

# PSO기반 디젤엔진 제어 게인 맵 자동생성 알고리즘

강 원 모 · 안 창 선\*

부산대학교 기계공학부

## PSO-based Diesel Engine Control Gain Map Auto-tuning Algorithm

Weonmo Kang · Changsun Ahn\*

School of Mechanical Engineering, Pusan National University, Busan 46241, Korea

(Received 11 April 2022 / Revised 25 May 2022 / Accepted 4 August 2022)

**Abstract** : The diesel engine for power generation is controlled using a PID controller. To use the PID controller in a diesel engine, it is necessary to know the appropriate P, I, and D gains for each load applied to the engine. Manual determination of gain causes performance to vary, and the problem of relying too much on the engineer's skill may occur. Through the development of the gain tuning automation algorithm, tuning performance can be performed while ensuring constant performance without heavy reliance on individual engineers. The algorithm for automatic gain tuning stores the measured fuel amount, load, and engine rotation speed data by first applying a load to the engine and operating it. The model identification uses the stored data and creates a gain map using an optimization algorithm. The developed automatic gain tuning algorithm can replace the existing gain tuning method.

**Key words** : PID controller(비례-미분-적분 제어기), Gain auto-tuning(게인 자동 조정), Diesel engine(디젤엔진), PSO(입자 군집최적화), Gain map(게인 맵)

### 1. 서 론

발전용 디젤엔진은 전력망이 설치되지 않은 지역에서 전력을 공급하기 위한 방법으로 많이 사용되고 있다. 따라서 환경을 보전하며 디젤엔진을 동작하기 위한 제어 연구<sup>1)</sup>가 활발히 이뤄지고 있다. 디젤엔진을 이용해 안정적으로 발전을 진행하기 위해서는 엔진 회전속도가 일정 속도를 유지하는 것이 필요하다. 따라서 엔진 회전속도를 안정적으로 유지할 수 있는 제어기를 사용한다면 안정적인 발전이 가능할 것이다. 이때 많이 사용되는 제어기 중 하나는 비례-적분-미분(PID) 제어기이다. PID 제어기는 제어 대상에 따라서 제어 게인을 지정해야만 좋은 제어 성능을 발휘할 수 있다. 따라서 발전용 디젤엔진에 PID 제어기를 사용하기 위해서는 제어기를 이해하는 숙련된 엔지니어가 제어 게인을 튜닝 해 줘야한다. 엔진을 생산하는 과정에서 항상 수행되는 게인 튜닝을 자동으로 수행하는 알고리즘을 개발한다면, 초기 엔진 제

어기 설정 과정을 간소화하여 엔진 생산에 큰 도움이 될 것이다. 이를 위해서 엔진 제어기를 자동 튜닝하는 알고리즘을 개발하는 것이 이번 연구의 목적이다.

현재 퍼지 제어기를 활용하거나,<sup>2,3)</sup> 자가 튜닝 퍼지 제어기를 활용한 게인 설정방법,<sup>4)</sup> 퍼지 신경망<sup>5)</sup>을 활용한 제어기 구조, 신경망 회로를 이용한 최적 제어매개변수 식별<sup>6)</sup> 등 다양한 방안들이 효율적인 PID 제어기 튜닝 방법으로 제안 되고 있다. 현재 연구의 대상이 되는 디젤엔진은 Fig. 1에서 확인할 수 있듯이 게인 맵에서 부하에 따라 게인을 적용하는 PID 제어기를 사용 중이다. 이러한 제어기에 사용되는 게인 맵을 제작하기 위해서는 각 부하 별로 적절한 제어 게인을 찾는 작업을 반복 수행해야한다. 이를 숙련된 엔지니어가 직접 수행하는 현재의 상황에서는 엔진 제어 성능이 엔지니어의 숙련도에 의해 변할 수 있다. 따라서 이를 자동화 하는 방법이 필요하다.

\*Corresponding author, E-mail: sunahn@pusan.ac.kr

<sup>1)</sup>This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium provided the original work is properly cited.

