

논문 2017-12-28

빅데이터 분석을 통한 운전자 맞춤형 엔진 제어 장치 시스템의 개발

(Development of a Driver-Oriented Engine Control Unit
(ECU)-Mapping System With BigData Analysis)

김 식, 김 정 환*

(Shik Kim, Junghwan Kim)

Abstract : Since 2016 when the regulations related to vehicle structure and device modification were drastically revised, the car tuning market has been growing rapidly. Particularly, many drivers are showing interest in changing the interior and exterior according to their preference, or improving the specifications of their cars by changing the engine and powertrain, among others. Also, as the initial engine settings such as horse power and torque of the vehicle are made for stable driving of the vehicle, it is possible to change the engine performance, via Engine Control Unit (ECU) mapping, to the driver's preference. However, traditionally, ECU mapping could be only performed by professional car engineers and the settings were also decided by them. Therefore, this study proposed a system that collects data related to the driver's driving habits for a certain period and sends them to a cloud server in order to analyze them and recommend ECU mapping values. The traditional mapping method only aimed to improve the car's performance and, therefore, if the changes were not compatible with the driver's driving habits, could cause problems such as incomplete combustion or low fuel efficiency. However, the proposed system allows drivers to set legally permitted ECU mapping based on analysis of their driving habits, and, therefore, different drivers can set it differently according to the vehicle specifications and driving habits. As a result, the system can optimize the car performance by improving output, fuel efficiency, etc. within the range that is legally permitted.

Keywords : ECU-Mapping, CAN, Bigdata, Edison board, Cloud storage

I. 서 론

2016년 국토교통부는 “자동차 구조, 장치 변경에 관한 규정” (고시 제2014-462호)을 개정하였다 [1]. 이전의 규정은 자동차의 구조, 장치 변경을 엄격히 금지해 왔던 규정을, 용어를 튜닝으로 수정하고 경미한 튜닝을 제외한 자동차의 구조와 장치 변경은 교통안전공단에 신고 후 검사가 완료가 되면

*Corresponding Author (jhwem82@gmail.com)

Received: Apr. 19 2017, Revised: May 6 2017,

Accepted: May 19 2017.

Shik Kim, Junghwan Kim: Semyung University

※ 본 논문은 2015년도 세명대학교 교내학술연구비 지원에 의해 수행된 연구임

등록이 될 수 있게 네거티브 허용을 기본으로 규정을 변경하였다.

그로 인해 기존에 대부분의 튜닝은 자동차 정비소나 개인이 불법으로 하는 경우가 많았지만 환경부의 “운행차 배출 가스 검사 시행요령 등에 관한 규정”이 대대적으로 수정되면서 합법적인 튜닝 시장은 크게 성장하고 있다 [1]. 엔진의 마력, 토크, 최대 속도, 엔진 점화 시기 및 공기와 연료량 조절을 설정하여 엔진성능을 높이는 ECU (Engine Control Unit)-Mapping의 경우 최고 속도 제한 장치 해제와 배기가스 배출 허용치를 초과 하지 않는 범위 내에서 합법화되었다.

통상 운전자의 운전 습관은 각자마다 다르지만 각 자동차의 출고 시 ECU 수치는 동일하게 설정된

다. ECU-Mapping을 하려는 운전자도 자신의 운전 습관에 맞춰 설정을 하는 것이 아니라 ECU 튜닝 매장에서 추천하는 ECU-Mapping 수치로 설정한다.

ECU-Mapping의 경우 일반 운전자가 접근하기에는 전문지식이 필요하며 또한, 잘못된 설정을 하게 될 경우 불법으로 간주될 수 있으며 엔진에 손상이 갈 수 있기 때문이다. ECU-Mapping을 한 뒤에도 운전자와 맞지 않거나 배기가스의 배출이 법적 허용치를 초과할 경우 다시 설정을 해야 한다.

본 논문에서는 먼저, 차량 운행 때 마력, 토크 등 ECU 정보들을 정해진 기간 동안 실시간으로 획득하여 클라우드 스토리지에 저장한다. 그 후, 축적된 데이터를 분석하여 운전자에게 적합한 ECU-Mapping 수치를 결정한 후 ECU에 추천된 수치 데이터를 설정하는 시스템을 개발하였다.

이 시스템을 사용한다면 각 운전자의 운전 습관과 주행 환경에 맞춘 ECU-Mapping이 가능할 뿐만 아니라, 1년 이상 ECU 데이터가 수집될 경우 계절별 상이한 설정이 가능하며 그 결과 일반 운전자가 전문가의 도움 없이 관련 법규를 준수하면서 쉽게 차량 성능을 높이며 연료의 연소효율을 개선될 것으로 기대된다.

II. 본 론

운전 습관을 수집하여 ECU-Mapping 수치를 추천하는 시스템을 구현하는데 필요한 차량용 네트워크인 CAN (Controller Area Network)와 시스템 개발에 사용한 오픈소스하드웨어 (OSHW) 기반의 인텔의 에디슨 보드를 간단히 소개하고, 개발코자하는 타겟서비스인 기존 ECU-Mapping 기술의 소개와 문제점을 요약한다.

1. CAN

자동차의 발전으로 인해 많은 전장 장비가 탑재되면서 전용 배선이 증가되었다. 그로인해 비용, 복잡성, 무게 등 많은 문제점이 나타나게 되어 통합된 차량용 네트워크의 필요로 인해 1985년 독일의 Bosch에서 개발한 차량용 통합 네트워크 시스템이다 [2].

현대 자동차를 포함한 국내 4사와 GM, BMW 등 국내외 자동차 제조사들은 차량용 네트워크망인 CAN을 채택한 이유는 다음 4가지 장점 때문이다 [3, 4].

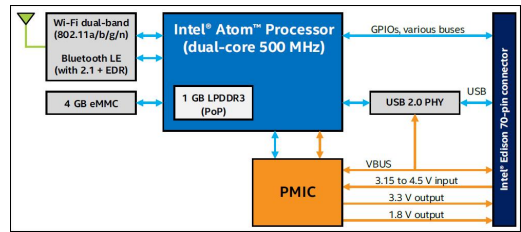


그림 1. 에디슨 보드의 블록 다이어그램
Fig. 1 Edison board's block diagram

첫째, 저비용, 경량의 네트워크 시스템이다.

둘째, Broadcast 통신 기반으로 인해 모든 전장 장비의 메시지를 확인 및 제어가 가능하다.

셋째, 전장 장비의 우선순위를 선정할 수 있다.

넷째, 오류 검출이 쉬워졌다.

본 연구에서 ECU-Mapping을 위한 데이터 수집은 Broadcast 통신 기반의 CAN을 통해 쉽게 이루어지게 된다.

2. 에디슨 보드

인텔에서 개발한 SoC (System on Chip) 형태로 아두이노와 오픈소스하드웨어 완전 호환관계를 유지하면서, CPU는 Intel Atom Processor와 1Giga급의 DRAM과 4GB의 플래시 메모리를 내장하고 있고, 다양한 프로그래밍으로 모듈형 기기를 제어할 수 있는 개발 플랫폼이다 [5].

그림 1는 에디슨 보드의 다이어그램이다. 1GB의 LPDDR3 RAM과 4GB 플래시 메모리가 장착되었다. 그리고 통신을 위해 802.11 (a, b, g, n) WiFi, Bluetooth 4.0, SD, UART, SPI, GPIO 및 USB 2.0 OTG가 빌트인으로 장착되었다.

