



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

碩士學位論文

신경망 제어기를 이용한  
풍력발전시스템의 피치제어

濟州大學校 大學院

風力特性化協同科程

洪 民 浩

2014年 8月

신경망 제어기를 이용한  
풍력발전시스템의 피치제어

指導教授 康 珉 齊  
洪 民 浩

이 論文을 工學 碩士學位 論文으로 提出함

2014年 6月

洪民浩의 工學 碩士學位 論文을 認准함

審査委員長 ----- ①  
委 員 ----- ①  
委 員 ----- ①

濟州大學校 大學院

2014 年 6月

# Pitch Angle Control of Wind Turbine System Using Neural Network Controller

Min-Ho Hong

(Supervised by professor Min-Jae Kang)

thesis submitted in partial fulfillment of the  
requirement for the degree of Master of Engineering

Multidisciplinary Graduate School Program for Wind Energy

GRADUATE SCHOOL

JEJU NATIONAL UNIVERSITY

2012. 08.

목 차	
Abstract .....	1
제 1 장 서론.....	2
제 2 장 풍력발전 시스템의 구성.....	5
2.1 Aerodynamics .....	5
2.2 Drive train .....	7
2.3 Generator .....	9
2.4 Controller .....	10
제 3 장 신경망 피치제어기 .....	12
3.1 피치제어를 위한 신경망.....	12
3.2 풍속과 발전기의 회전속도를 고려한 신경망 피치제어.....	14
3.2.1 알고리즘 .....	14
3.2.2 훈련요소 .....	16
3.3 발전기의 회전속도 변화율을 고려한 신경망 피치제어 .....	18
3.3.1 알고리즘 .....	18
3.3.2 훈련요소 .....	20
제 4 장 시뮬레이션 및 결과 .....	22
4.1 풍력발전기의 Matlab/Simulink 모델 .....	22
4.1.1 Aerodynamics model .....	23
4.1.2 Drive Train model .....	23
4.1.3 Generator model .....	24
4.1.4 Torque Controller model .....	24
4.1.5 Pitch actuator .....	25

4.2 신경망 제어기의 훈련을 위한 Matlab/Simulink 모델.....	26
4.3 신경망 피치제어시스템의 Matlab/Simulink 모델.....	31
4.3.1 풍속과 발전기의 회전속도를 고려한 신경망 피치제어시스템 .....	31
4.3.2 발전기의 회전속도 변화율을 고려한 신경망 피치제어시스템 .....	35
제 5 장 결 론.....	39
참고 문헌 .....	41

## List of Figures

Fig. 1 Annual installed global capacity 1996–2012(GWEC).....	2
Fig. 2 Block Diagram of Wind Turbine System .....	5
Fig. 3 Cp curve of NREL 5MW wind turbine .....	7
Fig. 4 Drive train model .....	8
Fig. 5 System Diagram of a DFIG-Based Wind Turbine System.....	9
Fig. 6 Ideal Power Curve of Wind Turbine .....	10
Fig. 7 Structure of Neural Network Controller .....	12
Fig. 8 Block diagram of wind turbine using NNC .....	14
Fig. 9 Flow chart of pitch control consider wind and generator speed .....	15
Fig. 10 Block diagram of wind turbine using NNC .....	18
Fig. 11 Flow chart of pitch control consider changing rate of generator speed .....	19
Fig. 12 Matlab/Simulink block diagram of NREL 5MW wind turbine .....	22
Fig. 13 Matlab/Simulink block diagram of Aerodynamics .....	23
Fig. 14 Matlab/Simulink block diagram of Drive Train .....	23
Fig. 15 Matlab/Simulink block diagram of Generator .....	24
Fig. 16 Matlab/Simulink block diagram of Torque Controller.....	24
Fig. 17 Matlab/Simulink block diagram of Pitch actuator .....	25
Fig. 18 Block diagram of Matlab/Simulink for Error back-propagation algorithm .....	26
Fig. 19 Error back-propagation algorithm .....	27
Fig. 20 Block diagram of Matlab/Simulink for Neural Network Controller .....	28

Fig. 21 Test outputs of NNC consider wind and generator speed for untrained region .....	29
Fig. 22 Test outputs of NNC consider changing rate of generator speed for untrained region .....	30
Fig. 23 Block Diagram of Matlab/Simulink for wind turbine system with NNC consider wind and generator speed .....	31
Fig. 24 Simulation results of wind turbine system with NNC consider wind and generator speed (step wind speed) .....	33
Fig. 25 Simulation results of wind turbine system with NNC consider wind and generator speed (random wind speed) .....	34
Fig. 26 Block Diagram of Matlab/Simulink for wind turbine system with NNC consider changing rate of generator speed .....	35
Fig. 27 Simulation results of wind turbine system with NNC changing rate of generator speed (step wind speed) .....	37
Fig. 28 Simulation results of wind turbine system with NNC changing rate of generator speed (random wind speed) .....	38



## List of Table

Table. 1 Inputs and desired output of NNC consider wind and generator speed .....	17
Table. 2 Inputs and desired output of NNC consider Changing Rate of Generator Speed .....	21

## Abstract

Wind turbine system can obtain the maximum wind energy using torque control under the rated wind speed, and wind turbine power is controlled as the rated power using pitch control over the rated wind speed. In this paper, we present two methods for wind turbine pitch controller using neural networks. The first method is that the difference between a rated and current speed of generator has been used for another input of neural network controller as well as wind speed. As a result, the generator speed and power are maintained as the rated ones. However vibrations of pitch angle and generator speed have been considered to be improved. And the second methods is that the changing rate of generator speed has been used for another input of neural network controller as well as the difference between the rated and current generator speed. As a result, problems of the first method have been improved. Error back-propagation algorithm is used for training the neural network pitch controller and Matlab/Simulink is used to confirm the performance of proposed method.

## 제 1 장 서 론

인류의 문명은 에너지의 개발과 이용에 의하여 촉진되고 지탱되었다고 해도 지나침이 없다. 인류는 원시시대 이래 불을 발견하여 이용함으로써 생활을 보다 편리하게 영위함과 동시에 많은 문화유산을 창조할 수 있었다. 하지만 오늘날 전 세계적으로 화석연료의 사용량 급증으로 인한 환경문제가 많은 국가들의 관심사가 되었고 앞으로도 계속 증가할 전망이다. 재생에너지의 일종인 풍력은 자원이 풍부하고 그 재생력이 무한하고, 광범위한 지역에 분포되어 있으며 환경오염이 없다는 점에서 화석연료의 사용량이 급증하고 있는 이 시대에 앞으로의 미래를 대비할 유망한 대체 에너지원으로써 기대되는 에너지원이다. 또한 풍력발전은 자연에너지인 바람을 이용하여 발전하기 때문에 바람이 불 때에는 수요에 관계없이 동작만 가능하다면 반드시 전력을 생산한다는 점에서 매우 유용하다. IEA(국제에너지기구)의 “World Energy Outlook 2012“ 보고서의 향후 25년간의 에너지 전망에 따르면 2035년경의 세계 에너지 수요는 30%정도 증가할 것으로 예측하고 있으며 신재생에너지는 전 세계 총 발전량의 1/3 이상을 담당하게 될 것으로 전망하고 있다.[1] 에너지 소비량이 많은 우리나라 또한 친환경적인 새로운 에너지원을 개발하기 위하여 많은 노력을 기울이고 있고 풍력은 이러한 신재생에너지 중에서도 가장 기대되는 에너지원이다.

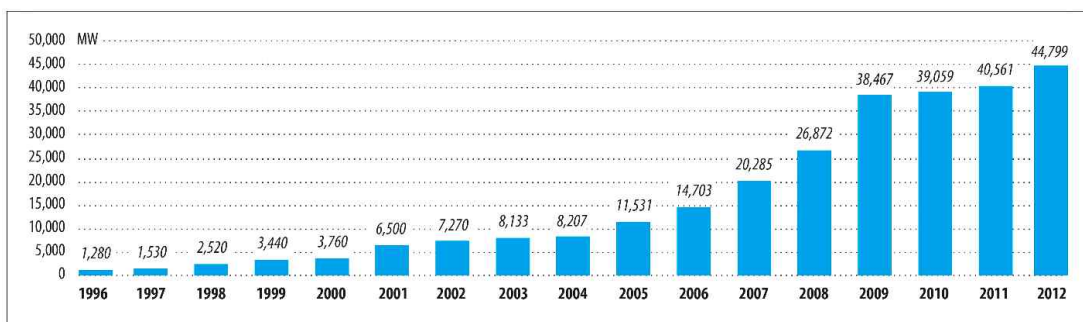


Fig.1 Annual installed global capacity 1996-2012(GWEC)

Fig. 1에 연도별 설치된 세계풍력 발전 용량을 살펴보면 그 설치 용량은 항상 증가하고 있는 것을 볼 수 있다. 처음 개발 단계에서는 설치 용량이 미미했지만 2005년부터 급격히 증가하기 시작했고 기술이 발전하고 풍력 발전기의 용량이

커지면서 현재는 한해에 40[GW]가 넘는 정도의 풍력 발전기들이 설치되고 있는 것을 볼 수 있다. 현재 육상뿐만 아니라 해상풍력에 대한 관심과 연구도 지속되고 있으므로 앞으로 한동안 풍력발전기의 설치는 계속 될 것으로 보이며 전 세계적으로 각 국가마다 앞으로의 계획을 세워 그 용량을 늘려갈 것으로 보인다.

풍력발전기의 수명은 20년 이상으로 한번 설치가 되면 안전을 최우선으로 하며, 그 효율을 최대한으로 이끌어 내어 최대한 많은 전력을 생산해야 한다. 이를 위하여 효율적인 제어전략이 필수적이다.

풍력발전기의 제어는 불어오는 바람의 세기에 따라 크게 두 가지의 경우로 나눌 수 있는데 발전기가 정격출력을 내는 풍속인 정격풍속 이하에서의 제어와 정격풍속 이상에서의 제어로 나눌 수 있다. 정격풍속 이하에서는 풍력발전기의 피치각을 0으로 고정시키고 발전기의 토크를 제어하여 전력 계수인  $C_p$ 를 최대의 값으로 유지시켜 바람의 에너지를 최대한 많이 받을 수 있도록 해야 한다. 풍력발전기의 전력 계수인  $C_p$ 는 이론상으로는 바람이 갖는 에너지의 59.3%까지 전기 에너지로 변환시킬 수 있지만, 현실적으로 기계적 마찰, 발전기의 효율, 블레이드(blade)의 형상 등 손실요인이 존재하기 때문에 실제 효율은 50%를 넘지 못한다. 정격풍속 이상에서의 풍력발전기는 이미 출력이 정격출력에 도달하였기 때문에 이를 유지하기 위하여 발전기의 토크를 일정하게 유지시키고 피치각을 제어하여 발전기의 회전속도를 조절한다. 풍력발전기를 구성하는 로터(rotor)나 발전기, 기어박스(gear box) 등 각각의 요소들을 선형모델로 가정하더라도 풍력 발전기의 전체적인 작동은 비선형적인 면을 보이게 된다. 때문에 풍력 발전기의 피치제어기는 주로 비선형적인 형태로 설계되어 실제 풍력발전기에 적용되고 있다.[2]

이러한 비선형적인 제어기 설계를 위하여 신경망 제어기의 비선형특성을 이용하여 풍력발전기의 피치각을 제어하는 방안을 제시하였다. 신경망 제어기는 단순한 기능을 가진 많은 수의 간단한 프로세서들이 서로 병렬 연결된 구조를 가지고 있으며 입력값과 출력값 사이가 매우 복잡한 형태의 비선형 모형화 방법으로 그 종류 또한 다양하다.

