

스파크 점화기관에서의 노킹판단 기준값의 자동수정 알고리즘 개발

The Development of Automatic Correction Algorithm for the Knocking Threshold in Spark Ignition Engine

강 성 현*, 장 광 수**, 서 정 인***, 전 광 민****
Sunghyun Kang, Kwangsoo Chang, Jungin Seo, Kwangmin Chun

ABSTRACT

In this study, a new knocking control algorithm was developed using the knock threshold value auto-correction algorithm. This algorithm uses the Fast Fourier Transform (FFT) method by measuring cylinder block vibration signals of a 1498 cc four-cylinder spark ignition engine. The experimental results show the proposed knock control algorithm provides improved performance compared to existing methods. The results also show that the proposed FFT algorithm provides real-time adjustment of the knock threshold value.

주요기술용어 : Knock Threshold Value(노킹판단 기준값), Vibration Signal(진동 신호), Power Spectrum(전력 스펙트럼), NTCPS(무차원 상호전력 스펙트럼), Knock Control System(노크 제어 시스템)

1. 서 론

본 연구의 목적은 엔진의 출력증가를 위해 점화 시기를 진각할 경우 발생하는 급격한 노킹의 발생을 감소시켜 노킹에 의한 엔진의 피로파괴를 방지하고 엔진의 출력효율을 향상시킬 수 있는 노킹 제어 시스템을 개발하는데 있다. 기존의 노킹 제

어 시스템^{1),2),3),4)}은 고정된 노킹판단 기준값을 사용하여 진동센서로부터 진동신호를 취득하고 이를 노킹판단 기준값과 비교함으로써 노킹을 판단하는 구조로 되어있다. 따라서 동일한 노킹 제어 시스템을 장착한 차량은 엔진의 상태에 관계없이 일정한 노킹판단 기준값을 갖게 되며, 엔진의 노후화 등에 의해 엔진의 상태가 변하였을 경우 그 기능을 적절히 수행하지 못하게 된다. 따라서, 본 연구에서는 자동수정 알고리즘을 사용하여 기존의 시스템에서 노킹판단의 목적으로 사용한 노킹판단 기준값을 일정 시간 엔진의 상태를 파악하여 자동으로 설정하는 알고리즘이며, 이 알고리즘은 엔진

* 홍익대학교 기계공학과 대학원

** 정회원, 홍익대학교 기계공학과

*** 정회원, 연세대학교 기계공학과 대학원

**** 정회원, 연세대학교 기계공학과

의 노화와 환경변화 및 각 엔진에 따른 차이점에 의한 노킹판단 기준값 설정의 난점을 극복할 수 있는 장점이 있다. 따라서, 본 연구에서는 엔진의 생산 오차와 엔진의 노화, 센서의 생산 오차 등, 여러 가지 환경적인 요인에 관계없이 항상 적절한 노킹판단 기준값을 결정하기 위하여 진동센서로부터의 진동자료를 이용한 디지털 신호 처리 기법을 적용하여, 적절한 비례 이득값을 결정하고 이를 바탕으로 노킹판단 기준값을 조절해주는 자동수정 알고리즘 (automatic correction algorithm)⁵⁾을 실험 대상 엔진에 직접 적용하여 실험하였다.

2. 노킹제어 알고리즘

2.1 노킹 발생 간격을 이용한 제어^{6),7)}

본 연구에서 노킹을 제어하기 위해서는 확실적인 방법을 도입하여 일정 시간 안에 발생하는 노킹의 발생 비율과 연관된 제어 변수로 노킹 발생 간격(knock occurrence interval, T_{int})을 정의하고 제어에 이용하였다.^{2),4)} 노킹 발생 간격은 노킹이 발생하고 그 다음 노킹이 발생할 때까지의 시간을 의미한다. 또한 기대 노킹 발생 간격(desired knock occurrence interval, T_d)을 설정함으로써 설계자가 원하는 일정수준으로 노킹 발생 간격을 유지할 수 있으며, 이로써 노킹 발생의 허용 빈도를 설계자가 결정할 수 있다. 노킹 발생 간격을 이용한 제어의 기본 개념을 Fig. 1에 나타내었다. 예를 들어 기대 노킹 발생 간격이 20사이클이라 하였을 때, 노킹 발생 간격이 기대 노킹 발생 간격보다 짧으면 점화시기를 지각시키지만, 노킹이 발생하더라도 노킹 발생 간격이 기대 노킹 발생 간격보다 길면 점화시기를 진각시키도록 되어 있다.

Fig. 2는 노킹 발생 간격을 이용한 제어기의 전체적인 구성을 나타낸다. 여기서 ϵ 은 이득 함수(gain function)가 기대 노킹 발생 간격의 변화에 의해서 영향받지 않도록 하기 위한 무차원 값이며, 다음 식(1)과 같이 나타낼 수 있다.

