

Motohawk ECU에 의한 가솔린기관의 연료분사제어 연구

이동원*, 조정권**

*아주자동차대학 자동차계열

**교통안전공단

e-mail:tongwon7@motor.ac.kr

Motohawk ECU에 의한 가솔린기관의 연료분사제어 연구

Tong-won Lee*, Jeong-kwon Jo**

*Dept of Automotive, Ajou Motor College

**Korea Transportation Safety Authority

요 약

Motohawk ECU(Engine Control Unit)에 의한 가솔린기관의 연료분사제어 알고리즘 연구를 통하여 연비를 절감할 수 있는 방안을 연구한다. 실제 차량엔진 ECU는 캘리브레이션 이외의 제어 알고리즘을 변경하여 구현하기가 쉽지 않으므로 상기의 프로토타입 ECU를 활용하여 연료분사제어 로직을 변경해가면서 연료분사와 관련된 여러가지 파라미터의 연구가 가능하다. 또한 프로그래밍은 Matlab과 Simulink로 구현할 수 있어서 최적연비를 얻기 위한 로직의 구현을 실시간으로 할 수 있으며 차량의 특성상 프로그램의 임베딩이 가능하여야 하므로 이러한 목적에 프로토타입 ECU에 의한 로직개발은 많은 장점이 있다. 이러한 ECU를 활용하여 여러 가지 제어 변수에 의한 최적화된 연료분사 로직의 설계 및 미치는 영향을 평가한다.

1. 서론

MotoHawkTM 는 실제 생산 전자 제어 하드웨어에서 Simulink/Stateflow 모델을 사용하여 개발하고 설치된 제어 어플리케이션을 테스트하고 검증하는 제어 소프트웨어 개발 시스템이다. MotoHawkTM 는 제어 기능 개발, 차량 교정, 플릿 테스트를 위해 개발되었다. MotoHawk 는 MotoTron 의 Control-CoreTM 생산 소프트웨어 체제와 ECU 하드웨어 제품군에서 구성된다. MotoHawkTM 는 단일 컨트롤러와 분배형 전기신호 제어방식을 사용하여 다양한 어플리케이션을 지원하는 프로그램이다. 이를 활용하여 가솔린엔진을 구동할 수 있는 프로토타입 ECU를 구성하는 소프트웨어를 개발하여 이 ECU 장비에 탑재한다. 이를 통하여 프로그램의 제어로직을 수정할 수 있는 임베디드 ECU를 개발함을 목표로 한다.

2. 연구 수행의 배경

연구수행의 배경은 이러한 엔진 ECU 를 개발함으로써 모터스포츠 및 레이싱 시장과 튜닝 시장을 겨

냥하여 대응할 수 있는 기술개발을 달성하는 것이다. 기타의 시장으로는 선박 및 중장비의 엔진제어 장치 시장이다. 애프터 마켓시장을 대상으로 하여 선박 등의 기계식 디젤 엔진의 전자화에 적용이 가능하며, 모터스포츠 차량 및 엔진튜닝시장과 중장비 및 선박 디젤 엔진 제어기에 활용이 가능하다. 또한 엔진 개조시장에서 엔진을 개조시 필요한 제어로직 및 데이터의 수정이 용이하여 LPG, CNG 등으로의 엔진 개조 시에 중추적인 엔진제어기의 역할이 가능하다.

본 연구의 목표는 장기적으로는 독자적인 엔진제어기의 개발을 목표로 하며 프로그램의 로직 및 데이터의 수정이 용이한 제어기를 개발함을 궁극적인 목표로 한다.

3. 연구개발 방법 및 과정

3.1 Matlab 개요

MATLAB은 알고리즘 개발, 데이터 시각화, 데이터 분석 및 산술 연산을 위한 고성능 테크니컬 컴퓨팅 언어이자 대화형 환경이다. MATLAB을 사용하면 C, C++ 및 Fortran과 같은 기존 프로그래밍 언어보다 빠르게 테크니컬 컴퓨팅 문제를 해결할 수

있다. MATLAB은 신호 및 이미지 처리, 통신, 제어 설계, 테스트 및 평가, 금융 모델링 및 분석, 그리고 전산생물학 등 다양한 응용 분야에서 사용할 수 있다. 부가 도구상자(별도의 제품으로, 특수한 목적의 MATLAB 함수 집합)로 MATLAB 환경을 확장하여 해당 응용 분야에서 특정 수준의 문제를 해결한다.