본 논문에서 사용된 신경망 제어기의 종류는 연결된 구조에 루프가 없어 주어진 입력에 대해 같은 출력만을 나타내는 앞먹임(feedforward)신경망 중에서도 가

장 많이 쓰이는 다층인식자(multi-layer perceptron, MLP) 신경망을 사용하였다. 사용된 신경망은 두 개의 입력과 하나의 출력을 가지고 있다. 입력이 두 개인 이유는 신경망 제어기의 입력이 하나이면 그만큼 훈련 오차가 거의 0에 가깝게 되고 정확성 또한 증가하게 되지만 입력에 대한 출력이 2차원적인 그래프가 되고 결국 정확한 제어를 하지 못하게 된다. 그렇기 때문에 두 개의 입력을 갖는 신경망 제어기를 사용하였고 출력은 피치각의 다음 변화량을 갖는 신경망 제어기를 설계하였다. 우선 풍속을 하나의 입력으로 하고 발전기의 정격 회전속도와 현재 회전속도의 차이를 또 하나의 입력으로 한 신경망 제어기를 설계하였다. 이때의 제어 전략으로는 발전기의 정격 회전속도와 현재 회전속도의 차이를 구하여 이 값이 양수이면 현재 발전기 회전속도가 정격보다 낮다는 말이므로 발전기 회전속도를 높이기 위해 풍속에 맞게 무조건 피치각을 감소시키고 발전기 회전속도의 차이가 음수이면 발전기 회전속도를 줄이기 위해 풍속에 맞게 무조건 피치각을 증가시켜 발전기의 출력을 제어하는 것이다. 다음으로 발전기의 정격 회전속도와 현재 회전속도의 차이를 하나의 입력으로 하고 발전기 회전속도의 변화율을 또 하나의 입력으로 한 신경망 제어기를 설계하였다. 이때의 제어 전략으로는 발전기의 정격 회전속도와 현재 회전속도의 차이에 발전기 회전속도의 변화율을 첨가함으로써 발전기 회전속도의 차이가 양수일지라도 현재 발전기 회전속도가 감소중이라면 피치각을 감소시키지 않고 그대로 유지하도록 하고 발전기 회전속도가 증가중이거나 변화가 없을 때에 피치각을 감소시켜 발전기의 출력을 제어하는 것이다.

제어기에 사용된 신경망은 다층인식자(multi-layer perceptron, MLP) 신경망 중에서도 오류역전파(Error back-propagation)가 사용되었고 여기에서 사용된 풍력 발전기 모델은 NREL에서 제공하는 5MW의 풍력 발전기이며 시뮬레이션은 Matlab/simulink에서 수행되었다.

## 제 2 장 풍력발전 시스템의 구성

일반적인 풍력 발전 시스템의 전체적인 구성을 블록도로 나타내면 Fig. 2와 같다.

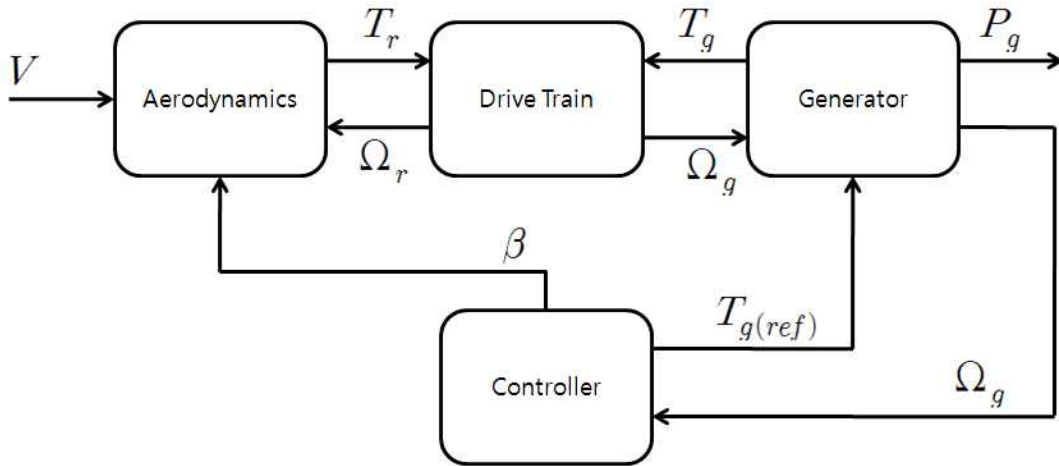


Fig. 2 Block Diagram of Wind Turbine System

풍속인 바람  $V$ [m/s]가 Aerodynamics으로 들어가면 공기 역학적 토크  $T_r$ 을 계산하여 Drive Train으로 보내준다 Drive Train은 이 공기 역학적 토크를 통해서 발전기의 토크  $T_g$ 와 발전기의 회전속도  $\Omega_g$ 를 계산하게 되고 계산된 발전기의 회전속도  $\Omega_g$ 가 Generator로 들어감으로써 발전기의 출력인  $P_g$ 가 계산된다. 그리고 풍력 터빈 시스템의 제어기인 Controller는 풍력 터빈 시스템의 풍속  $V$ [m/s]가 정격풍속 미만일 때에는  $T_{g(ref)}$ 를 통하여 발전기의 토크를 제어함으로써 항상 최대 출력을 갖게 하고 정격풍속 이상일 때에는 로터의 피치각인  $\beta$ [deg]를 제어하여 발전기의 회전속도를 정격 회전속도로 유지함으로써 발전기의 출력을 일정하도록 제어한다.

### 2.1 Aerodynamics

풍력 터빈 시스템은 바람이 가진 운동에너지를 전기에너지로 변환하는 시스템

이다. 일정한 면적을 통과하는 바람이 가진 운동에너지는 다음과 같이 식으로 표현 가능하다.

$$P_t = \frac{1}{2} \rho A v^3 [W] \quad \text{식 (2-1)}$$

식 (2-1)에서  $\rho$ 는 공기밀도 [ $kg/m^3$ ]를 나타내고  $A$ 는 일정한 면적의 단면적 [ $m^2$ ],  $v$ 는 풍속 [ $m/s$ ]를 나타낸다. 본 논문에서 사용된 공기밀도는  $1.223[kg/m^3]$ 이다. 이 바람이 풍력 터빈의 로터를 통과할 때 풍력발전기가 받는 기계적 에너지는 다음과 같다.[3,4]

$$P_r = \frac{1}{2} \rho A v^3 C_p(\lambda, \beta) [W] \quad \text{식 (2-2)}$$

식 (2-2)에서  $C_p(\lambda, \beta)$ 는 전력 계수(power coefficient)를 나타내며 블레이드 (blade)의 특성에 따라 달라진다. 전력 계수인  $C_p(\lambda, \beta)$ 는 주속비  $\lambda$ 와 피치각  $\beta$ 의 함수이며 여기서 주속비  $\lambda$ 는 현재 불어오는 바람에 대한 블레이드(blade) 끝단속도의 비율로 다음과 같이 표현된다.

$$\lambda = \frac{\Omega_r R}{v} \quad \text{식 (2-3)}$$

$C_p(\lambda, \beta)$ 는 다음과 같은 수식으로 근사화 시킬 수 있다.

$$C_p = 0.5176 \left( \frac{116}{\lambda_i} 0.4\beta - 5 \right) e^{-\frac{21}{\lambda_i}} + 0.0068\lambda \quad \text{식 (2-4)}$$

$$\frac{1}{\lambda_i} = \frac{1}{\lambda + 0.08\beta} - \frac{0.035}{\beta^3 + 1} \quad \text{식 (2-5)}$$

위의 수식 말고도  $C_p(\lambda, \beta)$ 를 근사화 시킨 수식은 여러 종류가 있다. 앞에서 말했듯이  $C_p(\lambda, \beta)$ 는 블레이드(blade)의 특성에 따라 다르게 나타난다. 본 논문에서는 NREL에서 제공하는 5MW의  $C_p(\lambda, \beta)$ 를 사용하였으며, Fig. 3에 나타나 있다.

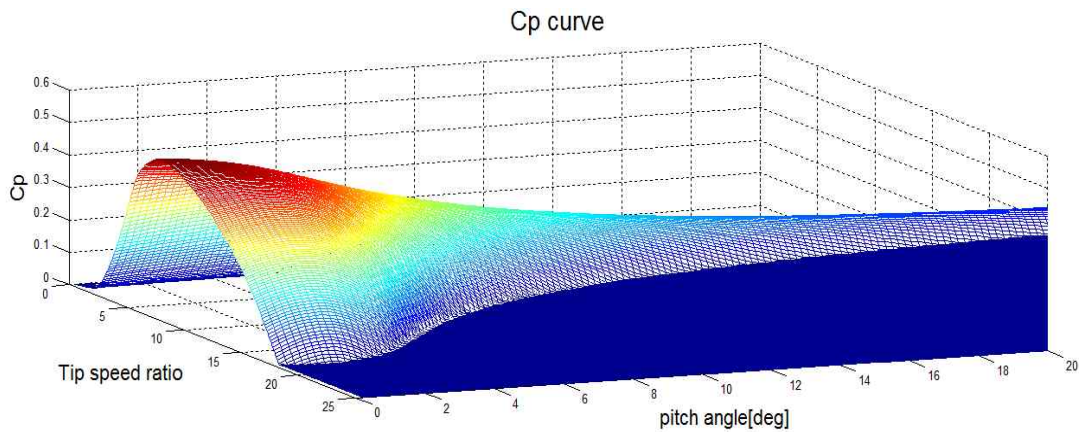


Fig. 3 Cp curve of NREL 5MW wind turbine

식 (2-2)의 공기역학적 출력  $P_r [W]$ 를 로터의 각속도로 나누면 공기 역학적 토크인  $T_r [Nm]$ 를 구할 수 있다.

$$T_r = \frac{1}{2\Omega_r} \rho A v^3 C_p(\lambda, \beta) [Nm] \quad \text{식 (2-6)}$$

## 2.2 Drive train

본 논문에서 동력 전달 장치 모델은 2-mass 모델을 사용하였으며 Fig. 4에 그 모델을 간단하게 나타내었다.[5] 동력 전달 모델은 바람을 직접 받아 회전하는 저속 회전축(Low speed shaft)의 토크  $T_r [N]$ 과 고속 회전축(High speed shaft)의 토크



$T_g$ [N]를 이용하여 수식을 통해 발전기 회전속도  $\Omega_r$ [rad/s]과 로터의 회전속도  $\Omega_g$ [rad/s]를 계산하는데 사용된다.

로터의 회전에 의하여 공기역학적 토크가 발생하게 되는데 이는 식 (2-7)과 같다. Fig. 4와 식 (2-7)을 이용하여 운동방정식을 적용하면 식 (2-8)을 유도할 수 있다.

$$J_r \frac{d\Omega_r}{dt} = T_r - K_s \left( \theta_r - \frac{1}{N_g} \theta_g \right) - B_r \left( \Omega_r - \frac{1}{N_g} \Omega_g \right) \quad \text{식 (2-7)}$$

$$J_g \frac{d\Omega_g}{dt} = \frac{K_s}{N_g} \left( \theta_r - \frac{1}{N_g} \theta_g \right) + \frac{B_r}{N_g} \left( \Omega_r - \frac{1}{N_g} \Omega_g \right) - T_g \quad \text{식 (2-8)}$$

여기서  $K_s$ 는 동력 전달축의 비틀림 상수이고,  $B_r$ 은 동력 전달축의 비틀림 댐핑 상수이다. 그리고  $N_g$ 는 기어박스(gear box)의 기어비이고,  $\theta_r, \theta_g$ 는 각각 저속 회전축(LSS)와 발전기축의 고속 회전축(HSS)의 각도를 의미한다. 또한  $J_r$ 는 저속 회전축과 고속 회전축의 각속도를 의미하며,  $J_r$ 은 3개의 블레이드(blade), 허브 그리고 저속 회전축(low speed shaft, LSS)의 효과를 모두 고려한 관성 질량(moment of inertia)이고,  $J_g$ 는 발전기 회전축의 관성 질량을 의미한다.

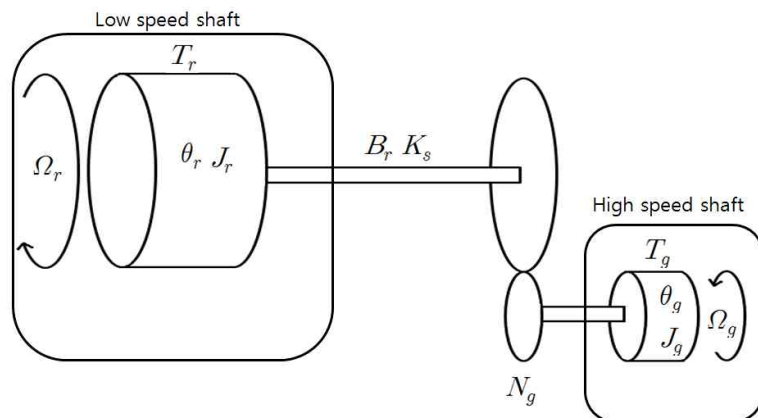


Fig. 4 Drive train model

## 2.3 Generator

풍력 발전 시스템에서 가장 흔하게 쓰이는 육상 발전기는 DFIG(Doubly-Fed Induction Generator)이다.