기본 제공되는 모듈에서 외부 인터페이스를 이용하여 사용자의 선택에 따라 확장 보드를 장착할 수 있으며 오픈 소스 H/W와 호환되는 핀홀이 제공한다. 그리고 OS는 오픈 소스 기반의 Yocto Linux, 개발 환경은 Aduino IDE, C, C++, Python, Node.js, HTML5을 지원한다 [6].

특히, IoT 개발자들을 위해 인텔에서 클라우드 스토리지를 무료로 제공받음으로써 장기간에 걸친 데이터 수집 시 저장 공간에 대한 제한을 해결할 수 있다. 인텔 클라우드 스토리지는 연결된 보드의 MAC 주소를 기반으로 데이터가 저장되기 때문에 IP나 통신망이 변경되더라도 저장된 데이터에 접근이 가능하다.

호환성이 뛰어난 장점을 가지고 있기 때문에 기존 아두이노 개발 틀을 사용했던 개발자들도 동일한 환경에서 좀 더 확장된 임베디드 개발 환경을 사용할 수 있다.

3. ECU-Mapping

ECU-Mapping은 엔진의 마력, 토크 및 엔진 연료 압축비 등을 설정하여 운전자의 운전 습관과 환경에 맞춰 설정하는 것이다. 초기 차량의 ECU 설정 수치는 운전자의 운전 습관과 상관없이 일괄된 고정된 수치로 설정되어 있다. 그래서 인테리어와 차량의 외형을 주로 튜닝했던 운전자들이 엔진 성능도 ECU-Mapping을 통해 차량의 성능을 높이는 튜닝을 하면서 새로운 고부가 가치 기술로 발전되고 있다 [7].

ECU는 엔진 제어에 필요한 데이터를 ROM영역에서 보관을 하고 마력, 토크 등의 각종 엔진 수치의 한계 값을 설정하여 넘지 않도록 제어를 한다. 한계 값을 초과할 경우 엔진에 무리가 갈 수 있으며 배기가스가 법적 허용치 이상으로 배출될 수 있기 때문에 유의해야 한다 [8].

표 1은 ECU가 엔진을 명령하는 데이터와 출력 가능한 데이터 중 중요한 항목들을 정리한 것이다.

각 항목들은 ECU가 자동차 제조사에서 설정한 수치를 기준으로 이론적인 수치이며, 실제 각 수치는 높거나 낮을 수 있다.

가령, RPM 1000과 엔진 부하 10%의 구간은 엔진 연료 압축비를 10으로 설정되어 있더라도 연료관에 타르, 연료 찌꺼기 등 엔진 때가 끼어 있을 경우 실제 압축비는 더 낮게 나타난다.

RPM과 엔진 로드를 기준으로 엔진 연료 압축비를 설정을 하게 되고 그것을 기준으로 마력과 토크가 결정된다.

ECU-Mapping은 엔진 연료 압축비가 가장 중요하다. 왜냐하면 엔진에 액체 상태의 연료가 들어간다면 인화점에 도달하는데 상당한 시간이 걸리지만 연료의 입자를 작게 만들고 노출 면적을 넓게 만들기 위해 연료에 강한 압력을 주어 기체 상태로 만들어 엔진에 분사된다. 이때, 기화시키는 역할을 하는 것이 엔진 연료 압력이다.

최초 차량이 출고될 때 운행에 관련된 엔진과 ECU 각 수치는 최대 성능으로 설정이 되어 있는 것이 아니라 실제 사양보다 다소 낮게 설정된다. 그 이유는 각 운전자의 운전 습관과 운행 환경이 다르기 때문에 고속이나 저속에 맞추는 것이 아니라 저속 50%, 고속 50%를 사용하는 경우에 맞춰 설정된다.

표 1. ECU 관련 데이터
Table 1. ECU related data

List	Display	Explanation	unit
Speed	"	Speed	km/h
Engine RPM	Revs	Engine Speed	rpm
Engine load	"	Engine load	%
Barometric pressure	Baro	Vehicle aroundtemperature	°C or °F
Ambient air temperature	Air temp	Engine aroundtemperature	°C or °F
Air Fuel Ratio	AFR	Fuel efficiency	l/100km
Fuel Rail Pressure	Fuel Rail	Fuel Rail Pressure	
Intake Manifold Pressure	Intake	Intake Manifold Pressure	°C or °F
Mass air flow rate	MAF	Mass air flow rate	%
Intake air temperature	Intake	Engine intake air temperature	°C or °F
Relative throttle position	R THR	Relative throttle position	°
Throttle position	throttle	Throttle position	°
Torque	Torque	Torque	Nm
CO ₂ in g/km	ac.CO ₂	CO ₂ in g/km	
Turbo boost & Vacuum gauge	Boost	Turbo boost & Vacuum gauge	psi
Engine coolant temperature	ECT	Coolant temperature	°C or °F
Horsepower (wheel)	HP	Horse power (wheel)	HP
Voltage		Vehicle inside Volt	v
Kilometers pre litre	KPL	Kilometers pre litre (instant)	km/L
Kilometers pre litre	KPL	Kilometers pre litre (Longterm)	km/L
Acceleration Sensor	Accel	Acceleration pedal zero point adjustment	
Acceleration peralposition D	Pedal D	Acceleration pedal pressure	
Acceleration peralposition E	Pedal E	Accelerator pedal monitoring	
EGR Commanded	EGR	Engine intake figure of exhaust gas	
Fuel flow rate/minute	Fuel flow	Fuel flow rate/minute	

	0	2	4	7	9	11	14	16	18	21	23	25	28	30	32
1500	0	2	5	7	9	12	14	14	14	14	14	14	14	14	14
2000	0	2	5	7	9	12	14	17	19	21	24	26	26	26	26
2500	0	2	5	7	9	12	14	17	19	21	24	26	28	31	31
3000	0	2	5	7	9	12	14	17	19	21	24	26	28	31	33
3500	0	2	5	7	9	12	14	17	19	21	24	26	28	31	33
4000	0	2	5	7	9	12	14	17	19	21	24	26	28	31	33
4500	0	2	5	7	9	12	14	17	19	21	24	26	28	31	33
5000	0	2	5	7	9	12	14	17	19	21	24	26	28	31	33
5500	0	2	5	7	9	12	14	17	19	21	24	26	28	31	33
6000	0	2	5	7	9	12	14	17	19	21	24	26	28	31	33
6500	0	2	5	7	9	12	14	17	19	21	24	26	28	31	33
7000	0	2	5	7	9	12	14	17	19	21	24	26	28	31	33

그림 2. ECU 맵 데이터

Fig. 2 ECU map data

그리고 ECU-Mapping을 마력과 토크만 올린다고 해서 차량의 성능이 높아지는 것이 아니라 각 수치에 맞는 공기와 연료량도 알맞게 혼합하여 엔진에 공급해야한다. 불완전 연소를 할 경우 연료의 대부분은 매연이 되어 배출되기 때문에 배기가스와 관련된 수치를 확인하는 것이 중요하다.

배기가스 법적 허용수치는 차량의 연료에 따라 다르지만 휘발유를 기준으로 일산화탄소 0.6/1.2% 이하, 탄화수소 120/220ppm, 질소산화물 790ppm 이하, 공기과잉률 0.90~1.10이다.

이 수치를 ECU에 설정하기 위해 ECU-Mapping을 시행하였을 때 허용수치 이상이라고 판단할 경우 Mapping 하는 도중에 경고가 나타난다.