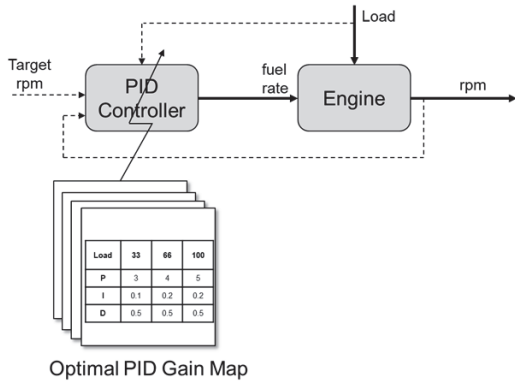


Fig. 1 PID Controller gain update process

엔진 튜닝 과정 중에 모든 동작을 실제 엔진으로 수행한다면 시간과 비용문제, 엔진의 고장이 발생할 수 있다. 이러한 단점을 극복하기 위해 엔진 모델을 구성, 엔진을 대체하여 튜닝 과정에 활용하는 방법이 있다. 이러한 방법은 튜닝 과정의 시간과 비용문제를 해결해줄 수 있으나, 엔진 모델의 정밀도에 따라 튜닝 성능의 영향을 받을 수 있다.

엔진 모델은 엔진의 데이터에서 모델을 추정하는 방법을 사용했다. 최소 자승법(LS, Least Square)를 활용하여 데이터에서 모델의 파라미터를 뽑아내는 방식을 통해 엔진 모델을 Identification(ID)하였다.

엔진 모델을 ID한 이후에는 제어를 자동으로 튜닝하는 방법으로 입자 군집 최적화(PSO, Particle Swarm Optimization)를 사용하였다. PSO는 최적의 값을 찾는 방법론 중 하나로 무작위성을 가지면서 정해진 범위 내에서 반복적으로 시뮬레이션을 수행, 영역 내에서 최적의 값을 찾아준다.

위 서술한 과정들을 반복 수행하면서 게인 맵을 자동으로 생성할 수 있다. 알고리즘을 통해 자동으로 게인 맵을 생성하고 이를 검증하는 과정은 엔진 시뮬레이션 모델을 통해 수행했다.

## 2. PID 게인 맵 자동생성 알고리즘

발전용 디젤 엔진의 경우 엔진이 부담해야하는 발전량의 크기에 따라 엔진에 필요한 연료량이 달라진다. 따라서 현재 PID 제어기는 엔진에 걸리는 부하의 변화에 대응하기 위해 부하 마다 적절한 제어 게인을 미리 찾아서 게인 맵을 제작하고 이를 사용하는 방식을 채택 중이다. 제어 게인 맵은 초기 튜닝 과정에서 제작 되어야한다. 위의 Fig. 2 같이 게인 맵을 자동 생성하는 알고리즘을 통해서 초기 튜닝에 사용되는 시간과 비용을 절약 가능하다.

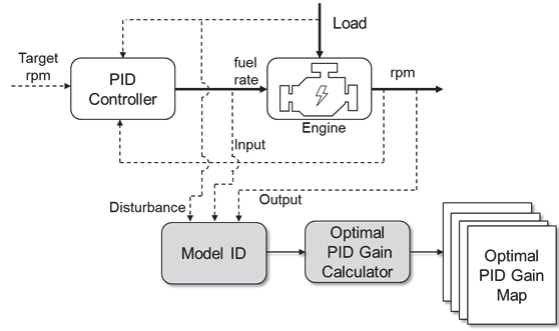


Fig. 2 PID Gain map auto-generate algorithm

### 2.1 모델 ID

발전용 디젤 엔진은 발전량, 즉 부하의 변화에도 일정한 회전속도로 동작해 일정한 전력 주파수를 유지하기 위한 제어가 필요하다. 이러한 목적을 달성하는 제어기 튜닝을 위해서는 부하에 따른 엔진의 반응을 보여주는 엔진 모델이 필요하다. 엔진 모델 ID는 Fig. 3에 나와 있는 방법으로 진행한다. 모델 ID를 진행하기 위해서는 먼저 데이터를 선형화 하는 과정을 거쳐야 한다. 선형화는 부하가 변하기 전의 연료량, 엔진 회전속도, 부하를 이용하여 선형화 한다. 엔진의 회전속도는 연료량과 부하의 영향을 받기 때문에 엔진 모델이 연료량과 부하의 변화에 반응하게 구성하였고 이를 2차 불연속모델로 생성하면 아래의 식 (1)과 같다.