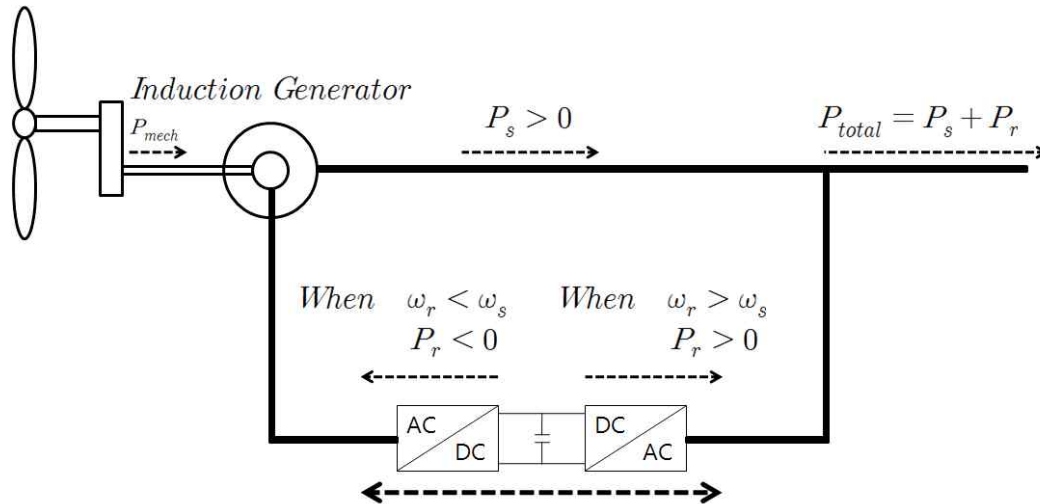


Fig. 5 System Diagram of a DFIG-Based Wind Turbine System

Fig. 5에 DFIG(Doubly-Fed Induction Generator)를 기반으로 한 풍력발전 시스템을 나타내었다.[6]  $P_s$ 는 고정자의 전력이고  $P_r$ 은 회전자의 전력을 나타내며  $\omega_r$ 은 회전자의 회전속도, 그리고  $\omega_s$ 는 고정자의 회전속도를 나타낸다. 그리고  $P_{mech}$ 는 풍력발전기로 받는 기계적 파워를 나타내고  $P_{total}$ 은 고정자의 전력과 회전자의 전력을 합한 총 전력을 나타낸다.

DFIG(Doubly-Fed Induction Generator)의 특징 중 하나는 바로 회전자가 그리드(grid)와 양방향으로 연결되어 있다는 점이다. 가운데 연결되어있는 AC/DC컨버터와 DC/AC컨버터를 제어하여 그리드(grid)와 전력을 주고받을 수 있는데, 고정자의 회전속도인  $\omega_s$ 가 회전자의 회전속도  $\omega_r$ 보다 클 경우에는 회전자 전력  $P_r$ 이 0보다 작아지게 된다. 이럴 경우에는 그리드(grid)로부터 전력을 받아 로터를 회전시키고 고정자의 회전속도인  $\omega_s$ 가 회전자의 회전속도  $\omega_r$ 보다 작을 경우에는 전력을 그리드로 전송하도록 제어한다.

실제의 육상 풍력발전기에서는 위와 같은 DFIG(Doubly-Fed Induction Generator)를 주로 쓰지만 피치제어의 연구와 같은 시뮬레이션에서의 발전기 모델은 간단하게 발전기 토크와 회전속도의 곱에 효율을 곱하는 정도로 적용하는 것이 일반적이다. 여기서 사용된 발전기 효율의 값은 0.95이다.

## 2.4 Controller

풍력발전기의 제어는 크게 두 가지 영역으로 나눌 수 있다. 첫 번째는 풍속이 정격풍속보다 낮은 영역에서 바람의 에너지를 최대한 많이 받기 위한 제어이다. 이 때는 피치각  $\beta[\text{deg}]$ 를 0으로 고정시키고 토크제어를 통해 주속비  $\lambda$ 를 최적으로 유지시켜  $C_p(\lambda, \beta)$ 를 최대로 만든다. 그리고 두 번째는 풍속이 정격풍속보다 높은 영역에서 발전기의 발전량을 정격으로 유지시키기 위한 제어이다. 정격풍속 이상에서는 로터의 회전속도를 일정하게 유지시키기 위해 피치각을 제어해야 한다.[7,8]

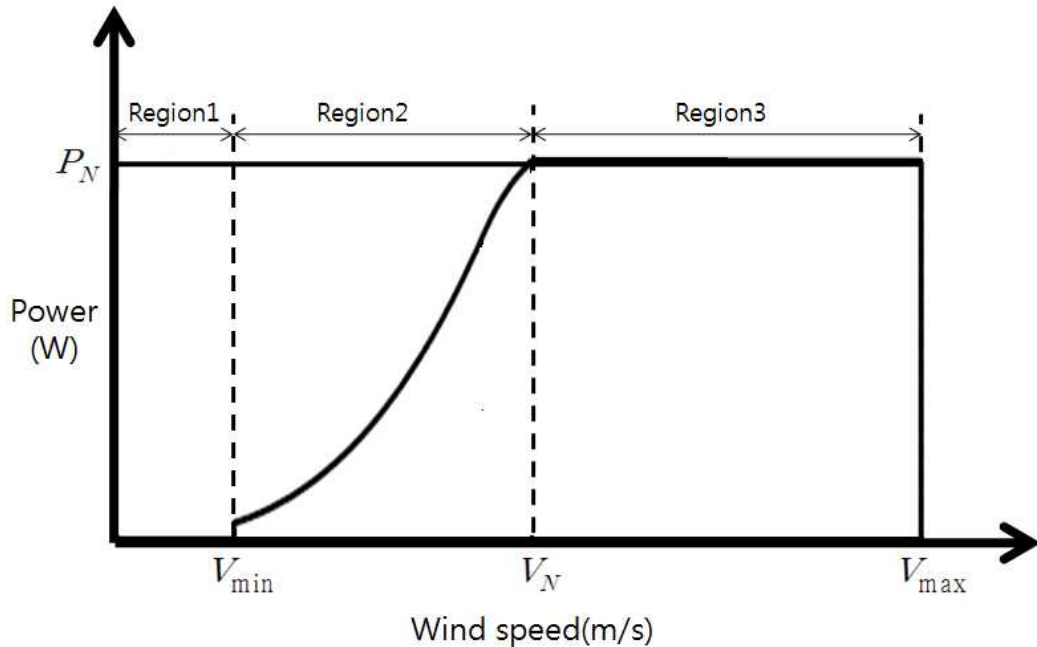


Fig. 6 Ideal Power Curve of Wind Turbine

Fig. 6은 풍력발전기의 풍속에 대한 전력의 이상적인 출력 곡선을 나타낸다. Region1은 시동풍속( $V_{\min}$ ) 이전의 부분으로 어느 정도의 바람이 불기 전까지는 풍력발전기를 동작시키지 않는 부분이다. Region2는 시동풍속과 정격풍속( $V_N$ )사이의 구간으로 풍력발전기가 받는 바람의 에너지를 최대한 많이 회수하도록 제어 되어야 하는 구간이다. 그렇기 때문에 이 영역에서의 그래프의 형태는 바람의 3승에 비례하는 곡선이 나타나며 이 영역에서는 피치각을 0으로 고정시키고  $C_p(\lambda, \beta)$ 의 값을 최대로 유지시키기 위해 주속비  $\lambda$ 를 일정하도록 제어해 주어야 한다. 이를 위하여 발전기 토크  $T_{g(ref)}[N]$ 를 이용하여 주속비가 일정하도록 제어하여 바람의 에너지를 최대한으로 받을 수 있도록 한다. Region3은 정격풍속에서 부터 종단풍속 사이의 영역으로 바람의 에너지가 발전기의 정격전력보다 큰 구간이다. 이 구간에서는 피치각을 제어하여 발전기의 회전속도를 일정하게 유지함으로써 발전기의 출력을 일정하도록 유지시킨다.

본 논문에서는 Region3 영역의 정격풍속 이상일 때 피치각을 제어하여 발전기의 출력을 일정하게 유지시키기 위한 신경망 피치제어기를 제안하였다. 총 두 가지 방법을 이용하여 신경망 피치제어기를 설계하였는데 첫 번째 방법은 풍속과 발전기 회전속도를 이용한 신경망 피치제어기이고 두 번째 방법은 발전기 회전속도의 변화율을 이용한 신경망 피치제어기이다.

## 제 3 장 신경망 피치제어기

### 3.1 피치제어를 위한 신경망

신경망은 단순한 기능을 가진 많은 뉴런들이 병렬 연결된 구조로 이루어져 있으며 그 종류 또한 아주 다양하다. 본 논문에서는 피드백이 없는 앞먹임(Feed-forward) 신경회로망 중에서 가장 많이 쓰이는 다층인식자(Multi-layer perceptron, MLP)의 한 종류인 오류 역전파(Error back-propagation) 신경망 제어기를 사용하였다. 다층인식자는 뉴런들이 층을 이루면서 배열된 구조를 지니며, 그 층들 사이에는 한 방향으로의 연결만 존재한다.

풍력발전기의 출력은 비선형성 특성을 가지고 있기 때문에 정격풍속 이상에서 풍력발전기의 발전기 회전속도를 일정하게 유지시키기 위하여 신경망 제어기를 이용해서 피치각을 제어하는 방법을 제안한다. 피치각을 제어하여 발전기 회전속도를 일정하게 유지시키는 것이 본 논문의 목적이므로 신경망 제어기의 출력은 한 개의 출력만을 다루었다.

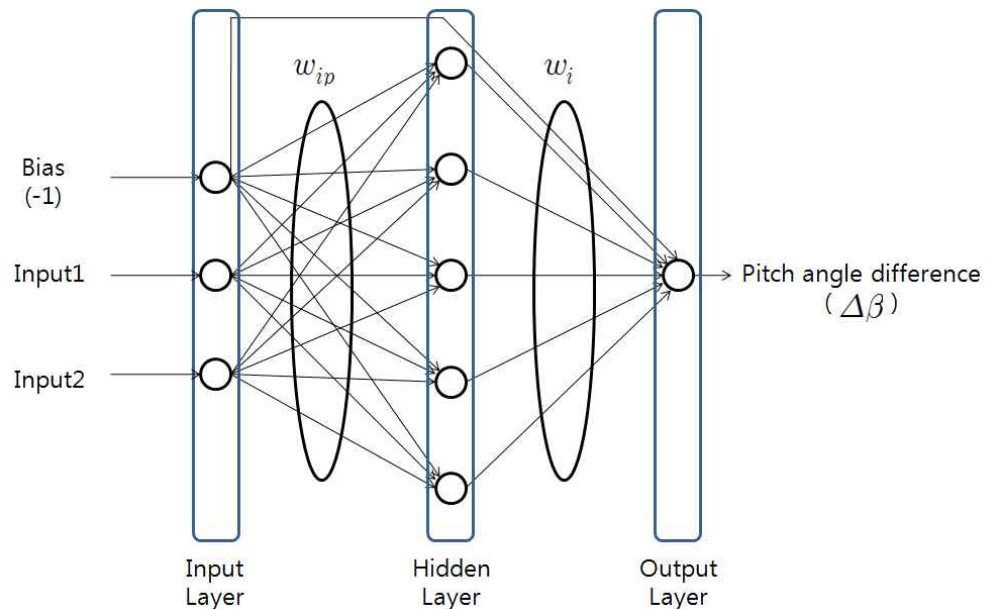


Fig. 7 Structure of Neural Network Controller

Fig. 7은 오류역전파(Error back-propagation) 신경망 제어기의 구조를 보여준다.[9,10] 본 논문에서 사용한 신경망 제어기의 구조는 Fig. 7에서와 같이 1개의 은닉층을 가지고 있으며 두 개의 입력과 한 개의 출력을 갖는다. 오류역전파 신경망에서는 효율과 안정적인 수렴을 위해 바이어스 뉴런이 입력층(Input layer)에 종종 사용되는데, 여기서도 -1의 값을 갖는 바이어스 뉴런이 입력층에 사용되었다. 각 층들을 연결하는 가중치는  $w_{ip}$ 와  $w_i$ 가 있는데  $w_{ip}$ 는 입력층의  $p$ 번째 뉴런과 은닉층의  $i$ 번째 뉴런을 연결하는 가중치를 나타내고  $w_i$ 는 은닉층의  $i$ 번째 뉴런이 뉴런이 하나뿐인 출력층(Output layer)과 연결하는 가중치를 나타낸다.