ECU-Mapping은 엔진 연료 압축비를 변경하여 동일한 RPM과 엔진 부하를 사용하더라도 연료를 적게 사용하며 더 높은 마력과 토크를 사용할 수 있게 하는 것이다.

그림 2의 ECU-Mapping을 위해 사용하는 ECU 맵 데이터의 일부이다.

맵 데이터를 보는 방법은 X축의 엔진 로드율과 Y축 RPM이 input이며 중앙의 데이터는 연료 압력을 나타낸 output이다. 맵 데이터를 읽는 방법은 Y축 RPM이 1500에서 엔진 부하를 2%를 주었을 때 엔진 압력은 5bar이다.

개인이 ECU-Mapping이 어려운 이유는 페달을 밟았을 때의 각과 RPM을 결정하여 토크가 결정되고 이 변수를 통해 기초 연료량이 선택이 되며, 이를 통해 부스트, 레일 압력, 흡기량 및 전기량 등을 계산한 뒤 설정을 해야 하기 때문이다.

이러한 어려움 때문에 운전자의 운전 습관과 환경을 고려하지 않고 자동차 정비소에서 추천하는 값을 설정하게 된다.

그래서 본 논문의 시스템은 ECU의 모든 정보를 일정 기간 클라우드 스토리지에 전송 후 빅데이터를

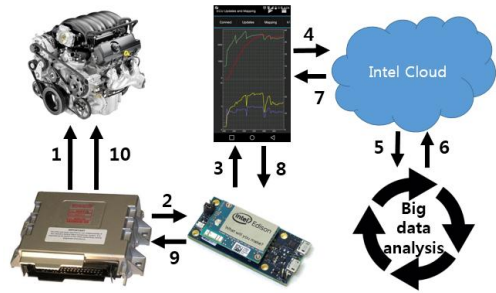


그림 3. 시스템 작동 순서

Fig. 3 System operating sequence

구성하여 ECU-Mapping과 관련된 메커니즘을 구현한 프로그램을 통해 자동으로 분석한다. 운전자는 자신에게 맞는 ECU-Mapping 값을 통해 차량 출력 상승 및 효율적인 연료의 연소를 돕는 시스템을 소개한다.

이때, ECU-Mapping과 관련된 메커니즘은 합법적인 선 안에서 설정이 이루어지며 위험한 설정일 경우 ECU 상에서 경고를 한다.

III. 개발 및 실험

본 논문이 작동하는 순서는 그림 3와 같다. 최초 ECU는 1번 경로로 엔진에게 마력, 토크, 엔진 연료 압축 비등을 전송하게 된다. 이 후 2번에서 각 정보들은 실시간으로 OBD2 (On-Board Diagnostics 2)를 이용해 에디슨 보드로 전송된다. 이때 에디슨 보드 내 LCD화면을 통해 정보를 확인할 수 있다. OBD2는 ECU의 데이터를 진단, 모니터링 할 수 있는 국제 표준 소켓이다.

에디슨 보드는 3번 과정 중에 블루투스를 사용하여 스마트폰으로 정보들을 실시간으로 전송하고 4번에서 스마트폰은 무선통신망을 이용하여 인텔 클라우드 스토리지로 전송하게 된다.

일정 기간 정보가 수집되었을 때 빅데이터 분석틀을 이용해 운전자에게 맞는 ECU-Mapping값을 추천하게 된다.

7번은 운전자는 스마트폰으로 서버에 접속하여 데이터를 받은 후 8번 과정에서 스마트폰이나 컴퓨터를 통해 ECU-Mapping 수치를 확인 후 에디슨 보드에 전송하고 ECU는 수치를 조절하게 된다. 이후 ECU는 엔진 수치를 조절한 뒤 운행하게 된다.

본 논문에서 소개하는 시스템 흐름도는 그림 4와 같다. 시스템은 운행 파트와 빅데이터 파트로 구성된다.

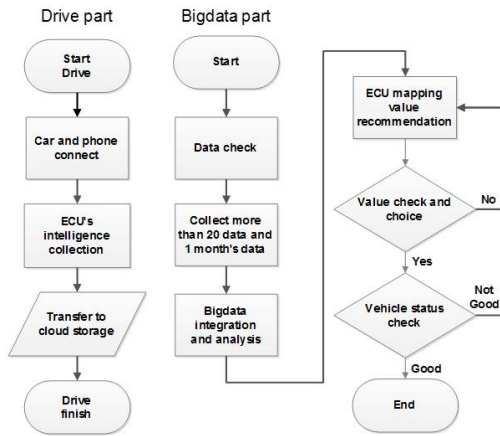


그림 4. 시스템 순서도
Fig. 4 System flowchart

운행이 시작되면 스마트폰과 차량이 블루투스를 통해 연결이 되고 RPM, 토크, 연료량, 엔진 연료 압축비 및 온도 등 ECU의 모든 정보가 운행이 끝날 때까지 수집이 되며 동시에 클라우드 스토리지에 전송이 된다.

클라우드 스토리지에 차량의 운행 정보가 20개 이상이나 한 달 이상의 정보가 축적되면 빅데이터 분석 툴을 사용하여 운전자에게 가장 적합한 ECU-Mapping 수치를 추천한다.

확인 후 에디슨 보드를 통해 ECU로 수치를 전송하여 ECU-Mapping을 하고 차량의 상태 확인을 한다. 이때, ECU-Mapping 수치가 이상이 있다고 판단될 경우 ECU에서 제공하는 정보를 통해 바로 확인이 가능하다.

ECU가 Mapping 수치에 이상이 있다고 판단되는 상황은 미리 설정해둔 배기가스 배출 수치가 초과, 부품 별 압축비가 사양 이상 설정된 경우이다. 이상이 생길 경우 설정 이전 초기 수치로 변경된다.

그림 5은 빅데이터 분석에 필요한 데이터 수집을 위한 프로그램 소스의 일부이다. 그림 5 (a)의 53번 줄 이하 소스는 차량의 마력, 토크, 연료 소모량 등을 확인하는 에디슨 보드에 삽입되는 프로그램 소스의 일부이며, 그림 5 (b)의 19번 줄 이하의 블루투스를 통해 스마트폰으로 전송하는 프로그램의 소스 일부이다.

그림 6은 클라우드 스토리지로 전송하는 스마트폰의 애플리케이션 소스의 일부이다.

인텔의 에디슨 보드는 전용 클라우드 스토리지인 “dashboard.us.enableiot.com”을 무료로 사용할 수 있으며 서버에 데이터가 수집된 후 빅데이터 분석

```

50 "event": "ECU_MAPPING_DATA",
51 "requestType": "POST",
52 "json": {
53   "data": {"Horse_Power": "{{Horse_Power}}", "torque":
54     "{{torque}}", "Ffuel_Less": "{{Fuel_Less}}"}
55   "ssid": "Car_ECU_Infor",
56   "type": "message"
57 },
58 "headers": {
59   "Content-Type": "application/json",
60   "Authorization": "Bearer __ARTIK_DEVICE_TOKEN__"
61 },
62 "noDefaults": true
    
```

(a)

```

19 int blue5=2; //Send setting
20 int blue6=3; //get setting
21 SoftwareSerial mySerial(blue5, blue6);
22
23 void setup()
24 {
25   Serial.begin(9600); //Serial monitor
26   mySerial.begin(9600); //bluetooth serial
27 }
    
```

(b)

그림 5. 차량 정보 수집 장치의 프로그램 소스
Fig. 5 Car information collecting device to program sources

```

172 |String cmd = "AT+CIPSTART=\"TCP\",\"";
173 cmd += "url=dashboard.us.enableiot.com"; // server IP
174 cmd += "\",80"; // port: 80
175 ser.println(cmd);
176
177 if(ser.find("Error")){
178   Serial.println("AT+CIPSTART error");
179   return;
180 }
181
182 //String, Data setting for GET method
183 String getStr = "GET /update?api_key=";
184 getStr += apiKey;
185 getStr += "&field1=";
186 getStr += String(strTemp);
187 getStr += "\r\n\r\n";
188
189 // Send Data
190 cmd = "AT+CIPSEND=";
191 cmd += String(getStr.length());
192 ser.println(cmd);
    
```

그림 6. 클라우드 전송 프로그램 소스
Fig. 6 Cloud transfer program source

을 통해 ECU-Mapping 수치를 추천받게 된다.

