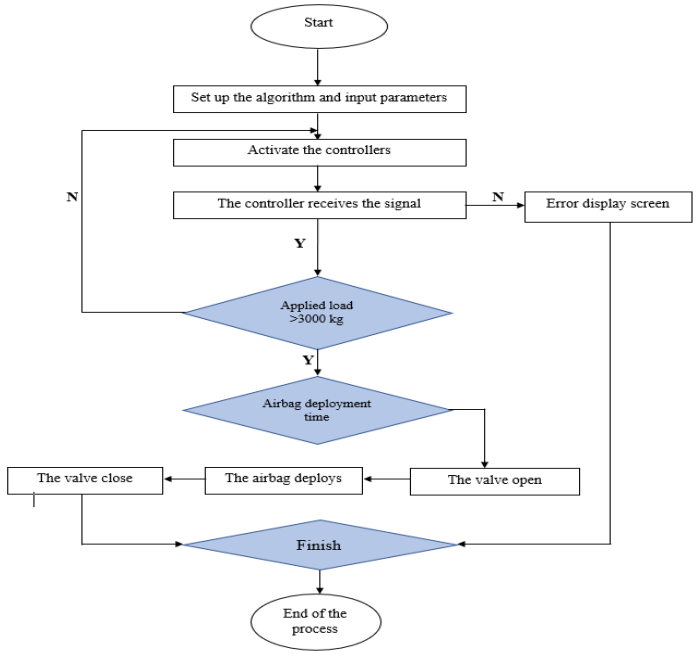

Figure 4. Logic Block Diagram of the Model Operation Based on the Algorithm

(Source: Prepared by the authors)

Table 1: Experimental Data Table of Airbag Operation on the Simulation Model

N.o	Valve opening activation time (ms)	Applied mass (kg)	Airbag malfunction status
1	30	3129	OFF
2	30	4221	OFF
3	20	5164	OFF
4	20	6108	OFF
5	10	7020	OFF
6	10	8114	OFF
7	10	9151	OFF

(Source: Prepared by the authors)

After conducting experimental measurements on the model, the collected data were used to construct the following airbag operation graph:

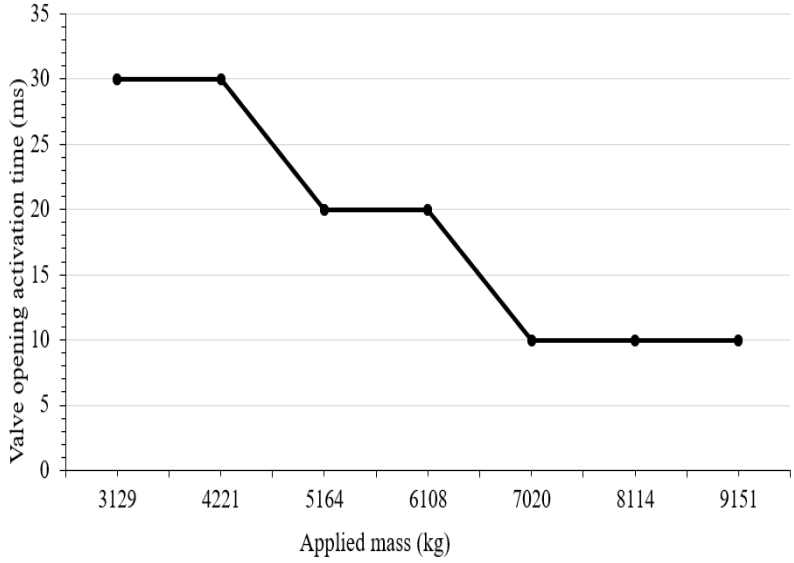

Figure 5. Airbag Operation Graph at Different Weight Levels

(Source: Prepared by the authors)

According to the graph illustrating the response capability of the airbag model, it can be observed that as the applied mass on the sensor increases, the time required to activate airbag deployment decreases. This improves the likelihood of ensuring the safety of the driver and passengers inside the vehicle. At the same time, it satisfies the initial objective: the airbag deployment time must correspond to the severity of the collision (Cho et al., 2010; Rajendra & Puranik, 2014).