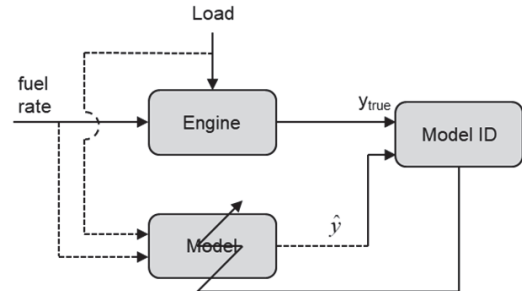


Fig. 3 Engine model ID

$$Y = \frac{b_1 z + b_2}{z^2 + a_1 z + a_2} U(z) + \frac{c_1 z + c_2}{z^2 + a_1 z + a_2} D(z). \quad (1)$$

여기서  $Y$ 는 엔진 회전속도,  $U$ 는 연료,  $D$ 는 부하,  $a_1, a_2, b_1, b_2, c_1, c_2$ 는 파라미터를 의미한다. 식 (1)에서 파라미터는 Fig. 4에서 나타나 있는 것처럼 엔진의 입, 출력 데이터와 엔진에 가해지는 부하로부터 추정하는 방법을 사용했다. 추정방법으로는 LS를 사용하였다.

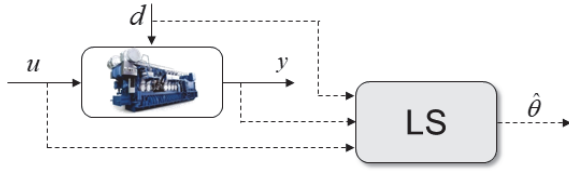


Fig. 4 Engine model parameter estimator

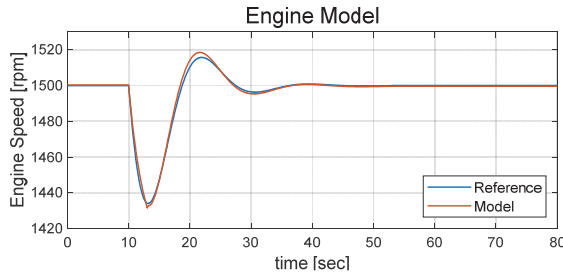


Fig. 5 Engine model identification result

$$\theta^* = \underset{\hat{\theta}}{\operatorname{argmin}} (y_{true} - f(\hat{\theta}))^2. \quad (2)$$

여기서  $\theta$ 는 파라미터,  $y$ 는 엔진 회전속도를 의미한다. 식 (2)는 LS가 수행하는 계산이다. LS는 근사 해와 실제 해의 오차가 최소가 되는 파라미터를 정답으로 찾는다. LS 계산을 통해 엔진 모델 ID를 수행할 수 있다.

Fig. 5는 엔진 모델 ID의 결과 중 하나를 그림으로 나타낸 것으로 엔진이 1,500 rpm으로 회전할 때 부하가 100%에서 0%로 감소하는 경우의 모델 ID 결과를 보여준다. 모델의 경우 최적화를 진행하는 데 있어 엔진과 동작의 경향성이 일치하는 것을 중점적으로 확인하였다.

## 2.2 최적 제어 게인 계산기

### 2.2.1 PSO

PSO는 확률적 인구 기반 최적화 방법으로<sup>7,8)</sup> 다양한 최적화 분야에서 사용되고 있다. PSO의 장점 중 하나는 국소 최적에 빠질 위험이 적다는 것이다. Fig. 6은 PSO를 기반으로 작동하는 최적 게인 계산기의 구조를 보여주고 있다. PID 제어기에서 최적의 게인을 구하기 위해서 최적화 방법 중 하나인 PSO 방법을 사용하였다. PSO는 다음 식 (3), (4)를 통해 최적화 과정을 수행한다.

$$dx_{id}^{n+1} = \omega \cdot dx_{id}^n + c_1 \cdot \operatorname{rand}() \cdot (p_{id}^n - x_{id}^n) + c_2 \cdot \operatorname{rand}() \cdot (g_{id}^n - x_{id}^n), \quad (3)$$