피치제어를 위한 신경망 제어기는 입력에 따른 출력이 목표한 값을 가질 수 있도록 뉴런들을 연결하는 가중치  $w_{ip}$ 와  $w_i$ 의 값을 찾아야 한다. 이를 위하여 임의의 값으로 가중치  $w_{ip}$ 와  $w_i$ 를 초기화하고 그 값을 통해 피치제어를 위한 신경망 출력의 오차를 계산하고 그 오차가 줄어들도록 가중치  $w_{ip}$ 와  $w_i$ 를 수정한다. 이 오차가 줄어들도록 가중치를 수정하는 식은 다음과 같다.

$$w_{ip}(t+1) = w_{ip}(t) + \frac{\alpha}{l} \sum_{n=1}^l X_{pn} f'_1(\text{net}_i) E_{in} w_i \quad \text{식 (3-1)}$$

$$w_i(t+1) = w_i(t) + \frac{\alpha}{l} \sum_{n=1}^l E_{in} f'_2(\text{net}) Z_{in} \quad \text{식 (3-2)}$$

$$E(n) = f_2(n) - \hat{Y}(n) \quad (n=1,2,\dots,l) \quad \text{식 (3-3)}$$

$$E_{total} = \frac{\sum(E(n))^2}{l} \quad (n=1,2,\dots,l) \quad \text{식 (3-4)}$$

식 (3-1)과 식 (3-2)에서  $X(n)$ 은 피치제어 신경망 입력들의 벡터이고  $Z_{in}$ 은 은닉층의 뉴런들의 벡터이다.  $f_2(n)$ 은 출력값의 변환함수,  $\hat{Y}(n)$ 은 목표값을 나타내며,  $f_1$ 은 은닉층의 변환함수를 나타낸다.  $\text{net}_i$ 와  $\text{net}$ 는 각각 은닉층  $i$ 번째 뉴런의 입력값과 출력뉴런의 입력값을 나타낸다. 그리고  $l$ 은 훈련 사이즈,  $\alpha$ 는 훈련 비율을 나타낸다. 은닉층 뉴런의 변환함수는 -1에서 1 사이의 값을 갖는 시그모이

드(Sigmoid)함수로 식 (3-5)과 같고 출력뉴런의 변환함수는 식 (3-6)과 같이 선형함수이다.

$$f_1(u) = \frac{1 - e^{-u}}{1 + e^{-u}} \quad \text{식 (3-5)}$$

$$f_2(u) = u \quad \text{식 (3-6)}$$

이와 같은 방법으로 전체적인 에러인 식 (3-4)가 원하는 값 이하가 될 때 까지 가중치의 수정과정을 반복하여 이에 만족하는 가중치  $w_{ip}$ 와  $w_i$ 를 구한다.

### 3.2 풍속과 발전기의 회전속도를 고려한 신경망 피치제어

#### 3.2.1 알고리즘

앞에서 설명 했듯이 Region 3에서는 풍력발전기의 출력을 정격으로 일정하게 유지되어야 한다. 이를 위하여 블레이드(blade)의 피치각을 제어하여 로터의 회전속도를 정격속도로 유지시켜야 한다.

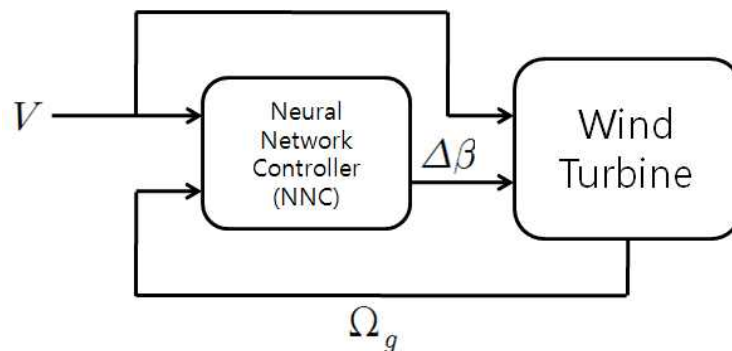


Fig. 8 Block diagram of wind turbine using NNC

본 장에서 제안하는 알고리즘은 현재 들어오는 풍속을 하나의 입력으로 받고 발전기의 정격 회전속도와 현재 회전속도의 차이를 또 하나의 입력으로 받아 정격 발전기 회전속도보다 현재 발전기 회전속도가 높으면 무조건 피치각을 증가

시켜 발전기의 회전속도를 낮추고 정격발전기 회전속도보다 현재 발전기 회전속도가 낮으면 피치각을 감소시켜 발전기의 회전속도를 높인다. 이런 제어알고리즘을 기반으로 신경망 모델을 훈련시키고 풍력발전기에 적용하여 제어하는 방식을 제안한다.

Fig. 8은 정격풍속 이상에서 신경망 제어기를 이용한 피치제어시스템의 블록다이어그램을 나타낸다. 이 시스템은 풍속  $V$ [m/s]가 풍력발전기로 들어가 현재 발전기 회전속도인  $\Omega_g$ [rad/s]를 구한 후 풍속  $V$ [m/s]와 발전기 회전속도  $\Omega_g$ [rad/s]가 신경망 제어기의 입력으로 들어가고 신경망 제어기는 발전기의 정격 회전속도와 현재 회전속도의 차이를 계산하고 이에 맞는 피치각  $\beta$ [deg]의 증감인  $\Delta\beta$ [deg]를 계산하여 풍력발전기로 보낸다. 그리고 현재 풍속  $V$ [m/s]와 현재 피치각  $\beta$ [deg]에서 신경망 제어기로부터 받은  $\Delta\beta$ [deg]를 합한  $\beta$ [deg]를 이용하여 다시  $\Omega_g$ [rad/s]를 구한다. 이런 방식으로 계속 반복하여 발전기의 정격 회전속도와 현재 회전속도의 차이를 줄이도록 풍력발전기의 피치각을 제어한다.

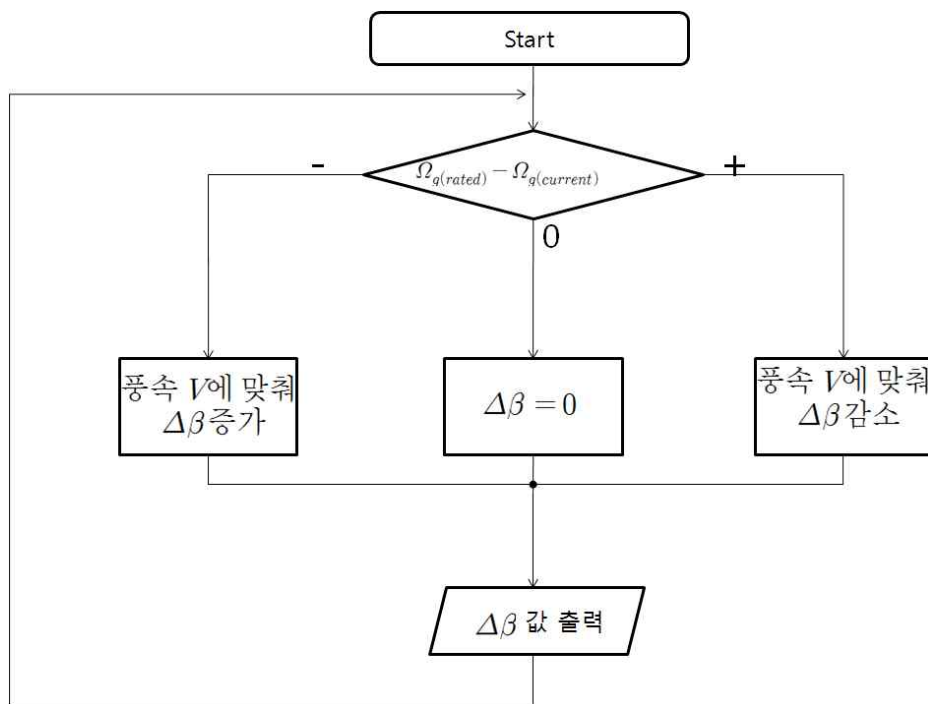


Fig. 9 Flow chart of pitch control consider wind and generator speed



풍속과 발전기의 회전속도를 고려한 피치제어 알고리즘을 Fig. 9에 나타내었으며 그 제어 전략은 다음과 같다.

I) 발전기의 정격 회전속도와 현재 회전속도의 차이인  $\Omega_{g(rated)} - \Omega_{g(current)}$ 를 입력으로 받는다.

II) 저풍속일 때는 비교적 피치각  $\beta$ [deg]의 변화량을 적게, 고풍속일 때는 피치각  $\beta$ [deg]의 변화량을 비교적 크도록 한다.

III) I)의 결과가 양수이면 피치각  $\beta$ [deg]를 감소시키고 음수이면 피치각  $\beta$ [deg]를 증가시키며, 0일 때는 다음 피치각  $\beta$ [deg]의 변화량을 0으로 한다.

IV) 다음 발전기 회전속도와 풍속을 받아 위 과정을 반복한다.

Fig. 9에서 알 수 있듯이 일단 발전기의 정격 회전속도와 현재 회전속도의 차이가 양수이면 현재 발전기의 현재 회전속도가 정격 회전속도보다 낮은 상태이므로 피치각  $\beta$ [deg]를 감소시켜 발전기 회전속도를 높도록 제어하고, 그 반대로 발전기의 정격 회전속도와 현재 회전속도의 차이가 음수이면 피치각  $\beta$ [deg]를 증가시켜 발전기 회전속도를 감소시키는 것이 이 알고리즘의 목적이다. 그리고 풍속이 적을 때는 피치각의 변화량을 적게, 고풍속일 때는 피치각의 변화량  $\Delta\beta$ [deg]를 비교적 크게 되도록 훈련 시켰다.

### 3.2.2 훈련요소

앞 절에서 제시한 알고리즘을 따라 신경망 제어기를 훈련시켜야 하는데 본 장에서 신경망 제어기의 훈련을 위하여 고려한 요소들은 다음과 같다. 첫째 전체 훈련 개수, 둘째 풍속의 범위, 셋째 발전기의 정격 회전속도와 현재 회전속도 차이의 범위 그리고 넷째 피치각의 변화량이다.

전체 훈련 개수는 너무 적으면 그만큼 훈련오차가 줄어들지만 훈련되지 않은 부분의 정확성이 줄어들게 되고 또 훈련 개수가 너무 많으면 훈련오차가 너무 커지기 때문에 적당한 훈련개수를 정해야 한다. 풍속의 범위는 정격풍속이

11.4[m/s]이므로 신경망 제어기의 입력풍속은 11[m/s]부터 21[m/s]사이에서 결정하였다. 실험을 통하여 다양한 범위를 적용해 보았고 이 정도의 범위가 적당함을 알게 되었다. 피치각의 변화정도는 -1과 1사이의 값을 가지도록 훈련시켰으며 훈련풍속 중에도 저풍속일 때에는 발전기의 정격회전속도와 현재 회전속도의 차이에 따라 비교적 피치각의 변화를 작게, 고풍속에서는 피치각의 변화를 비교적 크게 하도록 훈련시켰다.

Table. 2는 신경망 제어기의 훈련에서 두 개의 입력값과 하나의 목표값을 보여준다. 만약 Table. 1과 같이 풍속  $V$ [m/s]를 11[m/s]부터 21[m/s]사이에서 6개의 풍속으로 나누고 각 풍속 당 목표값을 9개로 잡는다면 전체 훈련개수는 54개가 된다. 이때의 목표값은 각각 -1, -1, -0.7, -0.1, 0, 0.1, 0.7, 1, 1 로 고정시키고 발전기의 정격 회전속도와 현재 회전속도의 차이는 각 풍속마다 차이를 두고 훈련시켰다.