In practice, a car traveling at approximately 25 km/h and experiencing a frontal collision at the

sensor location can trigger airbag deployment (Chawla et al., 2014). The average mass of a car is about 1000 kg. If the vehicle is moving at 25 km/h and comes to a complete stop within 0.2 seconds during a collision, the impact force can be calculated using the following formula (Shaikh et al., 2013).

$$F = m \cdot \frac{\Delta V}{\Delta t} \quad (1)$$

In this example, the mass of the vehicle (m) is 1000 kg, the velocity before impact (v_0) is 25 km/h (converted to m/s, v_0 = 6.94 m/s), and the collision time (Δt) is 0.2s (Steinbuch, 2014).

Δv represents the change in velocity after the collision. In this case, the final velocity v = 0 m/s, since the vehicle comes to a complete stop after impact.

$\Delta v = v - v_0 = 0 - 6.94 = -6.94$ m/s (Merkle & Harrigan, 2016).

Substituting into the formula, we have:

$$F = \frac{1000.(-6.94)}{0.2} = -34700 \text{ N} \quad (2)$$

Since the force value is negative, its direction is opposite to the initial direction of motion. Therefore, the magnitude of the impact force is 34,700 N (Tabani, 2017; Tike & Chaudhari, 2014).

To enhance the experimental aspect, mass was chosen instead of collision force because measuring mass is much easier, safer, and more practical to implement. From the above example, the impact force of 34,700 N can be converted to approximately 3,500 kg. In this model, the minimum weight required to activate the airbag in the programmed data is set at 3,000 kg. If the applied weight is below this threshold, the airbag will not deploy (Mai et al., 2023).

5. Structure and Operating Principle

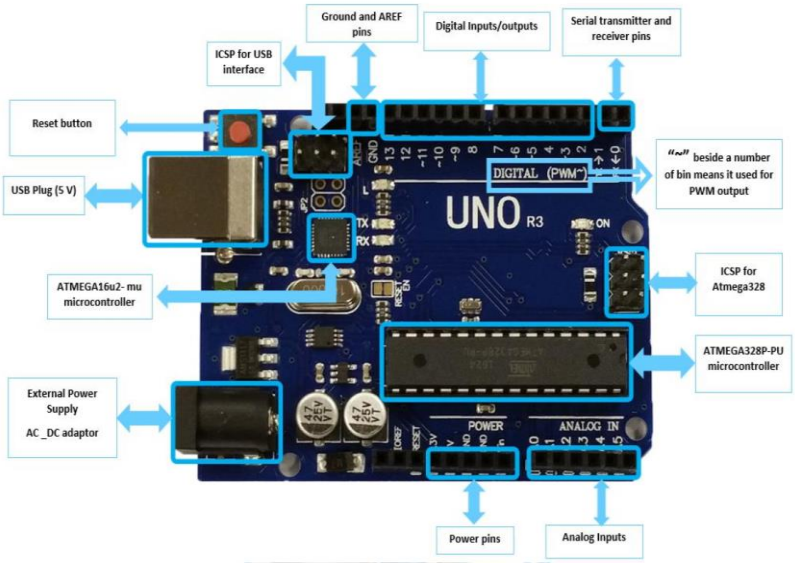

Figure 6. Arduino Chip Used in the Model

(Source: Arduino Uno R3 Microcontroller Overview)

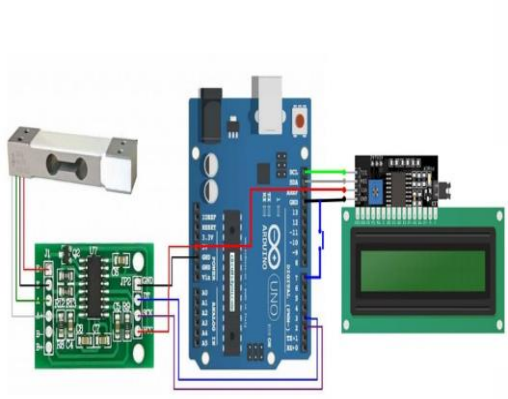

Figure 7. Pinout Annotation Diagram on the Circuit Schematic

(Source: Prepared by the authors)

After completing the programming and testing of the algorithm, the next stage was to select and assemble the electronic components, then upload the data to the Atmega328P microcontroller integrated on the Arduino board, which features 14 digital pins and 6 analog pins, for the purpose of controlling the entire operation of the model. The circuit diagram of several components in the airbag simulation model is configured according to the following specifications.