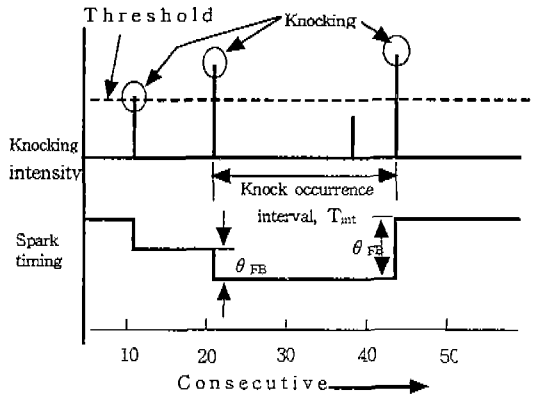


Fig. 1 Basic concept of knock control algorithm

$$\epsilon = \frac{T_{int} - T_d}{T_d} \quad (1)$$

$$\begin{cases} T_{int} : \text{노킹 발생 간격} \\ T_d : \text{기대 노킹 발생 간격} \end{cases}$$

(1)식에 의해 계산된 ϵ 에 따라서 미리 설정한 이득 함수를 참조하여 현재 점화시기(θ_{IG})에서의 점화시기 보정량(θ_{FB})을 결정하게 된다.

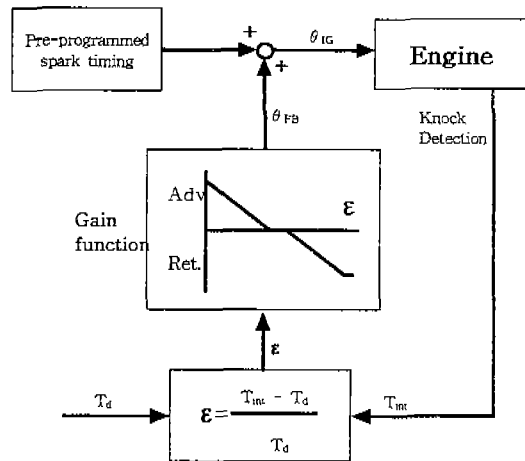


Fig. 2 Configuration of Knock interval controller

2.2 노킹판단 기준값의 자동수정 알고리즘⁵⁾

노킹 제어 시스템(KCS)에서의 노킹 검출은 연소사이클마다의 노킹센서의 최고값(peak value)이 연소실의 여러 가지 구성인자들, 특히 흡·배기 밸브의 개폐 등에 의한 배경잡음(background noise)에 의해 유발되는 기준값(background noise level: BGL)과 이 BGL값에 어떤 일정 비례 이득값을 곱하여 얻은 값인 노킹판단 기준값(knock threshold value, V_{th})을 초과하는 값이 검출되었을 때를 노킹으로 판단하는 구조를 갖는다. 그러므로 BGL값에 따른 적절한 비례 이득값(K)의 선정이 적절한 노킹의 판단을 이끌 것이며, 이에 따른 점화시기의 조절은 기존의 단순 노킹 제어를 채택한 엔진보다 유리한 엔진 성능을 보일 것이다. 그러나, 적절한 노킹판단을 하기 위한 노킹판단 기준값의 선정에 있어서 중요한 인자인 비례 이득값(K)은 각각의 모든 엔진에 대해 적절히 설정된 값이 아니며 단지 엔진 생산 초기에 몇몇 실험된 엔진들로부터 얻은 정보의 평균값이다. 따라서 이 값은 엔진의 생산오차나 엔진의 노화, 센서의 생산오차 등 여러 가지 환경적인 변화요인들 때문에 모든 엔진에 동일하게 적용되어질 수 없으며, 동일 엔진에 있어서도 각각의 엔진 속도에 따라 야기되는 BGL값 또한 다양하게 변화하기 때문에 이 비례 이득값은 하나의 고정된 값이 아닌, 그때 그때의 엔진의 상황에 따라 적절히 조절되어야만 한다. 그러므로, 앞에서 말한 환경적인 요인에 관계 없이 항상 적절한 노킹판단 기준값을 결정하기 위해서는 상대적으로 긴 시간동안의 엔진 상태를 파악하고 이에 따라 적절히 비례 이득값을 조절해주는 장치를 필요로 하며, 이를 위하여 본 연구에서는 디지털 신호처리 기법을 적용한 노킹판단 기준값의 자동수정 알고리즘(automatic correction algorithm)을 개발하였다.