빅데이터 분석은 빅데이터 오픈 소스 프로젝트인 아파치 소프트웨어의 Spark를 통해 관리 및 분석하였다. 개방형이기 때문에 누구나 사용가능하며 안정성과 확장성이 뛰어나기 때문에 빅데이터 분석에 많이 사용되는 툴이다.

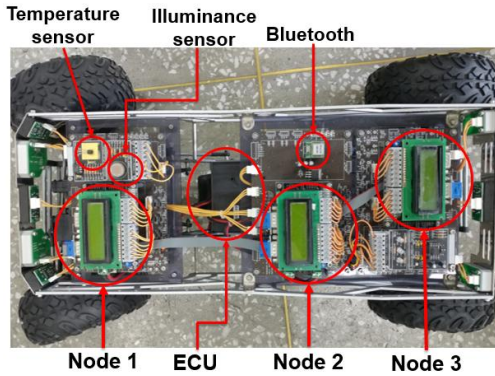


그림 7. 테스트 베드용 차량
Fig. 7 Test bed car

173, 174번 줄과 같이 무료로 제공되는 에드슨 보드의 클라우드 스토리지의 주소와 포트를 입력한다. 이후 182번 줄 이후부터 GET 방식으로 데이터를 수집하기 위해 String을 이용하여 데이터를 서버로 전송하게 된다.

이 프로그램들을 통해 마력, 토크 등 ECU-Mapping과 관련된 정보는 블루투스를 통해 스마트폰에 전송된 후 클라우드 스토리지로 다시 전송되어 저장된다.

수집된 데이터는 각 자동차의 엔진 배기량과 최초의 ECU 수치를 기준으로 하여 데이터 분석을 하게 된다. 마력과 토크를 상승시키면, 연료 소모는 거의 동일하게 유지를 하고 고속, 저속 등 운전 습관과 운행 환경에 따른 상황들을 고려하여 추천하게 된다.

그래서 변경되는 수치는 자동차 구동에 필요한 마력, 토크 및 엔진 연료 압축비이다.

1. 테스트 베드용 차량 제작

그림 7처럼 데이터를 실시간으로 획득해서 데이터베이스를 확보하고 ECU-Mapping을 조정하는 것이 유의미한가에 대한 테스트 베드용 차량을 제작하였다.

각 노드와 센서, 블루투스 모듈은 에드슨 보드를 기초로 조립하였다. 노드 1은 전조등, 전방 초음파 센서, 온도 센서, 조도 센서 제어, 노드 2는 모든 노드의 정보를 수집, 유무선 통신 제어, 모터 제어, 노드 3은 테일램프, 추방 초음파 센서 등을 제어하게 된다. 테스트 베드용 차량의 모터는 총 6개이며, 4개는 바퀴에 연결되어 있으며 나머지 2개는 에어컨과 엔진의 역할을 대신 하게 된다.

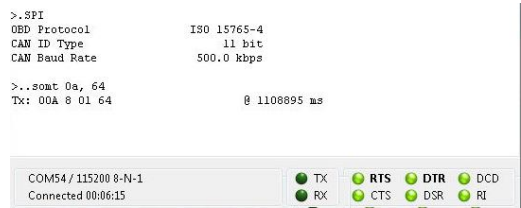


그림 8. ECUsim 프로그램
Fig. 8 ECUsim program

실제 차량과 동일한 방식으로 사용되는 시뮬레이터인 ECUsim을 테스트 베드용 차량에 탑재하여 데이터 수집 및 맵핑을 하여 테스트를 하였다.

Scantool에서 제작한 ECUsim은 오픈마켓에서 쉽게 구매할 수 있으며 실제 차량에서 수집이 가능한 데이터 중 순간 연비, 평균 연비 등과 같은 수식이 필요한 것을 제외한 단순 데이터는 동일하게 설정 및 수집이 가능하다. 그리고 확장성이 뛰어나 에드슨 보드와 같은 오픈소스 H/W와의 연결이 가능하다 [9, 10].

또한, ECUsim은 간단한 명령으로 운용이 가능하며 CAN 프로토콜도 완벽하게 지원한다. 데이터 전송 속도는 500kbps이며 마력, 토크, 엔진 연료 압축비 등 설정을 할 때 16진수로 변경하여 설정한다.

예를 들어 그림 8과 같이 엔진 연료 압축비를 100kPa로 변경하려면 데이터를 전송하는 명령어인 “somt“를 입력하고 100을 16진수로 바꾸어 64를 입력하면 된다 [11].

2. 테스트 베드용 차량 실험 및 결과

처음 ECU 설정은 장거리 고속 주행 50%와 단거리 저속 주행 50%로 분석된 수치로 설정하였다. 그리고 실험은 20일 동안 40번 주행을 2개의 주행 환경으로 나누어 진행하였다.

본 논문의 주행 조건은 빅데이터의 분석을 위한 충분 조건이 아닌 최소 조건이다. 또한, 실제 본 논문의 시스템이 정상적으로 동작하는지 확인하는 절차이며 실제 차량에서는 더 많은 데이터와 긴 기간 동안 수집하여 좀 더 정확한 ECU-Mapping 수치를 추천받을 수 있다.

첫 번째 환경은 장거리와 고속 주행 80% 단거리와 저속 20%, 두 번째 환경은 단거리와 저속 주행 80% 장거리 고속 주행 20%로 설정하여 실험하였다. 맵 데이터 측정은 ECUsim에서 제공하는 툴을 사용하였다.