3.2 Motohawk 소개

그림 1은 Motohawk에 의해 작성된 메인 엔진 제어 프로그램 화면이다. Motohawk ECU의 사양은 다음과 같다.

- Microprocessor: Motorola MPC565, 56MHz
- Memory: 1M Flash, 36K RAM (Optional 512K), 8K
- Serial EEPROM(Optional 64Kx8
- Parallel EEPROM)
- Operating Voltage: 6-32VDC
- Operating Temperature: -40°C to 105°C
- Sealed connectors operable to 10ft submerged

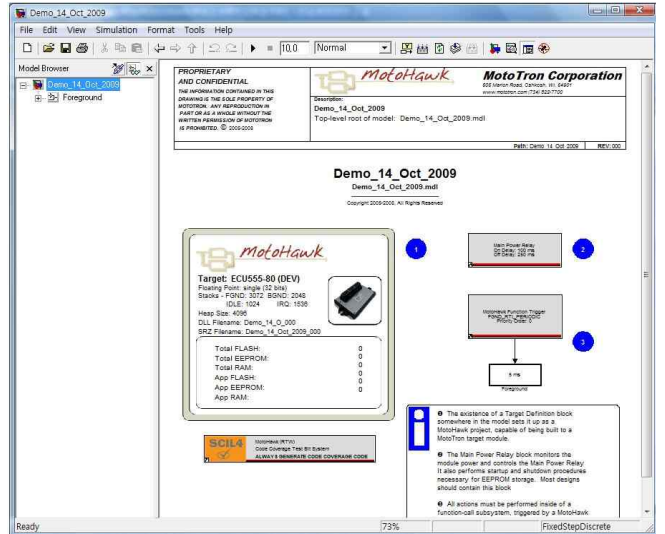
Inputs:

- Up to 34 Analog Inputs
 - Up to 8 Low Frequency Digital Inputs
 - 4 VR Frequency Inputs
 - 2 Wide Range O2 (I) Sensor Inputs (Bosch LSU4.2)
 - Dual Lambda Sensor Interface
 - 2 Dual Sensor Wide Band Knock Detectors
- Outputs:
- 6 3A Peak/1A Hold Injector Drivers
 - 6 6A Peak/2A Hold Injector Drivers
 - Up to 16 TTL Level Ignition System Outputs
 - Up to 10 6A Low Side PWMs
 - 1 6A Tachometer Output
 - 2 5A H-Bridge PWMs
 - 1 10A H-Bridge PWM
 - 1 Relay Driver (Main Power)

- Datalinks:
- CAN 2.0B Channels
 - ISO 9141 Channel
 - KWP2000/HWP2000, 10.4kbps)
 - RS485 Channel

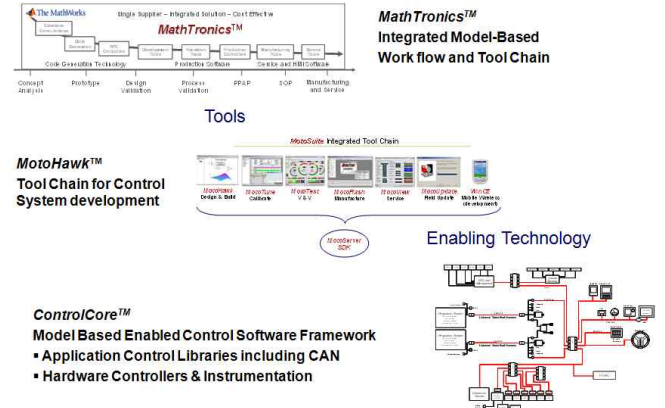
- Datalinks:
- CAN 2.0B Channels
 - ISO 9141 Channel
 - KWP2000/HWP2000, 10.4kbps)
 - RS485 Channel

그림 1은 작성된 Motohawk 소프트웨어의 엔진제어를 위한 구동화면이다.