2.3 노킹판단 기준값의 자동 수정의 개념

본 연구에서 사용된 중간값 필터링에 의해

구해진 BGL값에 어떤 일정 비례 이득값을 곱하여 얻은 값 이상의 값이 검출되었을 때를 노킹으로 판단하는 식(2)와 같은 형태를 갖는다.

$$V_{th} = K \times BGL + V_{offset} \quad (2)$$

$$\begin{cases} V_{th} & : \text{threshold value} \\ K & : \text{gain constant} \\ BGL & : \text{background noise level} \\ V_{offset} & : \text{offset value} \end{cases}$$

엔진속도의 변화등 여러 환경적인 요인에 관계 없이 항상 적절한 노킹판단 기준값을 결정하기 위해서는 상대적으로 긴 시간동안 엔진 상태를 파악하는 장치와 이에 따라 비례 이득값을 조절해주는 장치를 필요로 하며 이에 대한 기본적인 개념도는 Fig. 3과 같다. 매 사이클마다 진동센서로부터 취득되는 진동신호(VP)에서 중간값 필터링을 거쳐 BGL값이 계산되어지며, 이를 이용 노킹판단 기준값을 계산하여 원신호(VP)와 비교하여 현재 사이클에서 노킹이 발생했는지를 판단하고 이에 따라 적절한 점화시기를 조절하게 된다. 또한 일정 시간동안의 BGL값과 원신호를 이용하여 실제 노킹이 있었는지를 판단하고 이를 동일 시간 동안의 노킹 발생 비율과 비교하여 비례 이득값이 적절한지를 판단하고 이를 조절하게 된다.

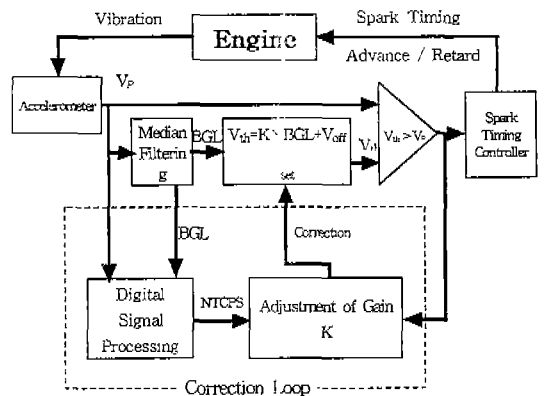


Fig. 3 Block diagram of auto-correction algorithm for threshold value

3. 진동자료를 이용한 시뮬레이션

엔진의 상태를 파악하기 위한 장치의 일부로서 신호처리용 알고리즘을 이용하였다. 노킹센서로부터 취득되는 진동신호는 이산화된 신호이다. 노킹판단 기준값에 의한 노킹이 아닌 실제로 엔진에서의 노킹의 발생여부를 판단하기 위해서는 전처리 과정으로 이 진동신호를 변환해야한다. 따라서 이산시간 신호에 대해서 푸리에 변환을 이용하려면 이산 푸리에 변환(Discrete Fourier Transform: DFT)이 필요하다. 또한 현재의 노킹 판단 기준값에 의한 노킹의 판단이 적절한 것인지를 판단하기 위해서는 무차원화된 평균 상호전력스펙트럼, 즉 NTCPS(Non-dimensional Total Cross Power Spectrum)가 식(3)과 같이 정의된다.⁵⁾

$$NTCPS = \frac{E[x_k y_k]}{E[x_k^2] + E[y_k^2]} \quad (3)$$

이를 바탕으로 노킹상황에 따라 적절히 비례이득을 조절하는 알고리즘을 사용하였다.⁵⁾

이와 같은 자동수정 알고리즘을 검증하기 위하여 엔진에서 얻은 진동자료를 이용하여 시뮬레이션을 수행하였으며, 장시간 사용하여 노킹이 심하게 발생하는 엔진을 A-type, 실시간 실험에 이용한 비교적 노킹이 적게 발생하는 엔진을 B-type으로 하여 구분하였다.

Fig. 4는 1500rpm의 몇몇 점화시기에 대하여 연속적인 3200사이클을 32사이클 단위로 나누어 매 32사이클별로 미리 결정된 고정된 노킹판단 기준값(55m/s²)을 초과하는 노킹이 발생한 횟수에 대한 노킹강도의 NTCPS값을 나타낸 것으로 노킹이 심하게 발생할수록 낮은 값을 갖게 된다. 실제로 노킹이 발생하지 않는 BTDC 16°에서 노킹이 발생한 사이클(number of knock ≥ 1)은 존재하지 않으며 이때 NTCPS값은 3이상의 값을 보인다. 반면 노킹이 발생하기 시작하는

BTDC 18°에서는 NTCPS값이 3이하의 값을 보이고, 노킹판단 기준값을 초과하는 신호가 없는 사이클(number of knock=0)에서는 NTCPS값이 대부분 3이상의 값을 가짐을 알 수 있다. 노킹이 심하게 발생하는 BTDC 20° 이상에서는 노킹이 발생하는 대부분의 사이클에서 2이하의 NTCPS값을 가짐을 알 수 있다. 따라서 점화시기가 진각될수록 노킹의 빈도와 강도가 커져서 NTCPS값이 점점 작아짐을 확인할 수 있으며, 노킹이 발생하지 않을 경우에는 NTCPS값이 전반적으로 2정도의 값을 가짐을 알 수 있다.