$$x_{id}^{n+1} = x_{id}^n + dx_{id}^{n+1}. \quad (4)$$

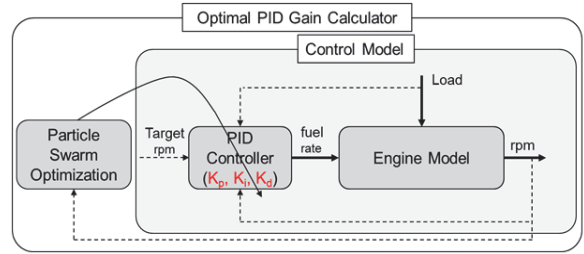


Fig. 6 Structure of PSO-based gain optimizer

여기서  $x$ 는 입자 위치,  $dx$ 는 입자 변위,  $p$ 는 입자가 지나온 최고 위치,  $g$ 는 입자들이 지나온 최고 위치,  $\omega$ ,  $c_1$ 는 계수,  $\operatorname{rand}()$ 는 무작위 계수를 의미한다. 위 식 (3)은 입자 하나의 새로운 속도가 식 (4)는 새롭게 구한 입자의 변위를 이용하여 새로운 위치를 결정해주는 식이다. 여러 개의 입자 각각은 초기에 무작위의 위치와 변위를 가진다. 입자가 지나온 위치들의 비용과 현재위치에서의 비용을 비교하여 가장 작은 비용을 가지는 위치를 선정하고 현재의 위치와 우수한 위치의 차이를 입자의 새로운 변위를 구하는 것에 이용한다. 또한 모든 입자들의 비용 기록에서 최소의 비용을 가지는 위치도 입자 각각의 새로운 변위를 구하는 과정에서 이용한다.

### 2.2.2 비용 함수

최적의 제어 게인을 PSO를 이용해 찾기 위해서는 입자의 위치가 제어 게인을 의미하고 입자의 위치에 따라 비용을 구할 수 있는 비용 함수를 설정해주는 것이 중요하다. 현재 찾고자 하는 제어 게인은 엔진에 가해지는 부하의 변화에도 일정한 회전속도를 유지해줄 수 있는 제어 게인이 필요하다.

$$\operatorname{cost} = \operatorname{cost}_1 + \rho_1(\operatorname{cost}_2 + \operatorname{cost}_3 + \operatorname{cost}_4), \quad (5)$$

$$\operatorname{cost}_1 = \int_0^\infty t \times |error| dt, \quad (6)$$

$$\operatorname{cost}_2 = \begin{cases} 0, OS < 8\% \\ (OS - 8)^2, OS \geq 8\% \end{cases}, \quad (7)$$

$$\operatorname{cost}_3 = \begin{cases} 0, US < 8\% \\ (US - 8)^2, US \geq 8\% \end{cases}, \quad (8)$$

$$\operatorname{cost}_4 = \begin{cases} 0, t_s < 8 \text{ sec} \\ (t_s - 8)^2, t_s \geq 8 \text{ sec} \end{cases}. \quad (9)$$

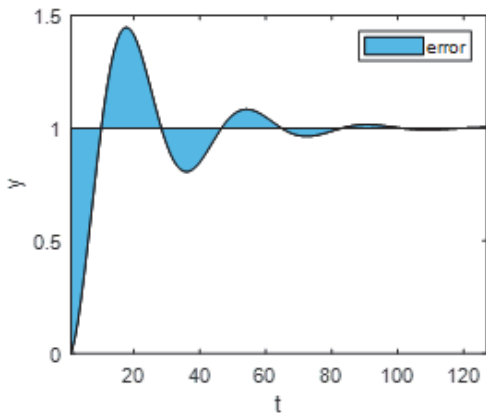


Fig. 7 Error concept

여기서  $\rho_i$ 는 계수로  $10^6$ ,  $t_s$ 는 시간,  $OS$ 는 Overshoot,  $US$ 는 Undershoot,  $t_s$ 는 Settling time을 의미한다. 식 (5)는 전체 비용 함수를 나타낸다. 식 (6)에서 오차 적분은 Fig. 7의 영역과 같이 타겟에서 떨어져 있는 오차를 적분해서 구한 면적의 크기를 의미한다. 식 (6)은 부하가 가해진 시점으로부터 목표 엔진 회전속도와 실제 엔진 회전속도의 오차에 대한 비용, 식 (7)은 Overshoot에 대한 비용, 식 (8)은 Undershoot에 대한 비용이며 식 (9)는 회전속도가 안정화 되는데 걸리는 시간인 Settling time에 대한 비용이다. 위 식 들에서 사용된 목표 지표들은 본 연구를 위해 임의로 지정된 값들이며, 엔진의 동작 조건 및 사용 목적에 따라 바뀌어야 하는 설계변수들이다. 비용 2, 3, 4는 설정된 한계치를 넘을 때만 비용이 발생한다. 비용1만 존재할 경우 전체 오차의 감소만 비용에 반영되고 큰 Overshoot이나 Undershoot이 발생하여도 전체 오차가 작다면 비용이 작아질 것이다. 하지만 큰 Overshoot과 Undershoot, 긴 Settling time은 엔진 제어에서 위험성을 가진다. 이를 예방하기 위해서 비용 2, 3, 4를 설정하였다. 비용 2는 Overshoot을 예방하기 위한 비용이다. 이때 원하는 Overshoot 크기를 넘지 않을 때는 비용이 0이 되고 Overshoot 크기가 원하는 크기를 넘을 경우 넘은 크기에 비례한 비용이 발생한다. 이를 통해 Overshoot을 예방한다. Undershoot과 Settling time의 비용 함수도 Overshoot과 동일한 설정을 통해 원치 않는 Undershoot과 긴 Settling time이 발생하는 것을 예방한다. 비용 2, 3, 4의 합의 크기와 비용 1의 크기가 차이가 클 경우 최적화 과정에서 의도한 대로 동작하지 않을 수 있다. 이를 예방하기 위해 계수를 매뉴얼 튜닝하여 비용 함수를 구성하였다.