		Engine Load (%)									
		10	20	30	40	50	60	70	80	90	100
E n g i n e	800.0	20.3	20.3	20.3	20.3	20.3	20.3	20.0	20.0	20.0	19.8
	1200.0	20.3	20.3	20.3	20.3	20.3	20.3	19.8	19.7	19.7	19.8
	1600.0	20.3	20.3	20.3	20.3	20.3	20.3	19.8	19.7	19.7	19.5
	2000.0	20.3	20.3	20.3	20.3	20.1	20.1	19.8	19.5	19.5	19.5
	2400.0	20.3	20.3	20.3	20.3	20.1	20.1	19.8	19.5	19.5	19.1
	2800.0	20.3	20.3	20.3	20.3	20.1	20.1	19.7	19.1	19.1	19.1
	3200.0	20.3	20.3	20.3	20.3	20.1	20.1	19.7	19.1	19.1	18.8
	3600.0	20.3	20.3	20.3	20.3	20.1	19.8	19.5	19.1	18.8	18.8
	4000.0	20.3	20.3	20.1	20.1	20.1	19.8	19.5	19.1	18.8	18.8
	4400.0	20.3	20.3	20.1	20.1	19.8	19.8	19.1	18.8	18.8	18.5
S p e e d (R P M)	4800.0	20.3	20.3	20.1	20.1	19.8	19.8	19.1	18.8	18.5	18.5
	5200.0	20.3	20.1	20.1	20.1	19.8	19.7	19.1	18.8	18.5	18.5
	5600.0	20.1	20.1	19.8	19.8	19.8	19.7	19.1	18.8	18.5	18.5
	6000.0	20.1	20.1	19.8	19.8	19.7	19.5	18.8	18.5	18.5	18.5
	6400.0	20.1	19.8	19.8	19.8	19.7	19.5	18.8	18.5	18.5	18.5

그림 9. 최초 ECU 맵 데이터
Fig. 9 The initial ECU map data

		Engine Load (%)										
		10	20	30	40	50	60	70	80	90	100	
E n g i n e	800.0	20.3	20.3	20.3	20.3	20.3	20.3	20.0	20.0	20.0	19.8	
	1200.0	20.3	20.3	20.3	20.3	20.3	20.3	19.8	19.7	19.7	19.8	
	1600.0	20.3	20.3	20.3	20.3	20.3	20.3	19.8	19.7	19.7	19.5	
	2000.0	20.3	20.3	20.3	20.3	20.1	20.1	19.5	19.5	19.5	19.5	
	2400.0	20.3	20.3	20.3	20.3	20.1	20.1	19.5	19.5	19.5	19.1	
	2800.0	20.3	20.3	20.3	20.1	20.1	19.7	19.1	18.8	18.8	18.8	
	3200.0	20.3	20.3	20.3	20.3	20.3	20.3	19.8	19.5	19.1	18.8	18.8
	3600.0	20.3	20.3	20.3	20.1	19.8	19.5	19.1	18.8	18.8	18.8	
	4000.0	20.3	20.3	20.1	19.8	19.8	19.1	18.8	18.8	18.5	18.2	
	4400.0	20.3	20.3	20.1	19.8	19.8	19.1	18.8	18.5	18.5	18.2	
S p e e d (R P M)	4800.0	20.3	20.3	20.1	19.8	19.7	19.1	18.8	18.5	18.2	18.2	
	5200.0	20.3	20.1	20.1	19.8	19.7	19.1	18.8	18.5	18.2	18.2	
	5600.0	20.1	20.1	19.8	19.7	19.5	19.1	18.5	18.5	18.2	18.2	
	6000.0	20.1	20.1	19.8	19.7	19.5	19.1	18.5	18.2	18.2	18.2	
	6400.0	20.1	19.8	19.8	19.7	19.5	18.8	18.8	18.2	18.2	18.2	

그림 10. 첫 번째 주행 환경의 맵 데이터
Fig. 10 Map data of first drive environment

테스트베드용 차량의 경우 엔진출력은 전기모터가 담당하고, 실제 자동차의 경우 가솔린엔진을 사용하고 있다. 그러나 ECUSim은 실제 자동차에서 사용하는 RPM과 엔진 부하를 중심으로 엔진 연료 압축비를 결정하는 설정을 동일하게 사용하기 때문에 단순 실험을 위한 환경에서는 큰 차이가 없다.

하지만 테스트 베드용 차량의 모터는 실제 자동차의 엔진과 큰 차이가 나기 때문에 맵 데이터의 수치를 10배 높게 나타내게 하였다. 단순히 단위만 변경하였기 때문에 결과와는 무관하다. 그리고 맵 데이터를 편의상 최초의 환경을 0번, 첫 번째 주행 환경은 1번, 두 번째 주행 환경을 2번으로 설정한다.

테스트 베드용 차량의 ECU-Mapping은 변화폭을 확인하기 위해 마력, 토크 등의 한계치를 설정하지

		Engine Load (%)									
		10	20	30	40	50	60	70	80	90	100
E n g i n e	800.0	20.5	20.5	20.3	20.3	20.3	20.1	20.0	20.0	20.0	19.8
	1200.0	20.5	20.3	20.3	20.3	20.1	20.1	19.8	19.7	19.7	19.8
	1600.0	20.3	20.3	20.3	20.1	20.1	20.1	19.8	19.7	19.7	19.5
	2000.0	20.3	20.3	20.1	20.1	20.1	20.1	19.8	19.5	19.5	19.5
	2400.0	20.1	20.1	20.1	20.1	20.1	20.1	19.8	19.5	19.5	19.1
	2800.0	20.1	20.1	20.1	20.1	20.1	20.1	19.7	19.1	19.1	19.1
	3200.0	20.1	20.1	20.1	20.1	20.1	20.1	19.7	19.1	19.1	18.8
	3600.0	20.1	20.1	20.1	20.1	20.1	19.8	19.5	19.1	18.8	18.8
	4000.0	20.1	20.1	20.1	20.1	20.1	19.8	19.5	19.1	18.8	18.8
	4400.0	20.1	20.1	20.1	20.1	19.8	19.8	19.1	18.8	18.8	18.5
S p e e d (R P M)	4800.0	20.1	20.1	20.1	20.1	19.8	19.8	19.1	18.8	18.5	18.5
	5200.0	20.1	20.1	20.1	20.1	19.8	19.7	19.1	18.8	18.5	18.5
	5600.0	20.1	20.1	19.8	19.8	19.8	19.8	19.1	18.8	18.5	18.5
	6000.0	20.1	20.1	19.8	19.8	19.7	19.5	18.8	18.5	18.5	18.5
	6400.0	20.1	19.8	19.8	19.8	19.7	19.5	18.8	18.5	18.5	18.5

그림 11. 두 번째 주행 환경의 맵 데이터
Fig. 11 Map data of second drive environment

않고 실험하였다.

그림 9은 ECUSim의 최초 설정을 분석한 맵 데이터이다. 0번 환경에서 엔진의 최대 엔진 연료 압축비는 20.3bar 이며, 최저는 18.5bar로 나타났다.

그림 10는 1번 환경의 빅데이터를 기초로 주행 후 빅데이터를 분석한 뒤 설정한 맵 데이터이다. 그림 9와 비교하였을 때 저속 구간은 동일한 엔진 연료 압축비를 보이고 있다.

하지만 고속 운전 구간을 살펴봤을 때 압축비가 더 낮게 나타난 것을 볼 수 있다. 예를 들어 RPM 6400, 엔진 부하 100% 구간을 비교할 때 0번 환경은 18.0bar이며 1번 환경은 18.2bar로 설정되어 있다.

18.2bar의 엔진 연료 압축비로 설정하더라도 기존과 같은 성능을 나타낼 수 있다는 것을 확인할 수 있다.

그림 11은 두 번째 주행 환경에서 저장된 데이터베이스 중 실시간으로 나타나는 빈도를 기초로 ECU 맵 데이터를 변경하여 설정하였다.

그림 9와 비교하였을 때 높은 RPM과 엔진 부하 영역은 동일한 수치로 설정된 것을 확인할 수 있다. 하지만 저속구간은 크게 차이가 나타난다.

자동차가 정차 후 출발을 할 때 연료를 가장 많이 소모를 한다. 그렇기 때문에 RPM 800, 엔진 부하 10%는 20.5bar로 늘어났으며 저속 운행을 시작하였을 때 대부분 20.1bar로 바뀐 것을 확인할 수 있다.