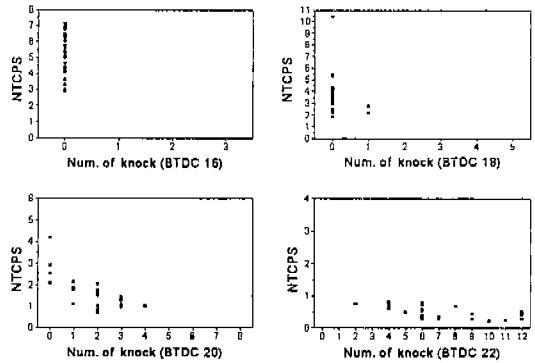


Fig. 4 NTCPS for different spark timings

(1500rpm, WOT, A-type engine)

Fig. 5는 예비실험에 의해서 가장 적절하다고 판단된 노킹판단 기준값(55m/s²)으로 고정시킨 후 1500rpm에서 노킹 발생 간격을 이용한 점화시기 제어를 실행한 결과이다. 점화시기가 노킹 발생 점화시기와 최대 제동 토크 점화시기 사이에 머물러 있음을 볼 수 있다. 이를 바탕으로 자동수정 알고리즘을 이용한 점화시기 제어로 시뮬레이션을 실시하여 1500rpm에서 비례 이득값이 매우 높게 (K=3.0) 설정되었을 경우에 대한 결과를 Fig. 6에 나타내었다. 초기에 매우 높게(K=3.0) 설정된 비례 이득값에 의해 지나치게 높게 노킹판단 기준값이 설정되어 노킹이 발생되고 있음에도 불구하고 노킹판단 기준값이 적절치 않아 노킹 판정에 대한 오류가 발생하고 있는 것을 볼 수 있다. 그러나 시

간이 지남에 따라 수정 알고리즘에 의해 비례 이득값이 줄어 노킹판단 기준값이 수정되어 감을 볼 수 있다. 또한 적절한 노킹판단 기준값(55m/s^2)으로 수렴 후 점차 비례 이득값의 수정량이 줄어들어 1400사이클(1500rpm 경우 약28초) 이후에는 비례 이득값의 조정 없이 안정된 노킹판단 기준값을 유지함을 볼 수 있다. 이렇게 안정된 노킹판단 기준값에 의해 초기 수정 과정을 제외하고는 주로 최대 제동 토크 점화시기와 노킹 발생 점화시기 사이에 점화시기가 머물고 있는 것을 볼 수 있어 제어 성능에도 큰 문제가 없는 것을 알 수 있으며, 또한 노킹판단 기준값이 유지된 이후 노킹이 심하게 발생하지 않는 것을 볼 수 있어 노킹제어가 되고 있음을 알 수 있다.

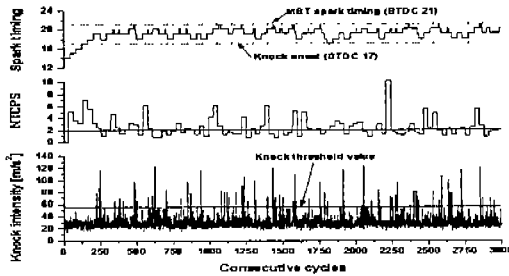


Fig. 5 Simulation result of fixed threshold value

(1500rpm, WOT, Td=20, A-type engine)

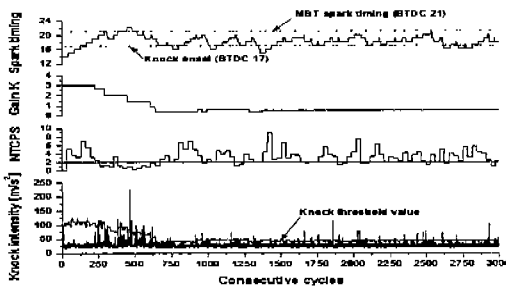


Fig. 6 Simulation result of automatic correction algorithm for threshold value

(A-type, 1500rpm, WOT, Td=20, too high initial gain K)

Fig. 7은 초기 비례 이득값이 낮게($K=0.5$) 설정되어 있을 경우에 노킹판단 기준값이 적절히 조절됨을 보여주고 있어, 낮은 비례 이득값에서도 적절한 노킹판단 기준값으로 수렴함을 볼 수 있다.

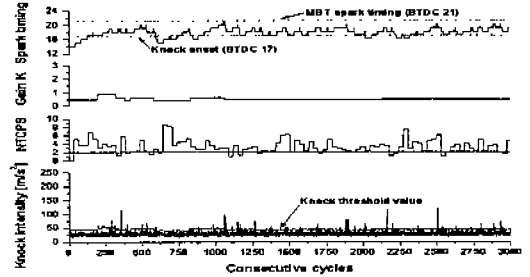


Fig. 7 Simulation result of automatic correction algorithm for threshold value

(A-type, 1500rpm, WOT, Td=20, low initial gain K)

Fig. 8은 1500rpm에서 3000사이클동안 각 점화시기에 대해 노킹이 발생한 사이클의 백분율을 구한 것이다. 자동수정 알고리즘에 의한 노킹 발생률이 고정된 노킹판단 기준값에 의한 노킹 발생률보다 전반적으로 저감된 영역에서 나타나고 있다. 하지만 Fig. 9에서 보듯이 전반적인 노킹 발생률은 자동수정 알고리즘이 고정된 노킹판단 기준값으로 노킹 제어를 할 때 보다 더 낮음을 알 수 있다. Fig. 10은 비례 이득값이 수렴한 사이클부터 1500사이클동안 노킹강도를 각 점화시기별로 나타낸 결과로 자동수정 알고리즘과 노킹판단 기준값이 고정되었을 경우를 비교하였으며, 전반적으로 두 알고리즘이 같은 노킹강도를 보임을 알 수 있다. 따라서 적절하지 못한 노킹판단 기준값이 자동수정 알고리즘에 의해 기존의 여러 실험에 의해 결정된 고정 노킹판단 기준값을 추종해 감을 알 수 있다.