Table. 1 Inputs and desired output of NNC consider wind and generator speed

$V$ [m/s]	$\Omega_{g(rated)}$ $-\Omega_{g(current)}$	목표값 $\Delta\beta$	$V$ [m/s]	$\Omega_{g(rated)}$ $-\Omega_{g(current)}$	목표값 $\Delta\beta$	$V$ [m/s]	$\Omega_{g(rated)}$ $-\Omega_{g(current)}$	목표값 $\Delta\beta$
11	-20	1	15	-13	1	19	-6	1
11	-15	1	15	-8	1	19	-5	1
11	-5	0.7	15	-3	0.7	19	-2	0.7
11	-2	0.1	15	-2	0.1	19	-1	0.1
11	0	0	15	0	0	19	0	0
11	2	-0.1	15	2	-0.1	19	1	-0.1
11	5	-0.7	15	3	-0.7	19	2	-0.7
11	15	-1	15	8	-1	19	5	-1
11	20	-1	15	13	-1	19	6	-1
13	-15	1	17	-7	1	21	-5	1
13	-10	1	17	-6	1	21	-4	1
13	-4	0.7	17	-3	0.7	21	-2	0.7
13	-2	0.1	17	-2	0.1	21	-1	0.1
13	0	0	17	0	0	21	0	0
13	2	-0.1	17	2	-0.1	21	1	-0.1
13	4	-0.7	17	3	-0.7	21	2	-0.7
13	10	-1	17	6	-1	21	4	-1
13	15	-1	17	7	-1	21	5	-1

### 3.3 발전기의 회전속도 변화율을 고려한 신경망 피치제어

#### 3.3.1 알고리즘

피치제어는 발전기의 정격 회전속도와 현재 회전속도의 차이를 비교하여 피치각의 증감을 결정한다. 그러나 발전기의 정격 회전속도와 현재 회전속도의 차이만을 고려한 신경망 피치제어 알고리즘에서 시스템이 진동하거나 다운되는 문제가 발생하였다. 그래서 이러한 문제점들을 개선하기 위하여 본 장에서는 추가적으로 발전기 회전속도의 변화율을 고려한 피치제어 알고리즘을 제안한다.

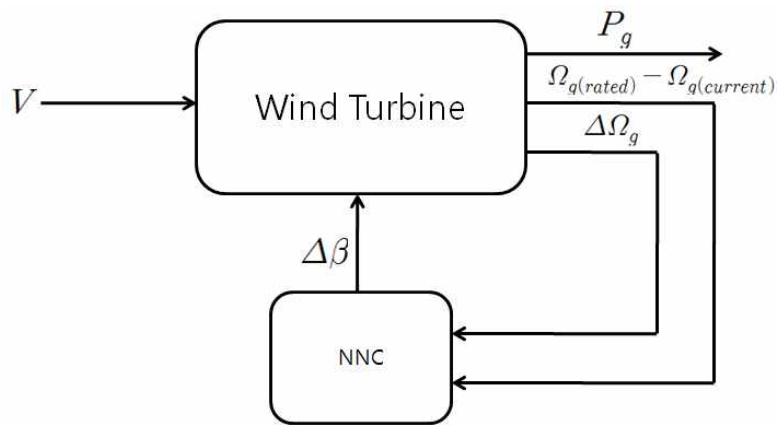


Fig. 10 Block diagram of wind turbine using NNC

Fig. 10은 위에서 설명한 두 개의 입력을 가진 신경회로망 제어기를 풍력 발전기에 적용시킨 것을 보여주는 블록 다이어그램이다. 2개의 입력을 제어요소로 사용하여 피치각의 변화율  $\Delta\beta[\text{deg}]$ 를 출력으로 보냄으로써 발전기의 회전속도를 정격으로 유지시킨다. 이 시스템의 동작 순서는 우선 풍속  $V[\text{m/s}]$ 가 풍력발전기로 들어가 발전기의 정격 회전속도와 현재 회전속도의 차이  $\Omega_{g(rated)} - \Omega_{g(current)}$ 와 발전기 회전속도의 변화율  $\Delta\Omega_g$ 을 구한다. 그리고 발전기의 정격 회전속도와 현재 회전속도의 차이  $\Omega_{g(rated)} - \Omega_{g(current)}$ 와 발전기 회전속도의 변화율  $\Delta\Omega_g$ 이 신경망 제어기의 입력으로 들어가고 신경망 제어기에서는 이에 맞는 피치각  $\beta[\text{deg}]$ 의 증감인  $\Delta\beta$ 를 계산하여 풍력발전기로 보낸다. 그 후에 현재 풍속  $V[\text{m/s}]$ 와 현재 피치각  $\beta[\text{deg}]$ 에서 신경망 제어기로부터 받은  $\Delta\beta[\text{deg}]$ 를 합한  $\beta[\text{deg}]$ 를 이용하여 다시  $\Omega_g[\text{rad/s}]$ 를 구한다. 이와 같은 방식으로 발전기의 정격 회전속도

와 현재 회전속도의 차이를 줄이도록 풍력발전기의 피치각  $\beta[\text{deg}]$ 를 제어한다.

회전속도 변화율을 고려한 피치제어알고리즘을 Fig. 11에 나타내었으며 이 제어 알고리즘의 전략은 다음과 같다.

I) 발전기의 정격 회전속도와 현재 회전속도의 차이인  $\Omega_{g(\text{rated})} - \Omega_{g(\text{current})}$ 를 입력으로 받는다.

II) I)의 결과가 양수이면 발전기 회전속도의 변화율  $\Delta\Omega_g$ 를 고려하여  $\Delta\Omega_g$ 가 양수이면 다음 피치각의 변화율  $\Delta\beta[\text{deg}]$ 를 0으로 하고 음수이면  $\Delta\beta[\text{deg}]$ 를 음수로 정한다.

III) I)의 결과가 음수이면 발전기 회전속도의 변화율  $\Delta\Omega_g$ 를 고려하여  $\Delta\Omega_g$ 가 음수이면 다음 피치각의 변화율  $\Delta\beta[\text{deg}]$ 를 0으로 하고 양수이면  $\Delta\beta[\text{deg}]$ 를 증가시킨다.

IV) I)의 결과가 0이면 발전기 회전속도의 변화율  $\Delta\Omega_g$ 에 부호에 맞게 다음 피치각의 변화율  $\Delta\beta[\text{deg}]$ 를 결정한다.

V) 다음 발전기 회전속도와 풍속을 받아 위 과정을 반복한다.

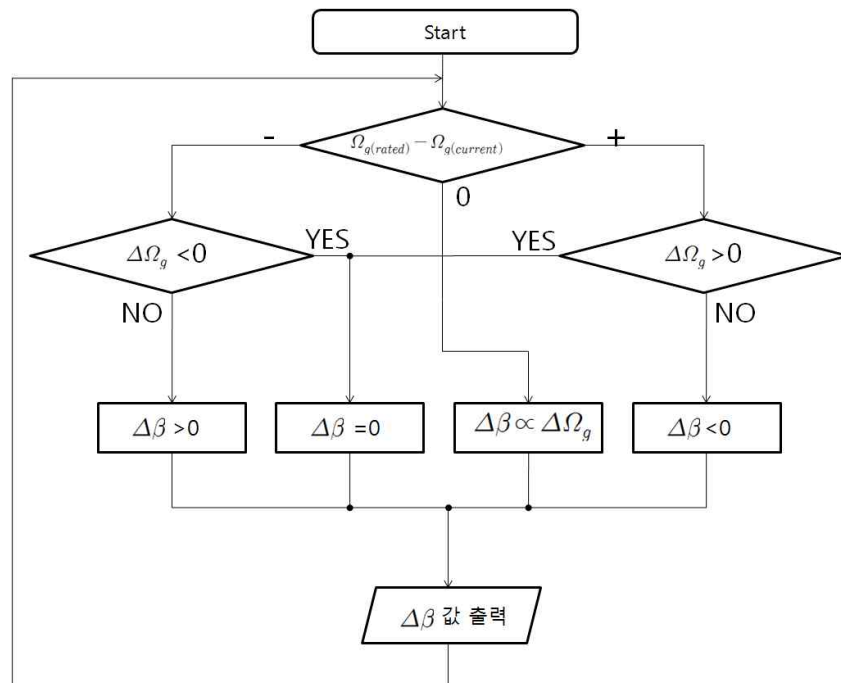


Fig. 11 Flow chart of pitch control consider changing rate of generator speed

위의 과정을 자세하게 설명하면 우선 발전기의 정격 회전속도와 현재 회전속도를 비교한다. 즉 발전기의 정격 회전속도가 현재 회전속도보다 클 때, 작을 때 또는 같을 때의 세 가지 경우로 구분한다. 그리고 각 경우에서 회전속도 변화율을 추가적으로 고려하여 피치각의 변화율을 결정한다. 먼저 발전기의 정격 회전속도와 현재 회전속도의 차이  $\Omega_{g(rated)} - \Omega_{g(current)}$ 가 양수이면 발전기의 현재 회전속도가 정격 회전속도보다 작기 때문에 피치각  $\beta[\text{deg}]$ 를 줄여서 발전기의 현재 회전속도를 높여줄 필요가 있다. 하지만 무조건 다음 피치각의 변화량  $\Delta\beta[\text{deg}]$ 를 음수로 놓지 않고 발전기 회전속도 변화율을 고려하여 발전기 회전속도가 현재 감소중이거나 0이면 피치각  $\beta[\text{deg}]$ 를 감소시키기 위해 다음 피치각의 변화량  $\Delta\beta[\text{deg}]$ 를 음수로 놓지만 발전기 회전속도가 증가하고 있다면 다음 피치각의 변화량  $\Delta\beta[\text{deg}]$ 를 0으로 하여 발전기 회전속도를 제어한다. 그 반대의 경우인 발전기의 정격 회전속도와 현재 회전속도의 차이  $\Omega_{g(rated)} - \Omega_{g(current)}$ 가 음수이면 발전기 회전속도 변화율을 고려하여 발전기 회전속도가 현재 증가중이거나 0이면 피치각  $\beta[\text{deg}]$ 를 증가시키기 위해 다음 피치각의 변화량  $\Delta\beta[\text{deg}]$ 를 양수로 놓고 발전기 회전속도가 감소하고 있다면 다음 피치각의 변화량  $\Delta\beta[\text{deg}]$ 를 0으로 하여 발전기 회전속도를 제어한다. 그리고 마지막으로 발전기의 정격회전속도와 현재 회전속도의 차이  $\Omega_{g(rated)} - \Omega_{g(current)}$ 가 0일 경우에는 발전기 회전속도 변화율에 크기와 부호에 비례하도록 다음 피치각의 변화량  $\Delta\beta[\text{deg}]$ 를 결정한다. 발전기의 정격 회전속도와 현재 회전속도의 차이에 따라 즉시 피치각  $\beta[\text{deg}]$ 를 증감하지 않고 발전기 회전속도 변화율을 고려하는 이유는 피치각  $\beta[\text{deg}]$ 가 변하는 즉시 발전기의 회전속도에 영향을 주는 것이 아니기 때문에 더욱 원활한 제어를 위하여 발전기 회전속도 변화율을 고려하여 피치제어 알고리즘을 고안하였다.

### 3.3.2 훈련요소

발전기 회전속도 변화율을 고려한 신경망 피치제어의 훈련을 위해 고려된 사항은 훈련 사이즈, 발전기의 정격 회전속도와 현재 회전속도의 차이의 범위, 발전기 회전속도의 변화율의 범위, 다음 피치각의 변화량  $\Delta\beta[\text{deg}]$ 이다.

Table. 3은 앞에서 제시한 알고리즘에 따라 신경망 제어기의 훈련을 위하여 두

개의 입력에 대한 피치각의 변화율  $\Delta\beta[\text{deg}]$ 을 설정한 것이다. 발전기의 정격 회전속도와 현재 회전속도의 차이와 발전기 회전속도의 변화율의 범위는 둘 다 -10, -5, 0, 5, 10으로 5개씩 잡아 훈련 사이즈는 25가 된다.