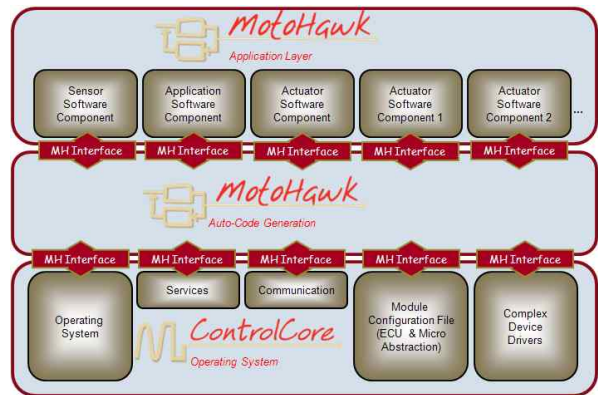
[그림 1] Motohawk 구동 화면

그림 2는 Motohawk의 시스템개발을 위한 도구들을 보여주고 있다. 그래픽을 기반으로 하여 제어로직을 Motohawk ECU에 임베딩할 수 있다.

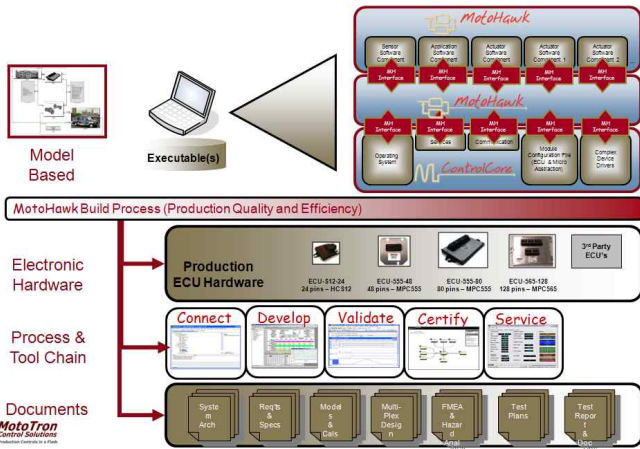


[그림 2] Motohawk의 시스템 개발도구

그림 3 및 그림 4는 Motohawk ECU의 프로그램 개발과정을 보여주고 있다.



[그림 3] Motohawk의 프로그램 개발과정(1)

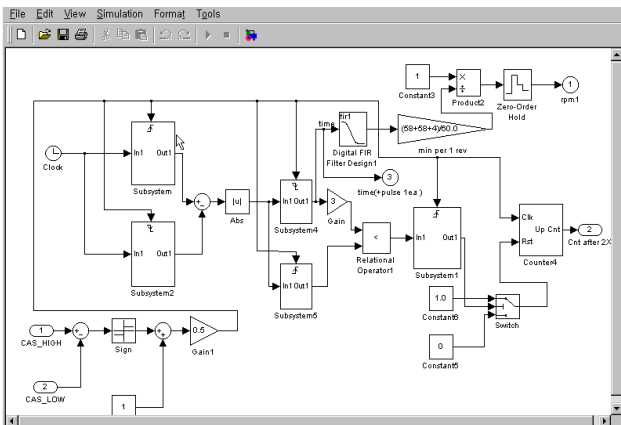


[그림 4] Motohawk의 프로그램 개발과정(2)

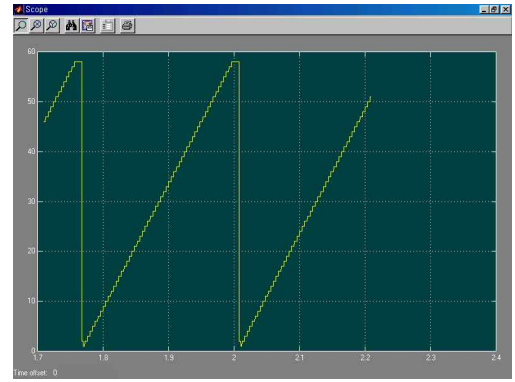
모델 베이스로 프로그램을 구성하여 Build 과정을 거쳐 ECU 하드웨어에 다운로드가 가능하며 초기에 ECU 로직 개발과정 기간을 단축할 수 있다.