실제 차량이 출고되었을 때는 모든 운전자의 운전 습관과 운행 환경을 고려할 수 없기 때문에 0번 운행 환경을 기준으로 설정을 한다. 그렇게 될 경우

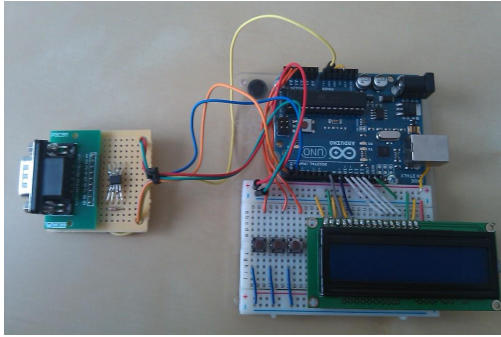


그림 12. 차량 정보 수집 장치

Fig. 12 Car information collecting device

각 운전자는 차량의 성능을 제대로 사용할 수 없으며 연비에도 영향을 줄 수밖에 없다.

테스트 베드용 차량을 통해 확인할 결과 빅데이터를 분석하여 운전 습관에 따라 ECU-Mapping을 할 수 있는 것으로 확인되었다.

3. 실제 차량 실험 및 결과

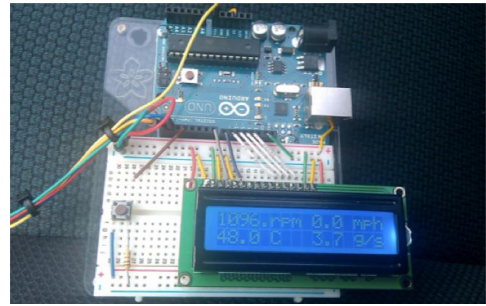
테스트 베드용 차량의 실험 결과가 성공적으로 나왔기 때문에 실제 차량에 탑재하여 실험하였다. 2014년식 엑센트 1.6 가솔린 모델을 사용하였으며 차량의 ECU와의 통신을 위해 본 논문의 시스템에 그림 12과 같이 제작하여 차량의 ECU에 연결하였다.

실험은 고속 주행 80%, 저속 주행 20% 위주로 운행하였으며 2017년 2월 5일부터 동년 3월 4일까지 20번 주행을 통해 ECU가 수집 가능한 모든 데이터를 수집하는 동시에 클라우드 스토리지에 전송하여 빅데이터를 생성하였다.

본 논문에서는 모든 정보 중 실제 본 논문의 시스템의 작동 여부를 확인하기 위해 ECU-Mapping에 기본이 되는 마력, 토크, 엔진 연료 압축비, 엔진 부하를 중심으로 설계하였다.

차량의 사양과 법적인 허용수치를 계산하여 마력, 토크, 연료 압축 등 한계치를 설정하고 빅데이터 분석을 통해 운전 습관에 맞는 ECU-Mapping 수치를 추천한 뒤 운전자의 선택에 의해 차량에 적용한다.

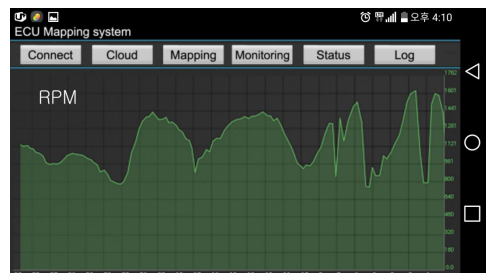
그림 13 (a)와 같이 실시간으로 차량의 정보를 확인할 수 있으며 그림 13 (b)와 같이 스마트폰으로 ECU에서 전송하는 수치들을 실시간 모니터링하거나 그림 13 (c)와 같이 각 항목을 선택하여 자세히 확인할 수 있다. 또한, 모니터링과 동시에 휴대



(a)



(b)



(c)

그림 13. 차량의 데이터 수집 및 모니터링

Fig. 13 Data collection and monitoring of car

전화 무선통신망을 통해 클라우드 스토리지로 전송한다.

모든 운행과 관련된 ECU 수치들이 20개 이상 혹은 한 달 이상 데이터가 수집되었을 때 빅데이터 분석 툴을 통해 마력, 토크 및 엔진 연료 압축비를 추천하게 된다.

운전자의 동의에 따라 ECU-Mapping을 하게 되면

		Speed	Revs	Fuel Rail	Av.CO2	Load	Intake
			rpm	psi	g/km	%	°C
	Gap	0	9.0	-1737.0			2.0
Real Value	1 Gear	6	763.0	4511.0	215.0	28.4	17.0
Set Value	No speed	6	756.8	4584.0	173.2	30.0	19.0
	Gap	0	6.2	-73	41.8	-1.8	-2
Real Value	2 Gear	12	770.8	4593.2	215.0	31.4	17.0
Set Value	No speed	12	781.3	4730.5	173.2	30.6	18.0
	Gap	0	-10.5	-137.3	41.8	0.8	-1
Real Value	1 Gear	6	788.3	4543.4	214.9	24.8	18.0
Set Value	Uphill	6	791.0	4781.0	173.2	35.1	21.0
	Gap	0	-2.7	-237.6	41.7	-10.3	-3
Real Value	2 Gear	12	771.5	5874.0	213.9	54.2	19.0
Set Value	Uphill	12	798.7	5497.1	172.2	70.3	21.0
	Gap	0	-27.2	376.9	41.7	-18.1	-2

그림 14. ECU의 항목 별 검사
Fig. 14 Inspection item of ECU

		Engine Load (%)										
		13.7	20.0	26.3	32.5	38.9	45.2	51.5	57.8	64.1	70.4	76.7
E n g i n e	800.0	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7	13.6	13.6	13.6
	1200.0	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7	13.6	13.6	13.6
	1600.0	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7	13.6	13.6	13.6
	2000.0	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7	13.6	13.6	13.6
	2400.0	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7	13.6	13.6	13.6
	2800.0	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7	13.6	13.3	13.3
	3200.0	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7	13.3	12.6	12.9
	3600.0	14.7	14.7	14.7	14.7	14.7	14.7	13.3	13.3	12.9	12.6	12.4
	4000.0	14.7	14.7	14.7	14.7	14.7	14.7	13.3	12.6	12.2	12.2	12.2
	4400.0	14.7	14.7	14.7	14.7	14.7	14.7	13.1	12.6	12.1	11.8	11.8
(R P M)	4800.0	14.7	14.7	14.7	14.7	13.6	13.6	12.5	12.3	11.8	11.5	11.5
	5200.0	14.7	14.7	14.7	14.7	13.3	12.7	12.2	11.9	11.5	11.5	11.5
	5600.0	14.7	14.7	14.7	13.6	13.1	12.3	12.0	11.8	11.5	11.5	11.5
	6000.0	14.7	14.7	13.6	13.3	12.8	11.8	11.5	11.5	11.5	11.5	11.5
	6400.0	14.7	14.7	13.4	12.7	12.3	11.5	11.5	11.5	11.5	11.5	11.5
	6800.0	13.4	13.3	13.3	12.1	12.0	11.5	11.5	11.5	11.5	11.5	11.5

그림 15. ECU-Mapping 이전 맵 데이터
Fig. 15 Map data before ECU-Mapping

각 수치는 변경이 된다. 이후, 설정된 수치를 기준으로 시동을 켜 놓은 상태로 두면 ECU는 각 수치들을 계산하여 그림 14와 같이 항목별로 확인할 수 있다. 그림은 일부 항목만 나타났지만 표 1에 나타는 항목 대부분을 확인할 수 있다.