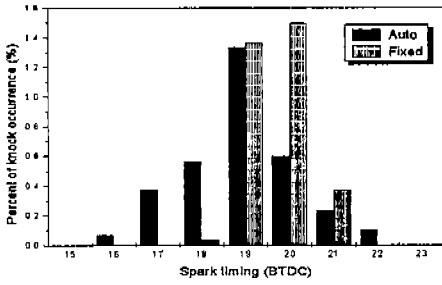


Fig. 8 Percentage of knock occurrence for different spark timings during 3000 cycles

(A-type engine, 1500rpm, WOT, Td=20 : Sim.)

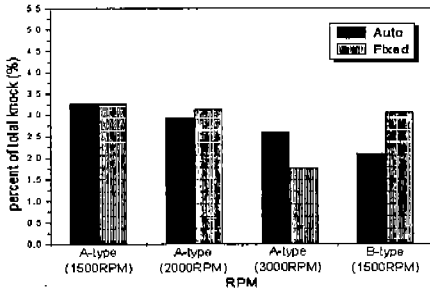


Fig. 9 Percentage of total knock under different engine speeds during 3000 cycles

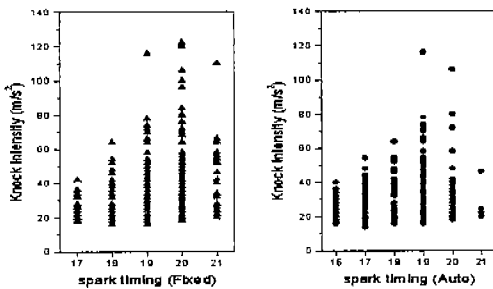


Fig. 10 Comparison of knock intensities after steady state of proportional gain for knock threshold value (A-type engine, 1500rpm, WOT : Sim.)

점화시기 제어 및 자동수정 알고리즘은 C

언어를 이용하여 구성하였으며, 알고리즘에 이용된 개략적인 함수의 흐름도를 Fig. 11에 나타내었다. 이러한 함수들은 자동 수정 부분과 노킹 발생 간격 제어 부분으로 크게 나눌 수 있으며 두 부분은 서로 독립적으로 작동된다. 점화 장치 부문은 점화시기 진각 및 지각을 위하여 따로 타이머용 칩을 사용하여 제작하였고, 센서로부터 나오는 신호는 전하 증폭기와 아날로그 필터를 사용하여 신호를 취득하였다.

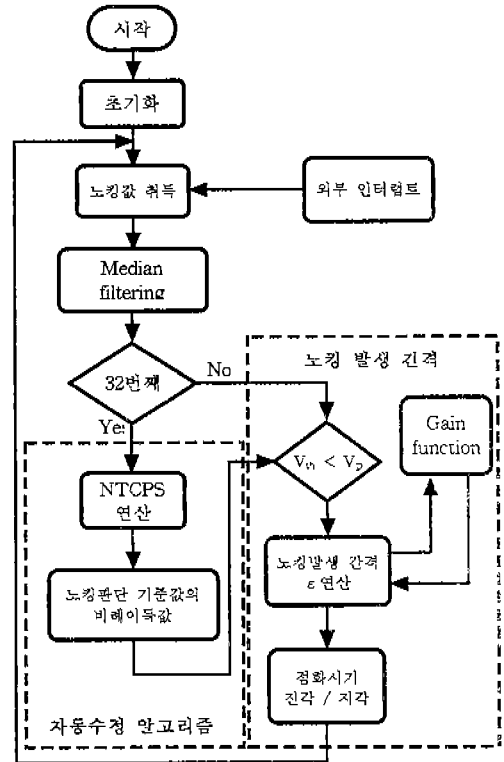


Fig. 11 Flow chart of auto-correction

Fig. 12는 실험 장치의 구성을 간단하게 나타낸 그림이다. 스위치로 엔진의 점화장치를 일반 상용 엔진에서 사용하는 C-ECU에서 PC-ECU로 바꿀 수 있으며 점화시기는 초기에만 서로 동기화 되어 작동하며 PC-ECU에 새로운 점화시기를 입력하면 독립적으로 자체 타이머 회로에 의해서 점화시기가 작동된다.