이때 최적화 변수는 P-gain, I-gain, D-gain이며 이 값들은 0이하의 값을 갖지 않도록 제한조건을 설정하였다.

### 2.2.3 계인 최적화 알고리즘

계인 최적화 알고리즘은 Fig. 8에 나타난 순서로 동작한다. 먼저 엔진에 가해지는 부하에 변화를 주고 이때의 엔진을 모델 ID한다. 이렇게 구해진 모델과 PSO를 사용하여 최적의 계인을 계산한다. 위의 과정을 부하에 따라 반복하면서 계인을 모두 구한다. 구해진 계인을 맵으로 제작하면 계인 최적화 알고리즘의 동작이 끝난다. 이러한 방식으로 구해진 계인 맵을 기반으로 부하에 따라 계인을 변경하는 방식으로 PID제어기가 동작한다.

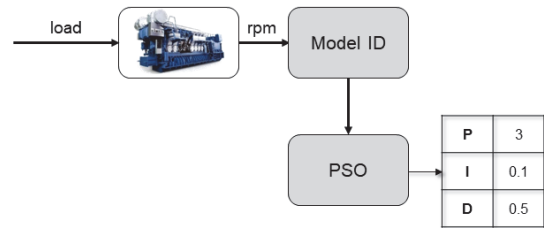


Fig. 8 Gain optimization algorithm

## 3. 검증

앞서 설명한 방법을 검증하기 위해 계인 맵 제작과 그 결과의 확인은 Fig. 9와 같은 순서로 진행했다. 먼저 엔진 모델 ID를 수행하고 계인 최적화 알고리즘을 동작시켜 최적의 계인 맵을 제작한다. 이 결과를 엔진 시뮬레이션 모델에 연결된 PID제어기에 적용하여 각 부하에서 엔진을 동작 시켜보고 그 결과를 통해 최적화 알고리즘을 검증했다.

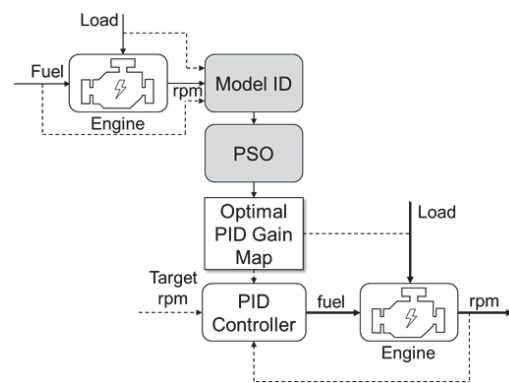


Fig. 9 Validation diagram

### 3.1 대상 엔진

검증을 위한 엔진으로는 SCANIA 차량 디젤엔진 시뮬레이션 모델<sup>9)</sup>을 사용하였다. 모델은 4행정 디젤 엔진으로 6기통, 12.4 L, 400 ~ 2,000 rpm의 회전속도 영역을 가



진다. 이 엔진 시뮬레이션 모델은 가스의 흐름 동역학과, 흡입관에서 압력 형성, 터보차저, EGR이 구현되어 시뮬레이션 상에서 엔진의 반응을 확인하기 위해 구성되었다.

### 3.2 알고리즘 이용 게인 맵 생성 결과

게인 맵은 Tables 1~3에서 확인할 수 있다. 게인 맵은 각 부하별 게인을 구하고 제작하였다. 각 부하 사이의 게인은 선형 보간을 이용하여 구할 수 있다. 알고리즘을 통해 선정된 게인 맵을 시뮬레이션을 통해 검증할 것이다. 이때 기존의 매뉴얼 튜닝과 게인 맵의 시뮬레이션 결과를 비용 함수 기준으로 비교 매뉴얼 튜닝보다 더 나은 성능을 발휘하는 것을 확인해 볼 것이다.