3.3 센서 신호를 얻기 위한 모듈

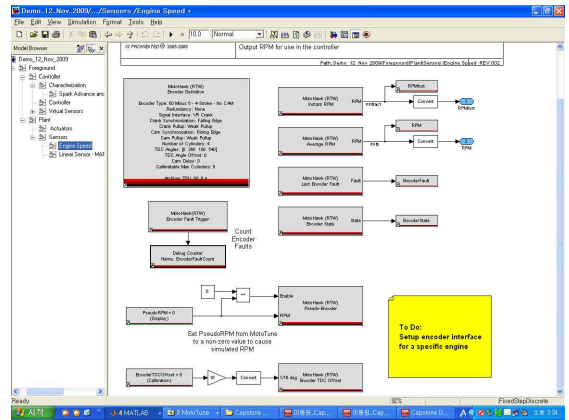
Motohawk 프로그램과 센서 신호를 받아 처리하기 위한 모듈 자동차의 ECU와 비슷한 역할을 할 수 있는 기능이 있다. 아래 그림 5는 Matlab-Simulink로 작성한 크랭크앵글센서로부터 엔진회전수를 구하는 알고리즘이며 복잡한 과정을 거쳐 엔진회전수를 계산할 수 있다. 그림 6은 이러한 과정을 거쳐 58X signal의 계산된 결과를 보이고 있다. 그러나 그림 7과 같이 Motohawk에서는 Encoder와 관련된 블록박스를 활용하여 크랭크앵글센서의 타입을 결정하여 쉽게 엔진회전수를 측정할 수 있다. 그림 8에서와 같이 이러한 블록박스를 활용하여 실제 엔진회전수를 측정하였다.



[그림 5] Matlab-Simulink에 의한 엔진회전수 측정 알고리즘



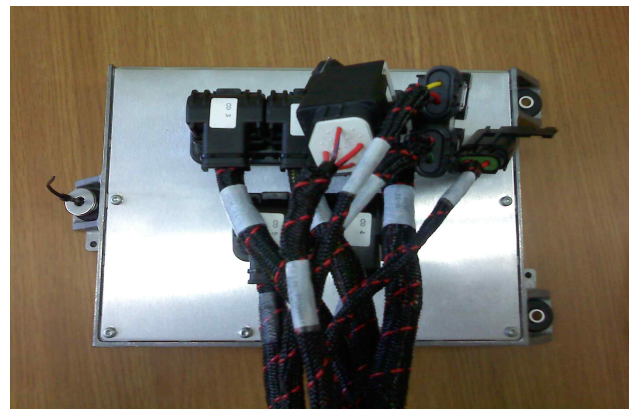
[그림 6] 58X signal output result (X axis(sec), Y axis(count unit))



[그림 7] 엔진회전수 측정을 위한 엔코더 로직

4. Motohawk ECU 구조도

그림 9는 Motohawk ECU565-128의 실제 모습을 보여주고 있다.



[그림 8] Motohawk ECU565-128 의 실제사진

그림 10은 실제 Motohawk의 센서-컨트롤러-액추에이터의 로직을 보여준다.

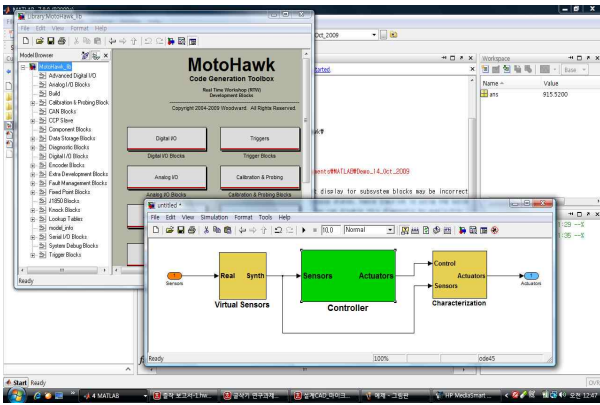
5. 결론 및 향후 계획

Motohawk를 활용하여 범용성 있는 엔진 프로토타입 제어를 설계하였다. 단순히 Matlab-Simulink를 활용하는 것에 비하여 엔진전용의 임베디드 제어 로직을 구현할 수 있었다.