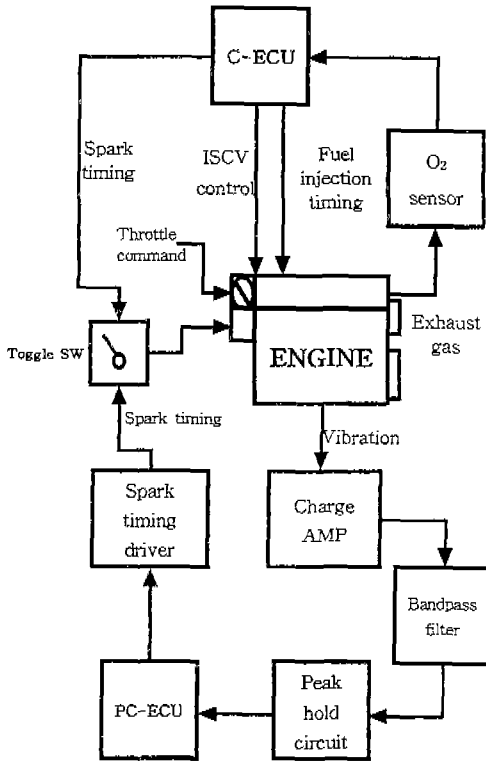


Fig. 12 Block diagram of PC-ECU under connection with C-ECU

실시간 실험에서 중요시 해야 할 또 다른 사항은 여러 가지 연산을 담당하는 컴퓨터, 즉 CPU부분이며 이는 차량의 ECU내의 중앙 처리 장치 부분에 해당한다. 실제로 차량의 엔진 제어는 ECU에 의한 연산과 내부의 제어 알고리즘에 의존하기 때문이다. 본 실험에서는 자동수정 알고리즘을 적용하기 위하여 intel사의 pentium 133MHz의 microprocessor가 장착된 컴퓨터를 사용하였다. 이는 자동수정 알고리즘의 FFT를 연산하는 부분이 많은 루프 연산으로 구성되어 있기 때문에 매 사이클마다 진동신호를 취득하여 이를 이용 중간값 필터링을 수행하고 FFT window인 매 32 points (진동센서로부터 취득된 32개의 진동값)마다 FFT연산, 전력 스펙트럼, NTCPS값 연산, gain값 수정 등을 하기 위해서는 충분히 빠른 연산속도를 가진 컴퓨터가 필요

하다고 판단되었기 때문이다. 위의 장치를 이용하여 실제로 중간값 필터링이 포함되어 있는 FFT 알고리즘의 연산시간을 FFT window point 별로 분석해본 결과는 Table. 1과 같다.

단위 : ms			
FFT window (points)	1개의 FFT window 연산시간 (A+B)	진동신호 취득시간 (A)	순수 FFT 연산시간 (B)
32	1.21	0.8	0.41
64	2.39	1.56	0.83
128	4.68	3.14	1.54
256	9.38	6.20	3.18
512	19.0	12.3	6.70

Table 1 Comparison of computing time for different FFT window sizes

1개의 FFT window 연산은 진동신호 취득과 순수 FFT만을 연산하는 두 부분으로 구성된다. 따라서 엔진 속도가 1500rpm일 때 진동신호는 매 실린더마다 취득할 경우 약 20ms단위로 취득해야 하며 3000rpm일 경우에는 10ms단위로 취득해야 한다는 결론을 얻게 된다. 이는 진동신호를 취득하여 단순히 노킹 여부를 판단하는 경우에는 문제가 되지 않지만, 실제 노킹판단 기준값의 적절함을 판정하기 위한 FFT 및 NTCPS를 연산해야 하는 FFT window의 마지막 사이클에서는 연산과 제어 알고리즘이 다음 진동신호 취득 이전에 종료되어 점화시기가 제어되어야 함을 의미한다. 133MHz의 CPU속도를 가진 PC로 DOS 환경에서 실시한 Table.1의 결과는 프로그래밍 언어와 기법에 따라서 다소 차이가 있겠지만, 본 알고리즘이 1500rpm에서는 512 points 이상, 3000rpm에서는 256 points이상의 FFT를 사용하기에 어려움이 있다는 것을 보여준다. 따라서 8MHz의 8 bit PC에서는 더 긴 연산시간이 소요되며 32 points에 대한 FFT 알고리즘만 연산하는데 약 6.82ms(32 window points의 순수 FFT 연산시간 0.41ms×133MHz/8MHz)가 소요

따므로 본 알고리즘을 적용시키는데 무리가 따를 수 있다. 일반적인 신호분석에서 FFT 알고리즘의 window point를 크게 할수록 본래의 연속신호에 가까운 분석을 할 수가 있다. 결국 8 bit PC와 유사하거나 더 느린 약 2MHz의 연산속도를 가진 ECU는 본 알고리즘을 적용하는데 적합하지 않을 수 있으며, 32 window points의 FFT 알고리즘을 사용하기 위해서는 적어도 16MHz의 기준 클럭속도를 가진 시스템을 사용하거나 속도가 느린 시스템에서는 어셈블리 언어나 기타 장치를 이용하여 연산시간을 단축시킬 필요가 있다. 따라서, 향후에는 FFT연산루틴에 어셈블리를 이용하여 연산부하를 감소시키고 프로그램의 각 함수들을 최적화 할 것이며, 또한 전용 DSP(Digital Signal Processor)를 병용하여 ECU와 독립된 FFT 연산작업만을 수행시킴으로써 상용 ECU에서 본 알고리즘이 작동할 수 있도록 할 것이다.