Table 1 Gain map when 1,000 rpm

Load [%]	Kp	Ki	Kd
0	0.2798	0.2012	0.2395
33	0.5424	0.1711	0.1002
66	0.5274	0.1693	0.1011
100	0.5021	0.1627	0.1131

Table 2 Gain map when 1,500 rpm

Load [%]	Kp	Ki	Kd
0	0.7410	0.2413	0.3128
33	0.5481	0.1723	0.1171
66	0.5540	0.1764	0.1249
100	0.5536	0.1769	0.1241

Table 3 Gain map when 1,800 rpm

Load [%]	Kp	Ki	Kd
0	0.6805	0.2346	0.4776
33	0.5136	0.3473	0.2301
66	0.8826	0.2221	0.3364
100	0.8277	0.2337	0.7798

### 3.3 시뮬레이션 검증

엔지니어가 매뉴얼로 튜닝한 게인과 최적화 알고리즘을 통해 구한 게인 맵의 결과를 비교하여 최적화 알고리즘이 적합한 게인을 찾아주는 지 확인하는 방법을 사용했다.

Figs. 10~12의 경우 각각 다른 회전속도를 가지지만 동일하게 부하가 100 %에서 0 %로 감소하는 경우의 시뮬레이션 결과이다. 게인 맵의 게인을 사용한 결과와 매뉴얼로 튜닝한 게인을 사용한 결과를 확인할 수 있다. 알고