ECU에서 단순히 계산하는 수치이기 때문에 실제 값과는 차이가 생길 수 있지만 이것을 기준으로 차량의 기본적인 이상 유무를 판단할 수 있다.

그림 15는 실험을 진행하기 전 차량의 맵 데이터를 측정된 표이다. 정확한 측정을 위해 테스트 베드용 차량과는 달리 자동차 정비소에서 확인하였다.

실제 차량은 수치상 최대 RPM 8000까지 상승시킬 수 있으나 일반적인 운행으로는 불가능하기 때문에 RPM 6800와 엔진 부하 76.7%까지의 수치만 확인하였다.

그림 16는 실험을 진행한 뒤 ECU-Mapping 이후 측정된 맵 데이터이다.

실험은 고속 위주의 운행을 했기 때문에 낮은 RPM과 엔진 부하 구간은 동일한 수치의 엔진 연료

		Engine Load (%)										
		13.7	20.0	26.3	32.5	38.9	45.2	51.5	57.8	64.1	70.4	76.7
E n g i n e	800.0	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7	13.6	13.6
	1200.0	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7	13.6	13.6
	1600.0	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7	13.8	13.2
	2000.0	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7	13.8	13.2
	2400.0	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7	13.7	13.3	12.5
	2800.0	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7	13.7	13.3	12.3
	3200.0	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7	13.7	12.6	12.3
	3600.0	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7	13.3	12.6	12.3
	4000.0	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7	13.3	12.6	11.8
	4400.0	14.7	14.7	14.7	14.7	14.7	14.7	14.7	13.1	12.3	11.8	11.2
(R P M)	4800.0	14.7	14.7	14.7	14.7	13.6	13.6	12.5	12.3	11.8	11.5	11.5
	5200.0	14.7	14.7	14.7	14.7	13.3	12.7	12.2	11.5	10.9	10.9	10.9
	5600.0	14.7	14.7	14.7	13.6	13.1	12.3	12.0	11.2	10.9	10.9	10.9
	6000.0	14.7	14.7	13.6	13.3	12.8	11.8	11.5	11.5	10.9	10.9	10.9
	6400.0	14.7	14.7	13.4	12.7	12.3	11.5	11.2	10.9	10.9	10.9	10.9
	6800.0	13.4	13.3	13.3	12.1	12.0	11.2	10.9	10.9	10.9	10.9	10.9

그림 16. ECU-Mapping 이후 맵 데이터
Fig. 16 Map data after ECU-Mapping

압축비로 설정된 것을 확인할 수 있다.

낮은 RPM과 엔진 부하 구간에서는 기존의 ECU 수치 변경 없이도 성능을 제대로 발휘할 수 있다고 판단한 것이다.

실제 차량이 운행될 때 고속 주행보다 저속 주행 시 연료가 더 많이 소모가 된다. 그 이유는 고속 주행 시 가속도로 인한 속도 유지와 급속한 토크변화가 없기 때문에 일정한 엔진 연료 압축비로 운행이 가능하지만 저속 주행 혹은 정차 후 출발할 때 차량 전체를 움직여야하기 때문에 순간적인 토크를 높이기 위해 연료를 많이 사용한다. 또한, 차량이 연료를 이용해 운동에너지와 가속으로 주행 중이던 브레이크를 밟을 경우 차량이 정차하기 때문에 이전에 모든 에너지는 사라지게 되므로 연료가 많이 소모된다 [12].

하지만 높은 RPM과 엔진 연료 압축비 구간을 비교하면 기존보다 낮은 수치의 엔진 연료 압축비가 설정되어 있는 것을 확인할 수 있다.

높은 RPM과 엔진 연료 압축비를 사용할 때 엔진 연료 압축비를 기존보다 낮게 설정하더라도 엔진을 기존과 동일하거나 향상된 성능으로 사용할 수 있다고 판단하였기 때문이다.

그래서 ECU-Mapping 후 마력, 토크, 연료 소모의 변화를 비교하기 위해 그림 17과 같이 다이내모 그래프로 측정하였다. 001.drf는 차량의 최초 설정된 수치를 측정된 것이며, 002.drf는 ECU-Mapping을 실행한 뒤 측정된 그래프이다.

다이내모 그래프에서 확인할 결과 본 논문의 시스템을 통해 엔진 연료 압축비 뿐 아니라 마력은 129.41HP에서 137.24HP로, 토크는 16.75kg.m에서 17.27kg.m로 상승시켜 사양을 높였다.

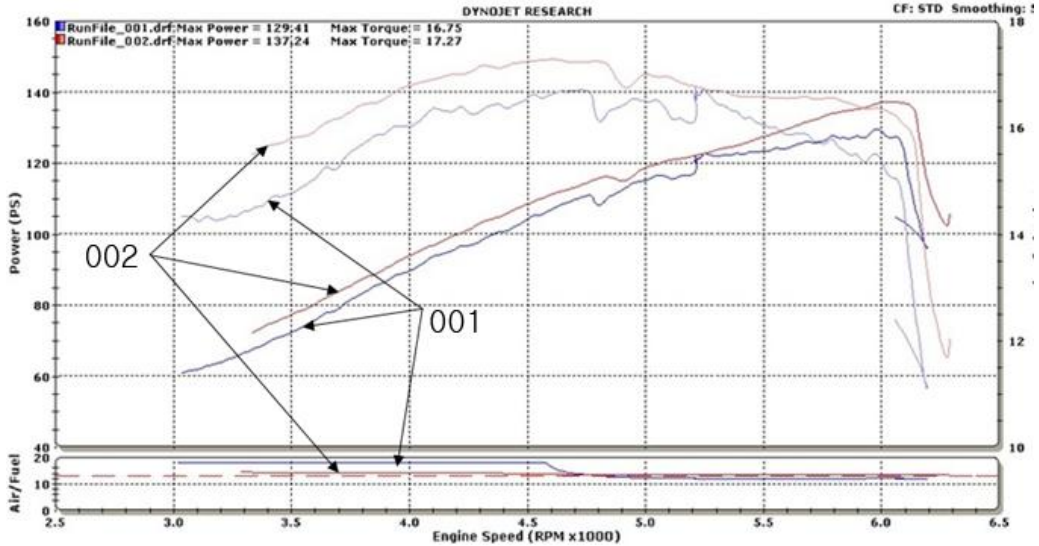


그림 17. ECU 맵 데이터와 다이노모 그래프 상의 실험 결과
 Fig. 17 ECU map data and result of test on dynamo graph

단순히 마력과 토크의 상승이 뿐만 아니라 연료 소모량을 확인하였을 때 저속 구간에서는 ECU-Mapping 이후가 더 많이 소모되는 것으로 확인되었지만 고속으로 갈수록 비슷한 소모를 나타내는 것으로 확인되었다.

그 이유는 ECU-Mapping 이 후 상승된 토크와 마력을 사용하기 때문에 토크의 변화가 심한 저속 구간에서는 오히려 연료 소모가 많지만 고속 위주의 운행을 할 경우에는 더 높은 토크와 마력을 사용함에도 연료는 ECU-Mapping 이전과 거의 동일하게 사용한다는 것을 확인하였다. 그것은 고속 주행시 연료 소모가 효율적으로 되었다는 것을 나타낸다.

또한, 10번의 각기 다른 ECU-Mapping 수치로 실험한 결과 배기가스, CO₂ 등 위법한 수치는 한 번도 나타나지 않았다. 그래서 배기가스, CO₂의 수치를 규정보다 더 낮게 설정한 후 실험하였다.