노킹을 결정하기 위한 방법으로는 예비실험에서 압력신호 자료에 대해 디지털 밴드패스 필터링(digital bandpass filtering)을 수행한 뒤 나타나는 압력진동의 최대 진폭크기를 이용하여 이때 진동센서로부터 나오는 신호의 최대 진폭크기를 노킹강도로 결정하였다.²⁾ 밴드패스의 범위는 스펙트럼 분석 결과 나타난 노킹주파수 영역중 1차 모드인 5~10kHz 대역을 선택하였다. 이는 노킹의 주파수 대역이 실린더의 크기에 따라 다소 다르지만 일반적으로 3~6kHz에 존재¹⁰⁾하며, 이를 바탕으로 진동신호의 스펙트럼을 분석하여 스펙트럼 밀도(spectrum density)가 크게 나타나는 영역을 중심으로 선택한 것이다.

실시간 실험은 각각 고정된 노킹판단 기준값과 자동수정 알고리즘에 의해서 조절되는 노킹판단 기준값에 대해서 노킹 발생 간격을 이용한 제어기로 실시하였다. Fig. 13은 고정된 노킹판단 기준값에 의한 점화시기 제어의 결과이다. 토크 이득 곡선에서 100%에 고정된 직선은 점화시기를

제어하지 않는 C-ECU의 토크 이득이다. 반면 100%의 영역으로부터 점점 상승하고 있는 것은 노킹 제어기에 의한 토크의 변화로 점화시기의 진각에 따라 향상되고 있음을 알 수 있다.

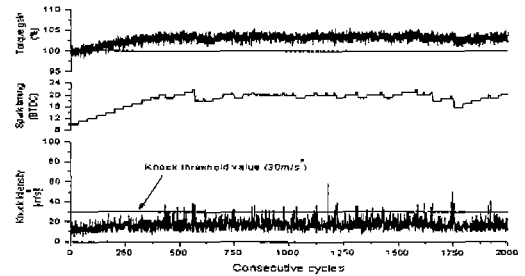


Fig. 13 Experimental result of fixed threshold value
(B-type engine, 1500rpm, WOT, Td=20)

Fig. 14와 Fig. 15는 자동수정 알고리즘에 의한 결과이다. Fig. 14는 초기 비례이득값이 높게 설정되었을 경우로 초기 노킹판단 기준값의 잘못된 설정으로 인한 노킹이 발생하고 있으나 비례 이득값이 수렴하는 1300사이클(26sec) 이후부터는 자동 수정된 노킹판단 기준값에 의해 안정된 점화시기를 보이며 토크 이득도 향상됨을 볼 수 있으나 점화시기 제어에 따른 토크 이득의 변동이 발생함을 볼 수 있다.

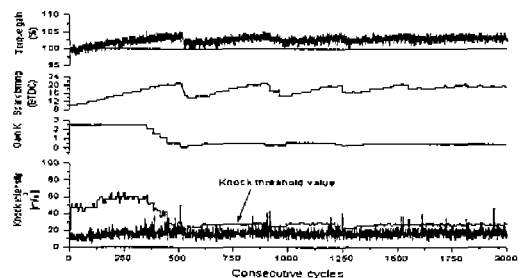


Fig. 14 Experimental result of automatic correction algorithm for threshold value
(B-type, 1500rpm, WOT, Td=20, too high initial gain K)

Fig. 15는 초기 노킹판단 기준값이 낮게 설정되었을 경우로 노킹판단 기준값이 높게 설정되었을 경우보다 비교적 빠르고 안정된 점화시기를 보이고 있으며 토크 이득도 전반적으로 안정됨을 볼 수 있다.

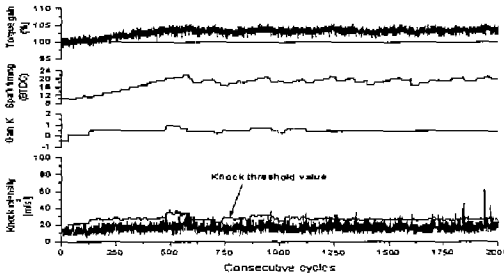


Fig. 15 Experimental result of automatic correction algorithm for threshold value

(B-type, 1500rpm, WOT, Td=20, low initial gain K)