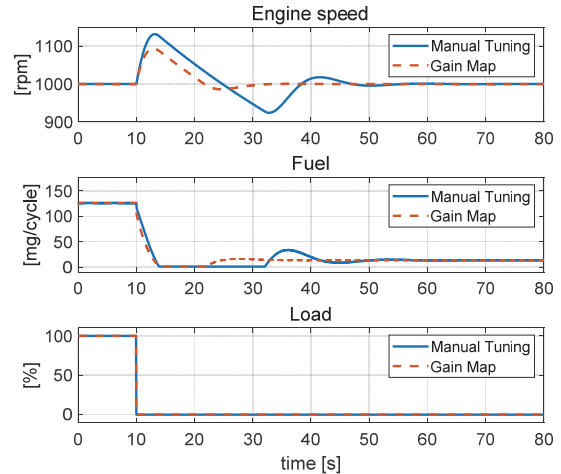


Fig. 10 Simulation result when 1,000 rpm, load 100 to 0 %

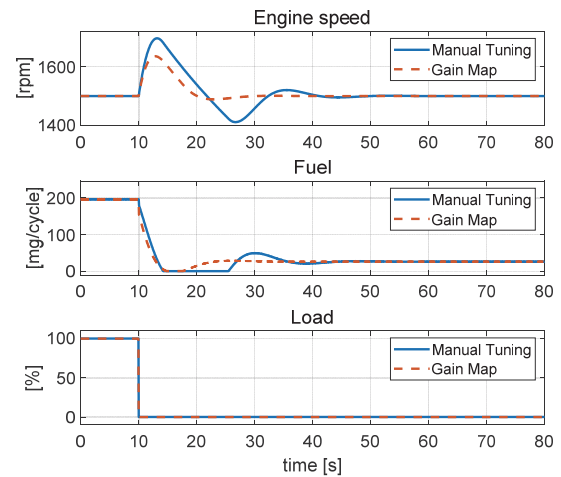


Fig. 11 Simulation result when 1,500 rpm, load 100 to 0 %

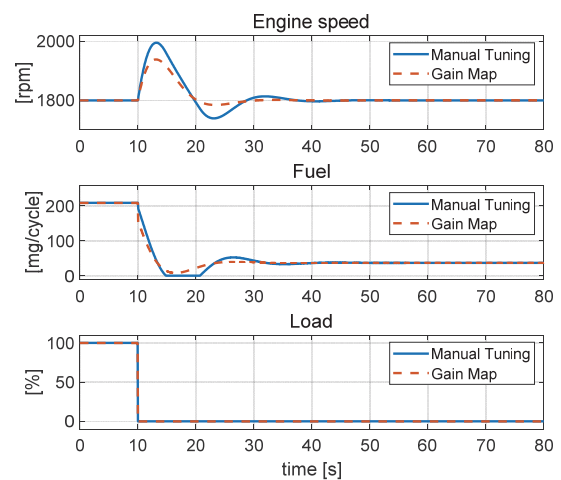


Fig. 12 Simulation result when 1,800 rpm, load 100 to 0 %

Table 4 Cost table

	Case1		Case2		Case3	
	Manual	Gain map	Manual	Gain map	Manual	Gain map
Cost <sub>1</sub>	1.1	0.4	1.1	0.4	0.8	0.4
Cost <sub>2</sub>	27	1.8	25.9	1	8.4	0
Cost <sub>3</sub>	0	0	0	0	0	0
Cost <sub>4</sub>	338.5	9	140.4	0	41.6	0

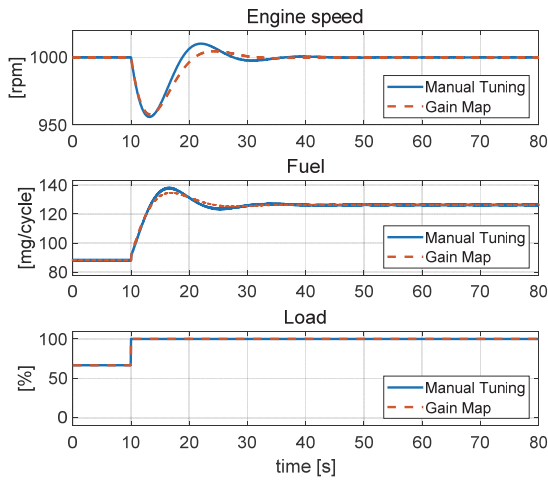


Fig. 13 Simulation result when 1,000 rpm, load 66 to 100 %

리즘을 통해 찾은 게인이 기존의 매뉴얼 튜닝보다 엔진 회전속도를 더 빠르게 안정 영역으로 제어하는 것을 확인할 수 있다. 이를 알고리즘에 적용한 비용 함수를 기준으로 알고리즘이 제대로 동작하고 있는 것을 확인해보았다.

Table 4의 세가지 경우에서 매뉴얼 튜닝과 게인 맵의 비용 변화를 보여준다. 이때 cost<sub>1</sub>은 전체 오차의 비용, cost<sub>2</sub>는 Overshoot의 비용, cost<sub>3</sub>는 Undershoot의 비용, cost<sub>4</sub>는 Settling time의 비용을 의미하고 이때 크기는 10<sup>6</sup>이다. Case1의 경우 Fig. 10의 비용 변화를 나타낸 것이다. 이때의 결과를 확인해보면 모든 비용은 감소하고 특히 Settling time의 비용이 크게 감소하였는데 Settling time이 개선된 것을 그림에서도 확인할 수 있다. Case2의 경우 Fig. 11의 비용 변화를 나타내고 모든 비용이 감소한 것을 확인할 수 있다. Case3의 경우 Fig. 12의 비용을 나타내고 있다. 이 경우 게인 맵에서 Overshoot, Undershoot, Settling time이 모두 설정한 범위 내로 수렴하여 0의 비용을 가지는 것을 확인할 수 있다.

Figs. 13~15에서는 부하가 66 %에서 100 %로 증가하는 경우 게인 맵의 게인을 사용한 결과를 확인할 수 있다. 이 경우에도 앞서 검증한 상황과 같이 매뉴얼 튜닝에 비해 더 작은 Undershoot과 안정 시간을 보여준다. 앞선