그림 18처럼 ECU-Mapping을 유지할 경우 차량에 문제가 생길 수 있다는 경고와 함께 최초의 ECU 수치로 되돌리게 된다. 또한, 각 수치를 확인하려면 클라우드 스토리지에 접속하면 그림 14에서 Ac CO₂ 항목이 설정한 값보다 높게 배출되는 것을 확인할 수 있다. 이 결과는 단순 이론적인 수치로 확인한 것이기 때문에 실제 차량 검사를 할 경우 다른 결과로 나올 수 있다.

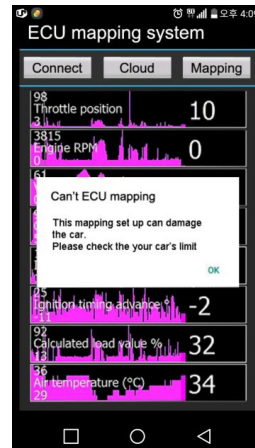


그림 18. ECU-Mapping 실패
 Fig. 18 ECU-Mapping failed

하지만 엔진에 타르, 찌꺼기 등을 제거하고 엔진 오일 교체를 잘 지킬 경우 비슷한 결과로 나타난다. 실제 엔진에 연료가 주입될 때 엔진 매가 많을 경우 엔진 연료 압축비보다 낮게 나타나며 그럴 경우 연료의 불완전 연소로 인해 배기가스, CO₂ 등이 더 많이 배출된다 [13].

실제 차량을 통해 실험한 결과 운전자의 운전 습관과 운전 환경을 통해 ECU 정보를 수집하여 빅데이터로 구성한 뒤 자료를 분석하여 ECU-Mapping

을 실행한 결과 마력과 토크는 상승시키지만 연료 소모는 기존과 비슷하게 소모한다는 것을 확인하였다.

IV. 결 론

본 논문은 각 운전자의 운전 습관을 정해진 조건을 만족하는 주행 내용만 선택하여 일정 기간 클라우드로 스토리지에 전송한 뒤 ECU의 모든 정보를 데이터베이스화하고 일정기간의 운행습관을 분석한 뒤 가장 적합한 ECU 수치를 분석하여 차량에 적용하는 시스템을 제안하였다.

기존 ECU-Mapping은 ECU 튜닝 매장에서 일반적으로 정해주는 수치를 통해 설정을 하였다. 그로인해 각 운전자의 운전습관과 운행환경에 맞지 않은 설정으로 단순히 마력과 토크만 높이는 비효율적인 튜닝이다.

이러한 튜닝은 연료의 비효율적인 연소로 인해 환경문제도 일어날 수 있으며 엔진에 연료 찌꺼기, 타르 등이 생성되어 ECU에서 명령하는 엔진 연료 압축이 제대로 이루어지지 않을 수도 있다. 그로인해 차량의 성능을 제대로 발휘하거나 연비의 최적화를 얻기 어렵다.

하지만 본 논문의 시스템을 통해 ECU 맵 데이터를 구성하는 마력, 토크 및 연료 압축 등을 모든 운전자가 각각 자신의 운전 습관과 환경에 맞춰 ECU-Mapping을 수행을 한다면 차량의 성능이 개선될 가능성을 보여주었다.

또한, 주행 데이터를 1년 치를 수집하여 데이터가 더 방대해지면 더 방대해진다면 월별, 분기별, 계절별로 ECU-Mapping이 가능하다.

현재 토크, 마력, 엔진 연료 압축비를 기준으로 ECU-Mapping을 하여 차량의 성능을 높였지만 향후 운습도, 자이로센서, 흡기 압력 및 가속 페달 각 등 ECU가 수집할 수 있는 모든 데이터를 기준으로 차량의 성능을 향상 시킬 수 있는 빅데이터 분석방법을 지속적으로 연구하고 있다.

Reference

- [1] Ministry of Land, Infrastructure and Transport, "http://www.lawnb.com/data/Focuslawdata/lawnbfocusB00062051651.pdf"
- [2] J. Park, S. Lee, K. Lee, "A Study on FIBEX Automatic Generation Algorithm for FlexRay Network System," IEMEK J. Embed. Sys. Appl., Vol. 8, No. 2, pp. 69-78, 2013 (in Korean).
- [3] C. Herber, A. Richter, T. Wild, A. Herkersdorf, "A Network Virtualization Approach for Performance Isolation in Controller Area Network (CAN)," Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium, pp. 215-224, 2014.
- [4] D. Kum, J. Hong, S. Jin, J. Cho, "Testing System for Automotive Software Using a General Purpose Development Board," Vol. 7, No. 1, pp. 17-24, 2012 (in Korean).
- [5] <https://software.intel.com/en-us/iot/hardware/edison>
- [6] J.B. Kim, "Understand of OSHW Based on OSHWA," OSIA Standards & Technology Review, Vol. 28, No. 1, pp. 8-24, 2015 (in Korean).
- [7] W. Lee, I. Ohm, "Effects of the Method of Changing Compression Ratio on Engine Performance in an SI Engine," Transaction of the Korean Society of Automotive Engineers, Vol. 9, No. 4, pp. 27-33, 2001.
- [8] A. Broring, A. Remke, C. Stasch, C. Autermann, M. Rieke, J. Mollers, "enviroCar: A Citizen Science Platform for Analyzing and Mapping Crowd-Sourced Car Sensor Data," Transactions in GIS, Vol. 19, No. 3, pp. 339-492, 2015.
- [9] J. C. Carnes, "Ecosim: An Engine Control Unit Simulator Used at Ford," SAE Technical Paper, No. 2006-01-1601, 2006.
- [10] J.H. Kim, S. Kim, "Autonomous-flight Drone Algorithm use Computer Vision and GPS," IEMEK J. Embed. Sys. Appl., Vol. 11, No. 3, pp. 193-200, 2016 (in Korean).
- [11] S. Kristian, M. Jeffrey, "OBDII Data Logger Design for Large-scale Deployments," Proceedings of the International IEEE Conference on Intelligent Transportation Systems, pp. 670-674, 2013.
- [12] K. Ahn, H. Rakha, A. Trani, M. V. Aerde, "Estimating Vehicle Fuel Consumption and Emissions based on Instantaneous Speed and Acceleration Levels," Journal of

Transportation Engineering, Vol. 128, No. 2, 2002.

- [13] A. Tsolakisa, A. Megaritisb, M.L. Wyszynskia, K. Theinnoia, "Engine Performance and Emissions of a Diesel

Engine Operating on Diesel-RME (Rapeseed Methyl Ester) Blends With EGR (Exhaust gas Recirculation)," Energy, Vol. 32, No. 11, pp. 2072 - 2080, 2007.

Shik Kim (김 식)



He received B.E degree in Dept. of Electronics, Kyungbuk Univ, in 1979 and M.S degree in Dept of information and communication Okayama prefectural University in 2004. He was a senior researcher in Agency of Defence Development(ADD) in Korea and a researcher staff in Texas Transportation Instute(TTI) in Texas, USA. He is currently a full professor in the school of information and communication in Semyung university. His research interest the embedded platform, the convergence of IT with automobile and drone

Email: shikm@semyung.ac.kr

Junghwan Kim (김 정 환)



He received the B.S. and M.S. degree in Information and Communication system from Semyung University. He is currently working toward Ph.D degree at Semyung university. His research interests include embedded system.

Email: jhwem82@gmail.com