Table. 2 Inputs and desired output of NNC consider changing rate of generator speed

$\frac{\Omega_{g(\text{rated})}}{\Omega_{g(\text{current})}}$	$\Delta\Omega_g$	$\Delta\beta[\text{deg}]$	$\frac{\Omega_{g(\text{rated})}}{\Omega_{g(\text{current})}}$	$\Delta\Omega_g$	$\Delta\beta[\text{deg}]$
-10	-10	0	0	5	0.5
-10	-5	0	0	10	1
-10	0	0.5	5	-10	-1
-10	5	0.7	5	-5	-0.5
-10	10	1	5	0	-0.3
-5	-10	0	5	5	0
-5	-5	0	5	10	0
-5	0	0.3	10	-10	-1
-5	5	0.5	10	-5	-0.7
-5	10	1	10	0	-0.5
0	-10	-1	10	5	0
0	-5	-0.5	10	10	0
0	0	0			

## 제 4 장 시뮬레이션 및 결과

### 4.1 풍력발전기의 Matlab/Simulink 모델

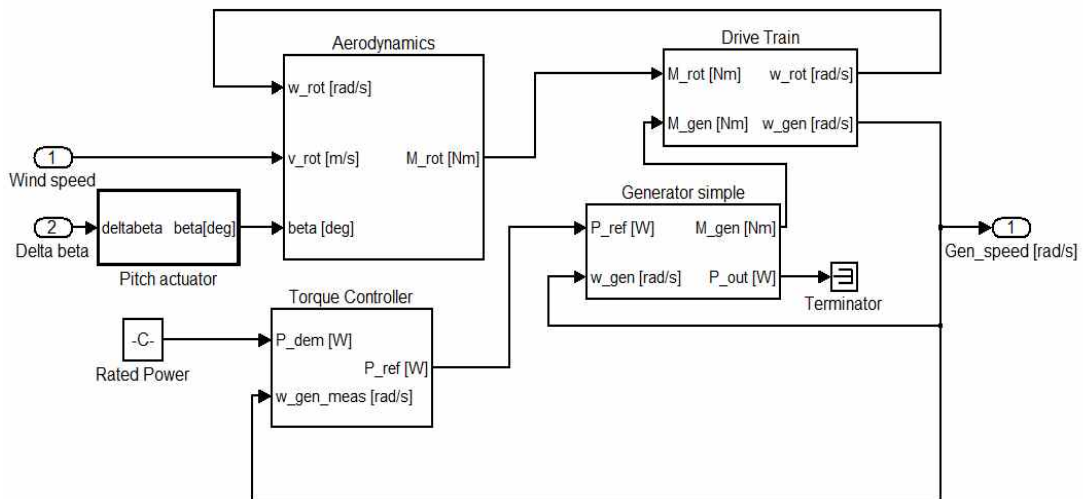


Fig. 12 Matlab/Simulink block diagram of NREL 5MW wind turbine

Fig. 12는 본 논문에서 사용한 NREL에서 제공하는 5MW의 풍력발전기의 Matlab/Simulink 모델이다.[11] 이 풍력발전기 모델의 제어방식은 풍력발전기로 들어오는 풍속이 정격풍속 이하일 때에는 Torque Controller블록에서 토크를 제어하여 바람이 갖는 에너지를 최대한 받아들이게 함으로써 효율을 높이고 정격 풍속 이상일 때에는 외부에 있는 신경망 제어기로부터 다음 피치각의 변화량을 받아 Pitch actuator를 통해 피치각을 계산하고 계산된 피치각을 이용하여 다시 출력과 발전기 회전속도를 구하여 외부에 있는 신경망 제어기의 입력을 보내줌으로써 발전기의 회전속도와 출력을 제어한다.

### 4.1.1 Aerodynamics model

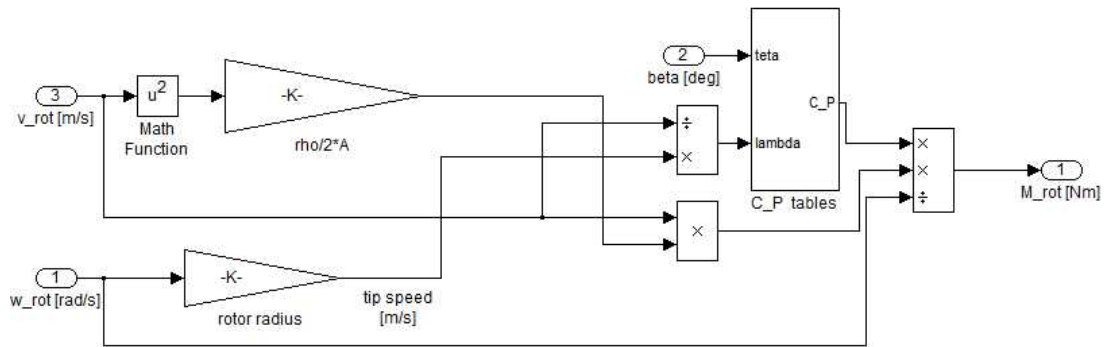


Fig. 13 Matlab/Simulink block diagram of Aerodynamics

Fig. 13은 Fig. 12의 Aerodynamics 블록 내부의 모습을 보여준다. 이 모델은 식 (2-6)을 Matlab/Simulink로 구현한 것이며 풍속과 로터의 회전속도, 피치각을 받아 로터의 토크(Torque)인  $M_{rot}[Nm]$ 를 계산한다.

### 4.1.2 Drive Train model

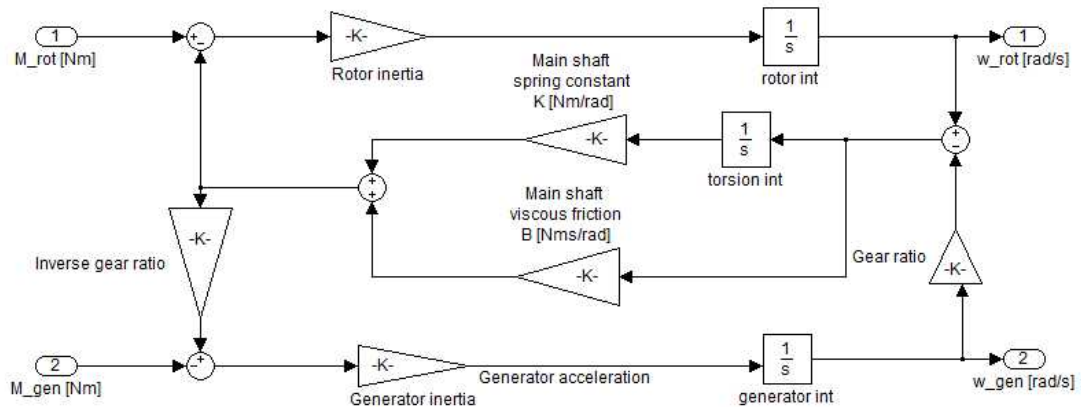


Fig. 14 Matlab/Simulink block diagram of Drive Train

Fig. 14는 Fig. 12의 Drive Train 블록의 내부의 모습을 보여준다. 이 모델은 식 (2-7)과 식 (2-8)을 Matlab/Simulink로 구현한 것이며 발전기 토크와 로터의 토크를 이용하여 발전기 회전속도와 로터의 회전속도를 계산한다.



### 4.1.3 Generator model

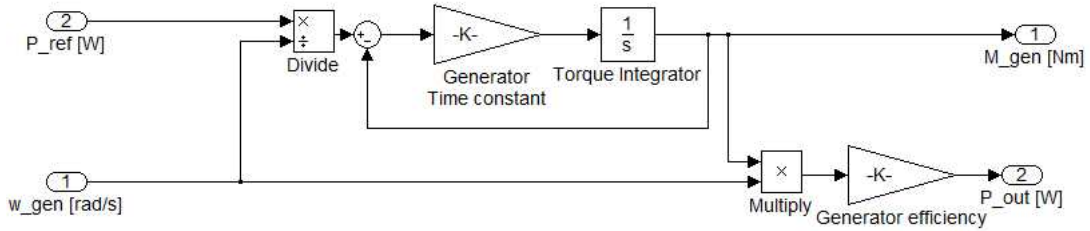


Fig. 15 Matlab/Simulink block diagram of Generator

Fig. 15는 Fig. 12의 Generator 블록의 내부의 모습을 보여준다. 앞에서 설명했듯이 발전기 모델은 간단하게 구성하였다.

### 4.1.4 Torque Controller model

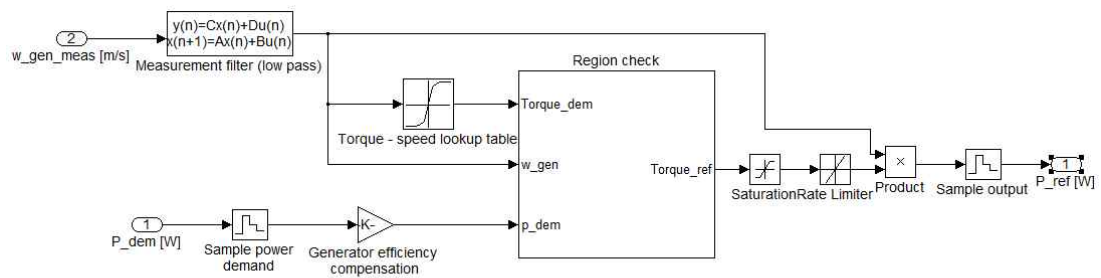


Fig. 16 Matlab/Simulink block diagram of Torque Controller

Fig. 16은 Fig. 12의 Torque Controller 블록의 내부의 모습을 보여준다. 이 블록은 정격풍속 이하일 때에는 Torque-speed lookup table을 통해 토크제어를 하여 최적의 출력을 낼 수 있도록 하고 정격풍속 이상에서는 정격전력을 현재 발전기 회전속도로 나눈 값을 보낸다.

#### 4.1.5 Pitch actuator

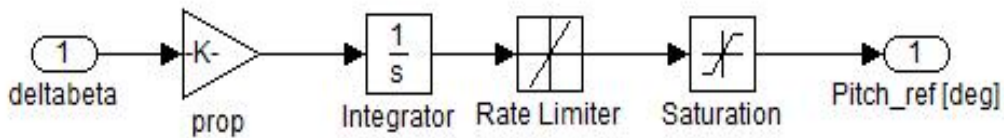


Fig. 17 Matlab/Simulink block diagram of Pitch actuator

Fig. 17은 Fig. 12의 Pitch actuator 블록의 내부의 모습을 보여준다. 훈련된 신경망 제어기의 출력은 피치각의 다음 변화량  $\Delta\beta[\text{deg}]$ 이다. 그러므로 실제 피치각이 적용되는 Aerodynamics 블록으로 들어가기 전에 이 블록에서 간단한 조절이 필요하다. 앞에서 훈련시키고자 한 피치각의 목표값은 비율을 고려하여 -1에서 1 사이의 값을 갖도록 훈련시켰다. 이렇게 훈련시킨 값들이 그대로 다음 피치각의 변화량으로 들어가게 된다면 실제 블레이드의 피치각의 1초당 최대변화량과 맞지 않기 때문에 피치각의 변화량인  $\Delta\beta[\text{deg}]$ 가 이 블록으로 들어오면 먼저 알맞은 비율을 곱해 주어야 한다. 본 논문에서 사용한 NREL 5MW 풍력발전기의 최대 피치각의 변화율은  $8[\text{deg}/\text{s}^2]$ 이므로 그에 맞춰 적당한 값을 주어야 한다. 그렇게 적당한 비율을 곱한 피치각의 변화량은 Integrator 블록으로 들어가 누적되게 되고 이 누적된 값들은 혹시 모를 크기의 어긋남을 방지하고 실제 피치각인  $0\sim 90$ 도 사이의 값을 벗어나지 않도록 Rate limiter와 Saturation 블록을 이용하여 다시 제한해 준다.

## 4.2 신경망 제어기의 훈련을 위한 Matlab/Simulink 모델

Fig. 11은 3.1절에서 나타낸 오류역전파 신경망의 학습 방법을 Matlab/Simulink를 통하여 구현한 것이다. 부수적으로 학습과정에서 실시간으로 전체오차가 변하는 모습을 보기 위하여 Scope를 달았다.

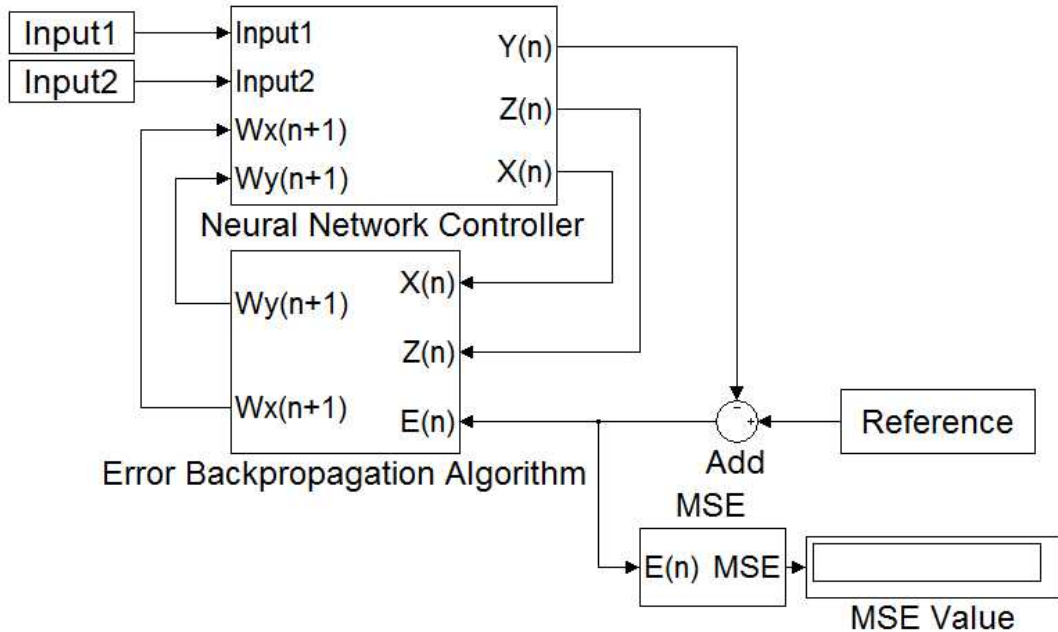


Fig. 18 Block diagram of Matlab/Simulink for Error back-propagation algorithm

본 논문에서 다루는 피치제어 신경망은 모두 두 개의 입력과 하나의 출력으로 되어 있으며 두 가지 모두 같은 훈련 방법을 쓰고 있으므로 Fig. 18에서 나타낸 Matlab/Simulink 모델을 이용하여 훈련시킨다. Fig. 18에서  $W_x$ 는 입력층과 은닉층 사이의 가중치, 즉 식 (3-1)의  $w_{ip}$ 를 나타내고  $W_y$ 는 은닉층과 출력층의 가중치, 즉 식 (3-2)의  $w_i$ 를 나타낸다. 두 개의 입력에 훈련사이즈인  $l$ 이 신경망으로 들어가면 신경망 내부에서는 두 개의 입력에  $-1$ 의 값을 갖는 바이어스가 더해지고 이 값들이  $3 \times l$ 인 행렬이 되어  $X(n)$ 이 된다. 그리고 뉴런이 5개인 은닉층의 변환함수의 출력값들에 바이어스값을 더한  $6 \times l$ 의 행렬이  $Z(n)$ 이 되고, 신경망의 출력인  $Y(n)$ 은  $1 \times l$ 인 행렬이 되어 이 값과 목표값의 차이인  $1 \times l$ 의 행렬이  $E(n)$ 이 된다. 이 세 개의 값들이 전체적인 오차인  $E(n)$ 을 줄이도록 다음 가중치 값들

을 계산하는 수식에 사용된다.