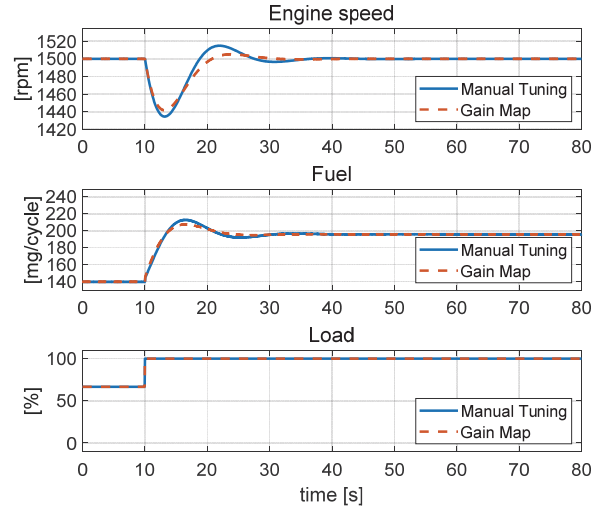


Fig. 14 Simulation result when 1,500 rpm, load 66 to 100 %

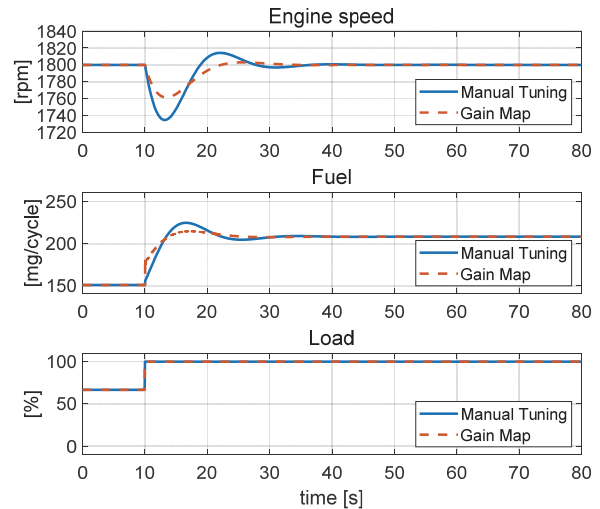


Fig. 15 Simulation result when 1,800 rpm, load 66 to 100 %

검증 과정을 통해 알고리즘이 매뉴얼 튜닝에 비해 회전 속도에 관계없이 더 우수한 성능을 보여주는 것을 확인할 수 있다.

#### 4. 결론

본 논문에서는 PID 제어기의 제어 게인을 자동으로 찾아서 제어 게인 맵을 만들어주는 알고리즘을 설계, 검증하였다. 검증 결과를 통해 매뉴얼 튜닝과 비교해 성능의 강건성을 확인할 수 있었다. 이를 통해 제안된 알고리즘을 사용하여 엔진 생산 시 초기 매뉴얼 튜닝 과정을 자동화 하고 이를 통해 생산시간의 단축, 생산된 엔진 제어기의 강건성을 확보할 수 있을 것이다.

## 후 기

이 과제는 부산대학교 기본연구지원사업(2년)에 의하여 연구되었음.

### References

- 1) S. H. Kang, S. Y. Oh, M. K. Lee, H. M. Kim and J. H. Kang, "Development of Closed-loop Controller for NH<sub>3</sub> Slip Reduction on Urea-SCR System," Transactions of the KSAE, Vol.29, No.2, pp.103-109, 2021.
- 2) D. E. Kim, B. J. Kwon, M. H. Shin and G. G. Jin, "Design of a Self-tuning PID Controller for the Speed Control of Marine Diesel Engines Using Gas," Proceedings of KFIS 2002 Fall Conference, pp.75-79, 2002.
- 3) S. S. Park, H. S. Lee, D. E. Kim and G. G. Jin, "Speed Control of Marine Diesel Engines Using Fuzzy Gain Scheduling," Journal of Advanced Marine Engineering and Technology, Vol.26, No.6, pp.638-645, 2002.
- 4) N. Farouk, L. Sheng and L. Said, "Speed Control System on Marine Diesel Engine Based on a Self-tuning Fuzzy PID Controller," Research Journal of Applied Sciences, Engineering and Technology, Vol.4, No.6, pp.686-690, 2012.
- 5) Z. C. Zhou and R. Chen, "Design on Fuzzy Neural Network PID Control System of Diesel Engine," Advanced Materials Research, Vol.756, No.4, pp.425-429, 2013.
- 6) I. J. Kim, J. C. Lee, J. Y. Choi and J. H. Kim, "Identification of Optimal Control Parameters for a Pneumatic Active Engine Mount System," Transactions of KSAE, Vol.20, No.2, pp.30-37, 2012.
- 7) M. R. Bonyadi and Z Michalewicz, "Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review," Evolutionary Computation, Vol.25, No.1, pp.1-54, 2017.
- 8) M. I. Solihin, L. F. Tack and M. L. Kean, "Tuning of PID Controller Using Particle Swarm Optimization (PSO)," Proceeding of the International Conference on Advanced Science, pp.458-461, 2011.
- 9) J. Wahlström and L. Eriksson, "Modelling Diesel Engines with a Variable-geometry Turbocharger and Exhaust Gas Recirculation by Optimization of Model Parameters for Capturing Non-linear System Dynamics," Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, Vol.225, No.7, pp.960-986, 2011.