국내에 이러한 툴을 활용한 애프터 마켓 시장에서의 경쟁은 아직은 미약하며, 국내 자동차 및 운송장비 및 선박 등의 엔진 제어기에 활용이 가능하다. 하드웨어보다는 소프트웨어를 활용한 무형자산의 성격이 강하여 프로그램 개발 및 자동차엔진에 장착 후 성능 시험 등이 중요하다. Matlab 및 Simulink와 연동된 Motohawk 개발 툴을 활용하여 그래픽기반의 엔진제어 프로그램을 작성하여 실차 장착 시험을 거쳐 여러 차종별로 프로그램의 신속한 수정이 가능한 임베디드 ECU(Engine Control Unit)를 제작하여 모터스포츠 시장 및 튜닝시장에 진출을 목표로 한다.

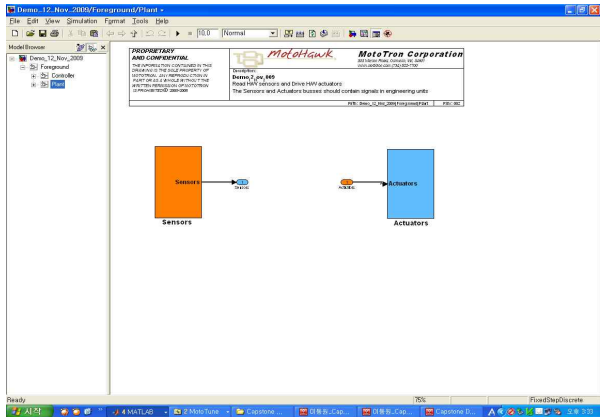
참고문헌

- [1] AMOS GILAT, 황철호외 역, “매트랩의 개요와 응용”, ITC(아이티씨)
- [2] JOHN G. PROAKIS, 김덕년, 오춘석, 윤동원, 이서영, 김남 외 역, “매트랩과 시뮬링크를 이용한 최신통신시스템 (제2판)”
- [3] Marc E. Herniter, 조한근 역, “MATLAB 프로 그래밍”, 한티미디어
- [4] Robert Lawrie, Mike McKenna, “An Architecture Based Design Process for Deploying Control Software on Production Hardware Using Motohawk”, SAE World Congress, Paper Number 2003-01-0853, 2003
- [5] Jeffrey D. Naver, Eric K. Bradley, John E. Szpytman, “Target Based Rapid Prototyping Control System for Engine Research”, SAE Technical Paper Series, 2006-01-0860, 2006
- [6] G.Q. Zhang, D.N.Assanis, “Manifold Gas Dynamics Modeling and Its Coupling With Single-Cylinder Engine Models Using Simulink”, Journal of Engineering for Gas Turbine and Power, ASME, Vol. 125, pp. 563-571, April 2003



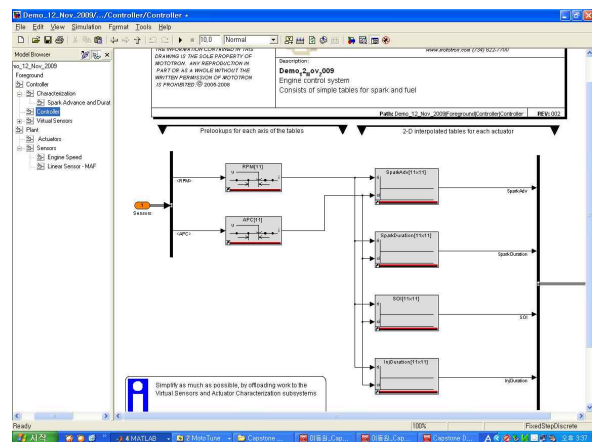
[그림 9] Motohawk의 Sensor-Controller-Actuator 로직

그림 11은 엔진 플랜트의 로직을 보여주고 있다. 엔진 플랜트는 센서와 액추에이터로 구성되어 있다. 센서 신호는 크랭크앵글센서 신호와 엔진 흡입공기량을 입력요소로 하였으며, 출력요소는 점화시기와 연료분사량 등이다.



[그림 10] Motohawk의 엔진 Plant 로직

그림 12는 출력변수 제어로직으로서, 측정된 센서 모듈로부터 엔진회전수와 엔진흡입공기량을 측정하여 점화시기, 점화 드웰시간, 연료분사시점 및 연료분사시간 등을 출력할 수 있는 모듈이다.



[그림 11] 출력 변수 제어