이로써 자동수정 알고리즘이 노킹판단 기준값을 처음부터 고정하여 사용했을 때보다 점화시기가 최대 제동 토크 발생시기로부터 다소 떨어져 있지만 연속적으로 발생하는 심한 노킹을 피할 수 있으며¹¹⁾, 엔진상태나 사용환경이 변하더라도 자동수정 알고리즘은 적절한 노킹판단 기준값을 결정할 수 있는 반면 고정된 노킹판단 기준값을 사용할 경우에는 이 기준값을 자동으로 조절해주는 알고리즘이 없으므로 잘못된 노킹판단 기준값에 의해 노킹 제어 시스템이 적절한 제어를 할 수 없음을 예상할 수 있다. 결국 장시간 엔진 상태를 파악하여 노킹판단 기준값을 자동으로 설정하는 자동수정 알고리즘은 적절한 노킹판단 기준값을 설정할 수 있으므로 노킹판단 기준값 설정의 어려움을 극복할 수 있게 할 것이며, 엔진의 사용환경이 변하더라도 적절한 노킹 제어로 엔진의 성능향상을 가져올 것이다.

4. 결 론

본 연구에서는 4기통 스파크 점화기관을 대상으

로 연속적인 노킹의 발생을 피하면서 점화시기를 제어하여 토크 이득향상을 목적으로 실린더 블록 진동 신호를 이용한 노킹 제어기를 개발하고 노킹판단 기준값의 자동수정 알고리즘을 연구하였으며, 이로써 다음과 같은 결론을 얻을 수 있었다.

1) 노킹제어를 사용함으로써 노킹제어가 없는 엔진에 비하여 연속적인 노킹을 피하면서 약 3%의 토크 이득을 얻을 수 있었다.

2) 노킹 발생 간격을 제어 변수로 사용함으로써 기대 노킹 발생 간격의 적절한 조절에 의해 일정 수준으로 노킹 발생 비율을 유지하면서 토크 이득을 최대로 할 수 있음을 실험을 통해서 알 수 있었다.

3) 무차원화된 평균 상호 전력 스펙트럼(NTCPS)값을 이용하여 노킹 상황의 판단을 위한 척도로 이용할 수 있음을 알 수 있었으며, 노킹판단 기준값의 자동수정 알고리즘에 의해 자동으로 노킹판단 기준값의 조절할 수 있으며 이를 이용 점화시기를 제어하여 토크 향상 및 노킹 발생을 억제할 수 있음을 실험을 통해서 알 수 있었다.

4) 노킹판단 기준값의 자동수정 알고리즘을 이용한 점화시기 제어가 고정된 노킹판단 기준값에 의한 점화시기 제어와 비교하여 보다 적은 노킹이 발생함을 알 수 있었으며, 자동적으로 노킹판단 기준값을 조절하여 노킹제어를 할 수 있으므로 엔진의 환경변화에도 적절한 성능을 유지할 수 있음을 기대할 수 있다.

후 기

본 연구는 1997년도 교육부 학술연구조성비(기제공학:ME97-D-23)에 의하여 연구되었습니다.

참 고 문 헌

1. 이재형, 함윤영, 장광수, 전광민, "스파크 점화기관의 노킹제어를 위한 퍼지제어기 개발", 대한기계학회 논문집, 제21권, 제2호, pp. 223~231, 1997.

2. 함윤영, “스파크 점화 기관의 노킹 제어 알고리즘 개발 및 노킹 판정 기준값 설정”, 연세대학교 대학원 박사학위 논문, 1996.
3. 이기형, “자동차용 가솔린 엔진의 연비향상을 위한 최신기술”, 한국자동차공학회지, Vol. 17. No. 1, pp. 44~53, 1995.
4. 김연준, 고상근, “노크센서를 이용한 점화시기 피이드백 제어에 관한 연구”, 한국자동차공학회지, Vol. 14. No.4, pp. 61~67, 1992.
5. 이재형, “스파크 점화기관의 노킹 제어 알고리즘 개발”, 홍익대학교 대학원, 석사학위 논문, 1996.
6. D.Sawada and T. Shigematsu, “Improvement of Spark Ignition Knocking Control Detector Performance by Learning Control”, SAE PAPER 810057, 1981.
7. 함윤영, 전광민, 이재형, 장광수, “진동신호를 이용한 스파크 점화기관 노킹제어 알고리즘의 개발”, '94한국 자동 제어 학술회의 논문집, pp. 142~147, 1994.
8. PAUL M. EMBREE, BRUCE KIMBLE, “C LANGUAGE ALGORITHMS FOR DIGITAL SIGNAL PROCESSING”, Prentice-Hall International, 1991.
9. 이재형, 장광수, 함윤영, 전광민, “디지털 신호처리 기술을 이용한 노킹판단 기준값의 자동수정 알고리즘 개발”, '96년도 한국자동차공학회 춘계학술대회 논문집, Vol. 1, pp. 519~524, 1996.
10. John B. Heywood, “Internal Combustion Engine Fundamentals”, Mc-Graw Hill, 1988.
11. 강성현, 서정인, 장광수, 전광민, “FFT를 이용한 노킹판단 기준값의 자동수정 알고리즘 개발”, '98년도 한국자동차공학회 추계학술대회 논문집, Vol. 1, pp. 160~165, 1998.