Operating Principle: This airbag operation simulation model uses an Arduino microcontroller in place of the control unit. A relay is employed to switch a solenoid valve on and off, allowing compressed air to flow from an air compressor into the airbag, thereby simulating airbag deployment. The solenoid functions as a valve to release air instead of using a real igniter, since an igniter can only be used once and is not suitable for repeated experiments.

The control system also uses a voltage conversion circuit to reduce the battery voltage from 12V to 9V for the Arduino microcontroller, preventing damage to the chip caused by overvoltage or overload. In addition, the model is equipped with an emergency stop switch, which can be activated immediately in case the input power becomes excessively high and causes a short circuit or fire hazard.

After checking and supplying power to all electrical components, the model operates as follows: A minimum load of 3 kg is placed on the load cell sensor. Through the HX711 weight conversion module connected to the load cell, the programmed algorithm on the Arduino interprets this input as an equivalent impact of 3000 kg. The Arduino then activates the relay, opening the solenoid valve to allow compressed air to inflate the airbag, simulating deployment.

Once the airbag is deployed, the reset button is pressed to return the system to its initial state, and the airbag is folded

back for the next experiment. Because the model uses an air compressor, the discharged air pressure is relatively strong, and the model can also be reused multiple times after activation. This makes it suitable for teaching, student learning, and experimental practice in future.

6. Conclusion

This paper presents the results of programming a control algorithm for the operational behavior of an airbag simulation model. Experimental results show that when applying this algorithm to the model, several control characteristics can be observed and verified:

As the applied mass on the sensor increases, the airbag deployment time decreases.

The algorithm allows adjustment of input parameters, including mass and deployment time, to achieve optimal safety performance.

If the sensor signal is lost, the microcontroller displays an error on the screen, similar to real automotive systems, enabling users to identify system faults.

After completing an experiment, pressing the reset button restores all input parameters to their initial state, allowing repeated experiments and data collection under different conditions.

References

Toyota. (2004). *Electrical diagnosis technician 2: SRS airbag and emergency seatbelt pretensioner Training manual*.

Toyota. (2018). *SRS Toyota Yaris: Supplemental restraint system – Airbag system Technical manual.*

K. Cho, S. B. Choi, S. Wee, & K. Shin. (2010). Design of an airbag deployment algorithm using a radar sensor. *In Proceedings of the 6th IFAC Symposium on Advances in AutomotiveControl*. Munich, Germany.

M. A. Rajendra, & V. G. Puranik. (2014). *Airbag deployment system based on pre-crash information*.

Ishpreet Chawla, et al. (2014). Advancement in vehicle airbag deployment system. *International Journal of Applied Engineering Research,* 9(20).

Tasnim N. Shaikh, et al. (2013). Air bag: A safety restraint system of an automobile. *International Journal of Engineering Research and Applications*, 3(5).

M. Steinbuch. (2014). *Active airbag systems.* Eindhoven University of Technology, Department of Mechanical Engineering, Control Systems Technology Group.

Andrew C. Merkle, & Timothy P. Harrigan. (2016, June). *Research literature review: The use of air bags for mitigating grade crossing and trespass accidents.*

Yacoob Tabani. (2017). *Automobile air bag inflation system based on fast combustion reaction* (Master's thesis, New Jersey Institute of Technology).

Rishikesh H. Tike, & Mukesh C. Chaudhari. (2014). The airbag system for two wheeler vehicle system. *International Journal of Research in Aeronautical and Mechanical Engineering.*

Mai Phuoc Trai, Nguyen Phuoc Thanh, Pham Quoc Phong, Phan Van Tuan, & Doan Nguyen Uyen Minh. (2023). Building a control algorithm for automotive airbag response capability. *Tra Vinh University Journal of Science, 13(SpecialIssue).* https://doi.org/10.35382/TVUJS.13.6.2023.2119.

Article info.

Received date: 2/5/2026

Revised date: 20/6/2026

Accepted date: 25/6/2026

Corresponding author: Dam Van Trang

Email: vantrang7794@dttec.edu.vn