Fig. 19는 Fig. 18의 훈련 모델에서 전체적인 오차가 줄어들도록 다음 가중치를 구하여 계산하는 부분인 Error Back-propagation Algorithm의 내부 구조를 나타낸 그림이다.

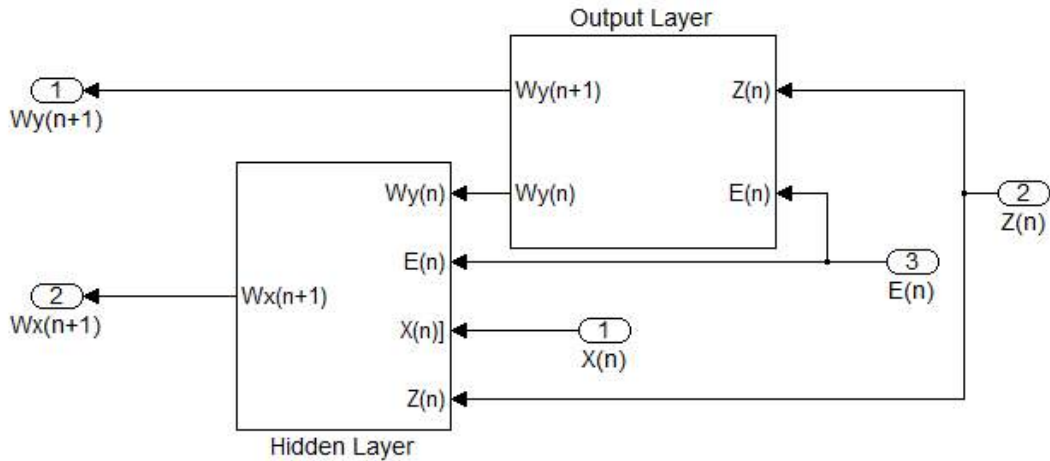


Fig. 19 Error back-propagation algorithm

먼저  $W_x(n+1)$ 을 구하는 과정은 식 (3-1)과 같고, Matlab/Simulink에서 그 식을 구현하고 계산한 과정은 다음과 같다.

I)  $1 \times 6$ 의 행렬인  $W_y(n)$ 을 전치시킨 후  $1 \times l$ 의 행렬인  $E(n)$ 을 곱하여  $6 \times l$ 의 행렬을 구한다.

II)  $6 \times l$ 의 행렬인  $Z(n)$ 을 은닉층의 변환함수인 식 (3-5)의 미분식에 넣어 그 값을 구한 후 I)의 결과와 원소 대 원소 곱셈을 한다.

III) 위의 결과로 나온  $6 \times l$  행렬은 출력층의 바이어스값까지 포함한 값이므로 은닉층에서의 가중치에 출력층의 바이어스값은 포함되지 않는다. 그러므로 바이어스값인 한 행을 제외한  $5 \times l$ 행렬을 selector를 이용하여 구해준다.

IV)  $3 \times l$ 의 행렬인  $X(n)$ 을 전치시켜 III)의 결과와 곱한 후  $\frac{\alpha}{l}$ 를 곱한다.

V) 이전의  $W_x(n)$ 값을 다시 더해 식 (3-1)과 같은 결과가 나올 수 있도록 해준다.

다음으로  $Wy(n+1)$ 을 구하는 과정은 식 (3-2)과 같고, Matlab/Simulink에서 그 식을 구현하고 계산한 과정은 다음과 같다.

I) 원래  $1 \times l$ 의 행렬인  $Y(n)$ 을 출력층의 변환함수의 미분식에 넣어 그 값을 구한다. 그러나 본 논문에서의 출력층의 변환함수는 기울기가 1인 선형함수를 사용하였으므로 그 미분식은 1이기 때문에 식 (3-2)의  $f_2'(net)$ 은 1이 된다.

II) I)의 결과와  $1 \times l$ 의 행렬인  $E(n)$ 의 값을 곱하고 그 값에  $Z(n)$ 의 전치행렬을 곱한 후, 그 결과에  $\frac{\alpha}{l}$ 을 곱한다.

III) 이전의  $Wy(n)$ 값을 다시 더해 식 (3-2)와 같은 결과가 나올 수 있도록 해준다.

이와 같은 방법으로 가중치의 값들을 전체적인 에러인 식 (3-4)가 줄어들도록 가중치들을 수정해 나간다.

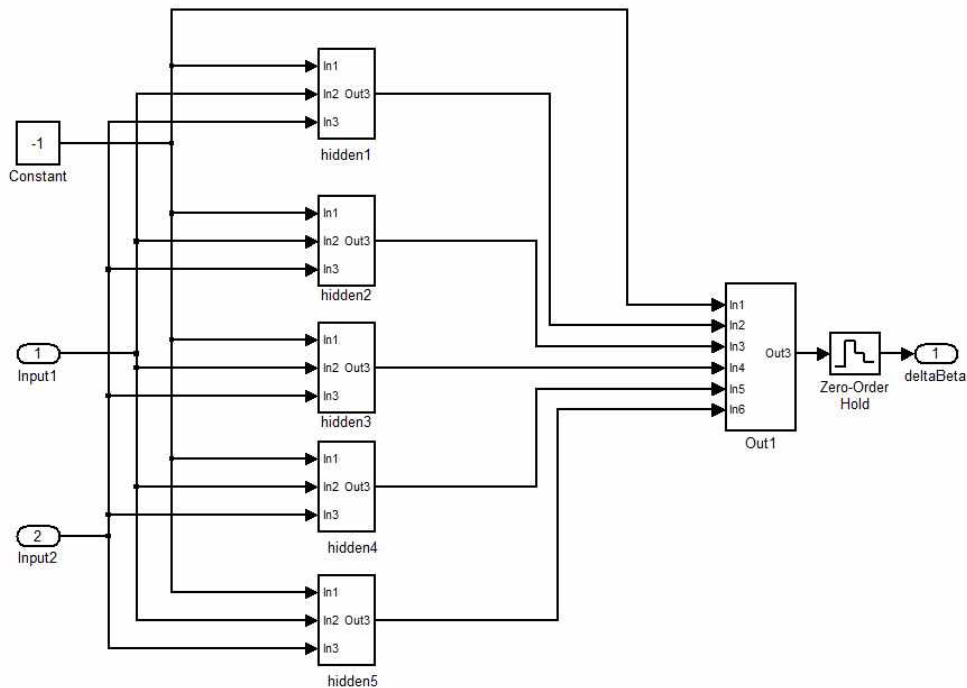


Fig. 20 Block diagram of Matlab/Simulink for Neural Network Controller

Fig. 20은 Matlab/Simulink를 이용하여 구현한 신경망 제어기의 모델을 보여준다. 이것은 Fig. 18에서 보이는 Neural Network Controller 블록과는 다른 것으로 훈련이 아닌 풍력발전기 모델과 연동시켜 사용하는 모델이다. 2개의 입력과 하나의 바이어스가 입력층을 이루고 있고 한 개의 은닉층에 다섯 개의 뉴런이 있으며 출력층에서는 다음 피치각의 변화량이 출력으로 나가게 된다. 은닉층에 있는 5개의 뉴런 각각의 내부에는 들어온 세 개의 입력에 가중치를 곱해 더한 후 식 (3-5)의 변환함수를 거쳐 출력되며 출력층에 있는 1개의 뉴런 또한 들어온 6개의 입력에 가중치를 곱해 더한 후 변환함수를 거쳐 출력되는데 여기서 변환함수는 식 (3-6)과 같이 입력이 그대로 나가는 함수이므로 6개의 입력에 가중치를 곱해 그 값들은 더한 값이 그대로 나가게 된다.

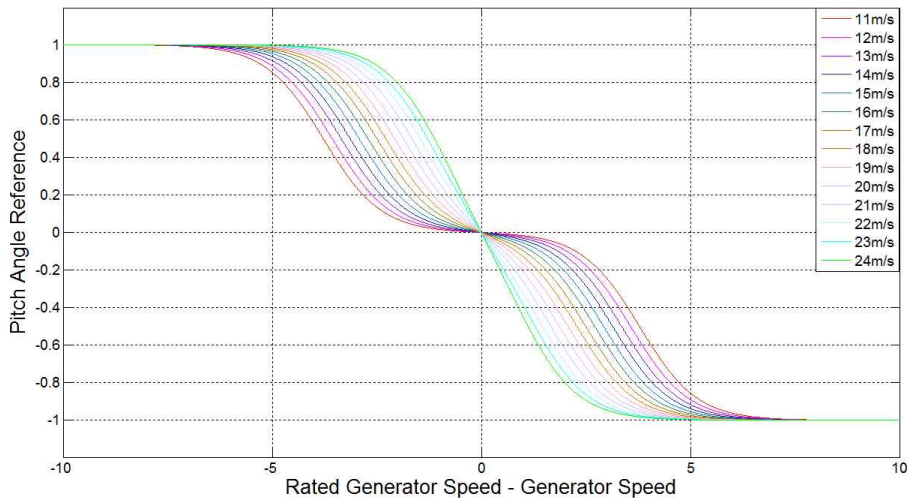


Fig. 21 Test outputs of NNC consider wind and generator speed for untrained region

3.2.2절의 Table. 1에서 설계한 신경망의 입, 출력의 값들을 Fig. 18의 Matlab/Simulink 훈련모델에 넣고 훈련시킨 후 그 가중치들을 Fig. 20에 적용시켜 충분히 훈련된 뉴런 연결 가중치들이 Table. 2에 포함되지 않는 풍속에서도 제대로 작동하는지를 알기 위하여 학습된 가중치를 신경망 제어기에 넣고 테스트하였다. 풍속은 11[m/s]부터 24[m/s]까지 1[m/s]씩 증가하는 형태로 넣고 발전기 정격 회전속도와 현재 회전속도의 차이는 -10에서 10까지 변하는 값을 입력으로 넣어서 실험하였다. Fig. 21의 결과를 보면 발전기의 정격 회전속도와 현재 회전속도의 차이에가 양수이면 피치각을 감소시키고 음수이면 피치각을 증가시

키는 것을 볼 수 있다. 그리고 풍속에 따라 저풍속 구간에서는 피치각을 비교적 적게 증감하고 고풍속 구간에서는 피치각을 크게 증감시키는 것을 볼 수 있다. 또한 이 테스트로부터 Table. 1에 포함되지 않은 풍속에서도 제대로 된 결과가 나타나는 것을 볼 수 있다.

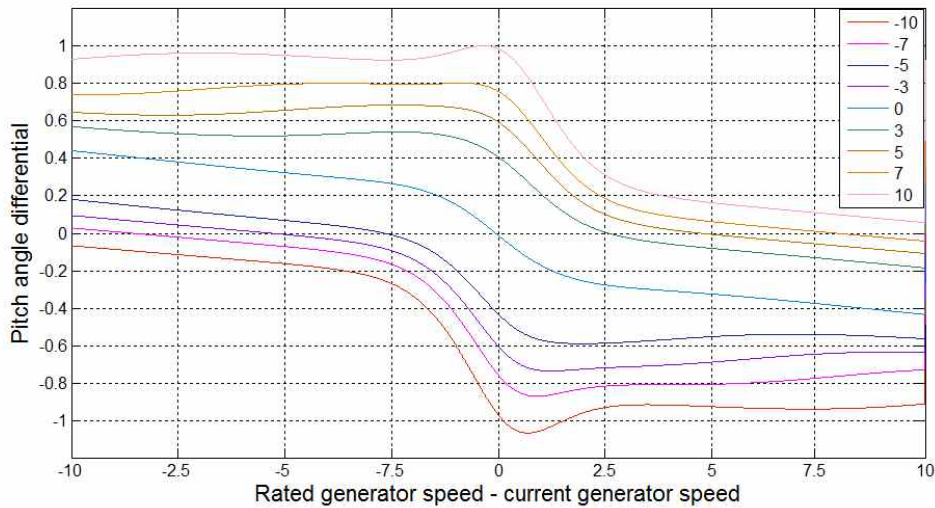


Fig. 22 Test outputs of NNC consider changing rate of generator speed for untrained region

같은 방법으로 Table. 2에서 설계한 신경망의 입, 출력의 값들을 Fig. 18의 Matlab/Simulink 모델에 넣고 훈련시켜 훈련된 가중치들을 Fig. 20의 신경망 제어기에 넣고 테스트 한 결과를 Fig. 22에 나타내었다. 테스트의 결과에는 Table. 2에 포함되지 않는 발전기 회전속도의 변화율과 발전기의 정격 회전속도와 현재 회전속도의 차이가 입력으로 들어갔을 때의 결과도 나타내었다. 그리고 앞 절의 훈련결과에서는 발전기의 회전속도를 기준으로 보면 발전기의 정격 회전속도가 현재 회전속도보다 높았을 경우에는 피치각을 무조건 감소시켜 현재 회전속도를 늘리고 발전기의 정격 회전속도가 현재 회전속도보다 낮을 경우에는 피치각을 무조건 증가시켜 현재 회전속도를 억제하려는 목적이 있었다. 그러나 발전기 회전속도 변화율을 고려한 신경망 제어기에서는 발전기의 회전속도에 발전기 회전속도 변화율까지 고려함으로써 발전기의 정격 회전속도보다 현재 회전속도가 아무리 높더라도 피치각을 무조건 증가시키지 않고 발전기의 회전속도 변화량이

지금 감소중이라면 피치각의 다음 변화량을 0으로 정하기도 하였다. 그리고 Fig. 21의 결과를 보면 발전기의 정격 회전속도와 현재 회전속도의 차이가 0인 부근에서 발전기 회전속도의 변화량이 클 때 조금 불안한 모습을 보이지만 학습에 포함되지 않은 -7, -3, 3, 7의 값에서도 학습에 포함된 값에서처럼 제대로 작동하는 것을 확인 할 수 있다.

### 4.3 신경망 피치제어시스템의 Matlab/Simulink 모델

#### 4.3.1 풍속과 발전기의 회전속도를 고려한 신경망 피치제어시스템

풍속과 발전기의 회전속도를 고려한 신경망 제어기를 풍력발전기에 적용한 Matlab/Simulink 시스템을 Fig. 23에 나타내었다.

이 시스템은 풍속이 풍력발전기 모델로 들어가 현재 발전기 회전속도인  $\Omega_g$  [rad/s]를 구한 후 풍속과 발전기 회전속도가 신경망 제어기의 입력으로 들어가고 신경망 제어기는 발전기의 정격회전속도와 현재 회전속도의 차이를 계산하고 이에 맞는 피치각  $\beta$ [deg]의 증감인  $\Delta\beta$ [deg]를 계산하여 풍력발전기로 보낸다. 그리고 현재 풍속과 현재 피치각에서 신경망 제어기로부터 받은  $\Delta\beta$ [deg]를 합한  $\beta$ [deg]를 이용하여 다시  $\Omega_g$ [rad/s]를 구한다.

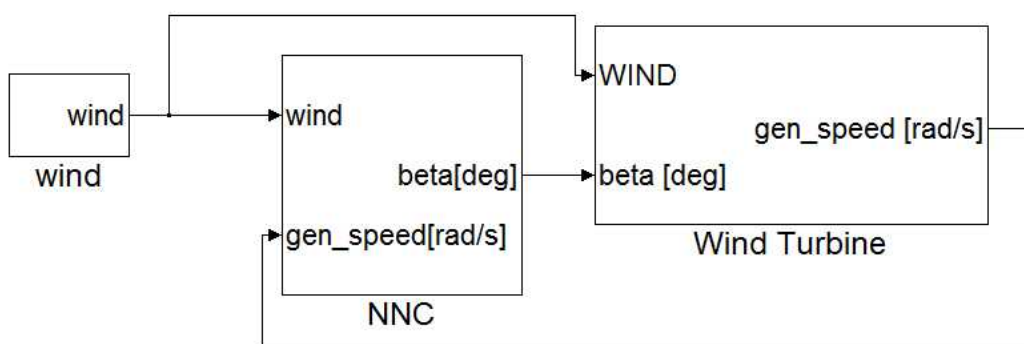


Fig. 23 Block Diagram of Matlab/Simulink for wind turbine system with NNC consider wind and generator speed.



Fig. 24는 풍속을 일정하게 증가시키는 계단함수형식의 풍속을 입력으로 했을 때의 풍력발전기의 출력, 현재 발전기 회전속도, 피치각의 변화를 보여준다. 풍속은 계단함수 형식으로 12부터 1씩 25까지 증가하는 형식을 입력으로 하여 발전량과 현재 발전기의 회전속도, 그리고 피치각의 변화를 나타내었다. 이 제어 시스템은 발전기의 정격 회전속도와 현재 회전속도의 차이에 따라 피치각의 증감을 이용하여 풍력발전기의 발전기 회전속도가 정격풍속 이상에서도 일정하게 유지되도록 제어하는 것을 목표로 한다. Fig. 24의 결과를 보면 피치각의 제어로 인하여 발전기 회전속도는 대체로 122.9[rad/s]로 제어되는 것을 볼 수 있다. 하지만 발전기 회전속도와 피치각의 변화의 처음 부분을 잘 살펴보면 처음의 발전기 회전속도는 정격을 넘어 크게 증가하는 것을 볼 수 있다. 그때의 피치각은 계속 증가하고 있으며 발전기 회전속도가 떨어지는 동안에도 증가하고 있다가 발전기 회전속도가 정격 이하까지 떨어질 때에야 피치각이 감소하기 시작한다. 발전기 회전속도가 정격 이하로 떨어지고 다시 올라갈 때에도 피치각은 같은 모습으로 계속 진동한다. 이와 같은 현상이 발생한 이유는 피치각이 변화하는 즉시 발전기 회전속도에 영향을 미치는 것이 아니므로 계속해서 이러한 진동이 일어나는 것으로 볼 수 있다. 고풍속 영역에서는 비교적 저풍속 보다는 그러한 현상이 영향을 적게 받아 피치각의 변화가 안정된 모습을 볼 수 있고 발전기 회전속도 또한 122.9[rad/s]로 잘 제어되는 것을 볼 수 있지만 발전기 회전속도가 조금씩 진동하는 모습에서 위와 같은 현상을 발견 할 수 있다.

Fig. 25는 고풍속에서 불규칙적으로 바뀌는 풍속을 입력으로 했을 때의 풍력발전기의 출력, 현재 발전기 회전속도, 피치각의 변화를 보여준다. 처음 부분에서 Fig. 24와 비교하였을 때 발전기 회전속도가 크게 튀었다가 진동하는 것을 볼 수 있는데, 이때의 이유는 앞의 풍속을 계단함수 형식으로 넣었을 때의 결과와 동일하게 피치각의 변화가 발전기 회전속도에 바로 영향을 주지 않으면서 생겨난 결과이다. 그리고 풍속의 변화가 급격할수록 발전기 회전속도의 오버슈트가 커지며 과도상태의 시간이 길어지긴 하지만 곧 정격 발전기 회전속도인 122.9[rad/s]로 제어되는 것을 볼 수 있다.

이 두 가지 방법으로 풍속을 변화시키며 살펴본 결과에서 풍속과 발전기 회전속도를 이용한 방법은 어느 정도 제어가 되긴 하지만 피치각의 변화가 발전기 회전속도에 바로 영향을 주지 않는다는 점이 문제가 된다는 것을 볼 수 있다.

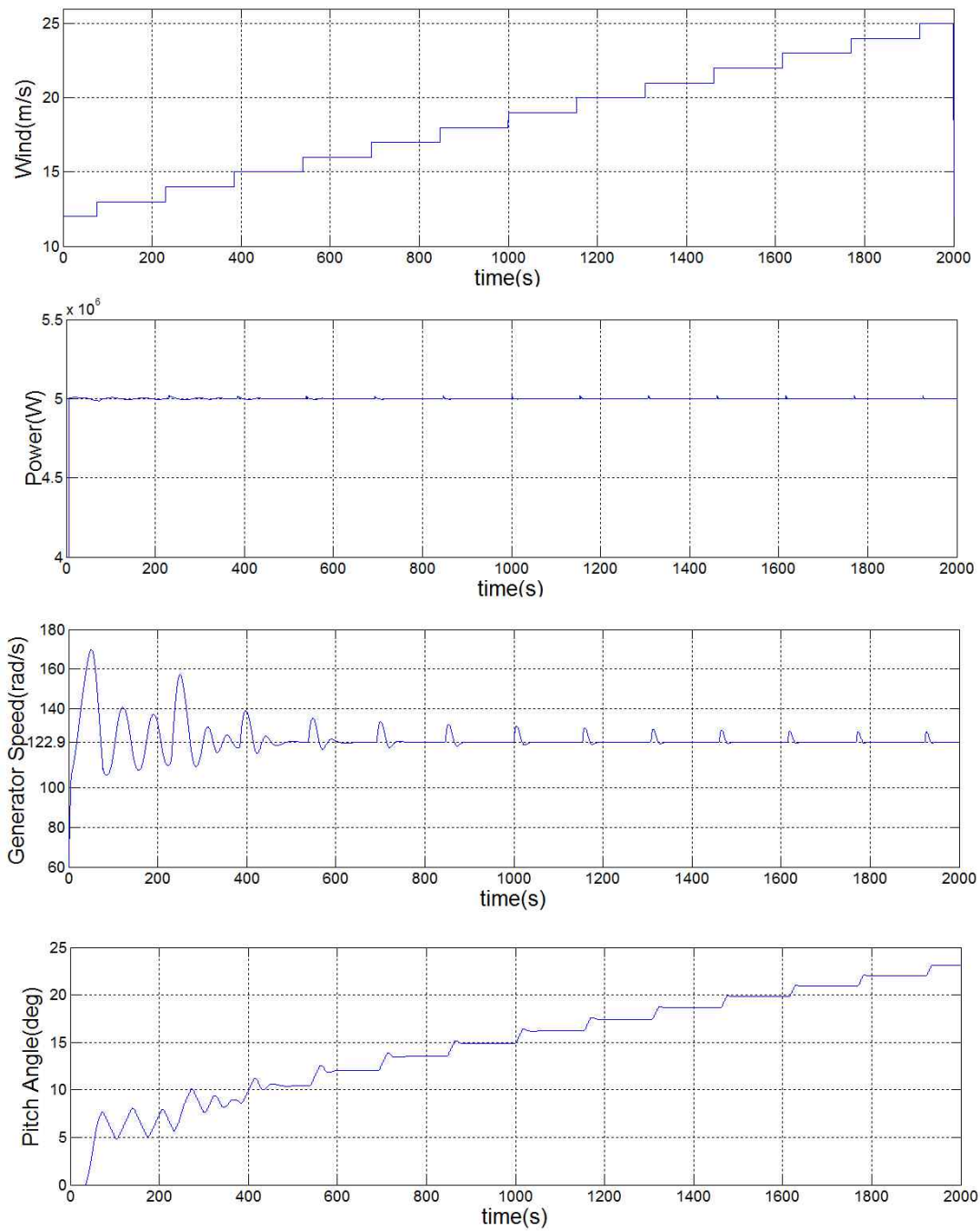


Fig. 24 Simulation results of wind turbine system with NNC consider wind and generator speed (step wind speed)

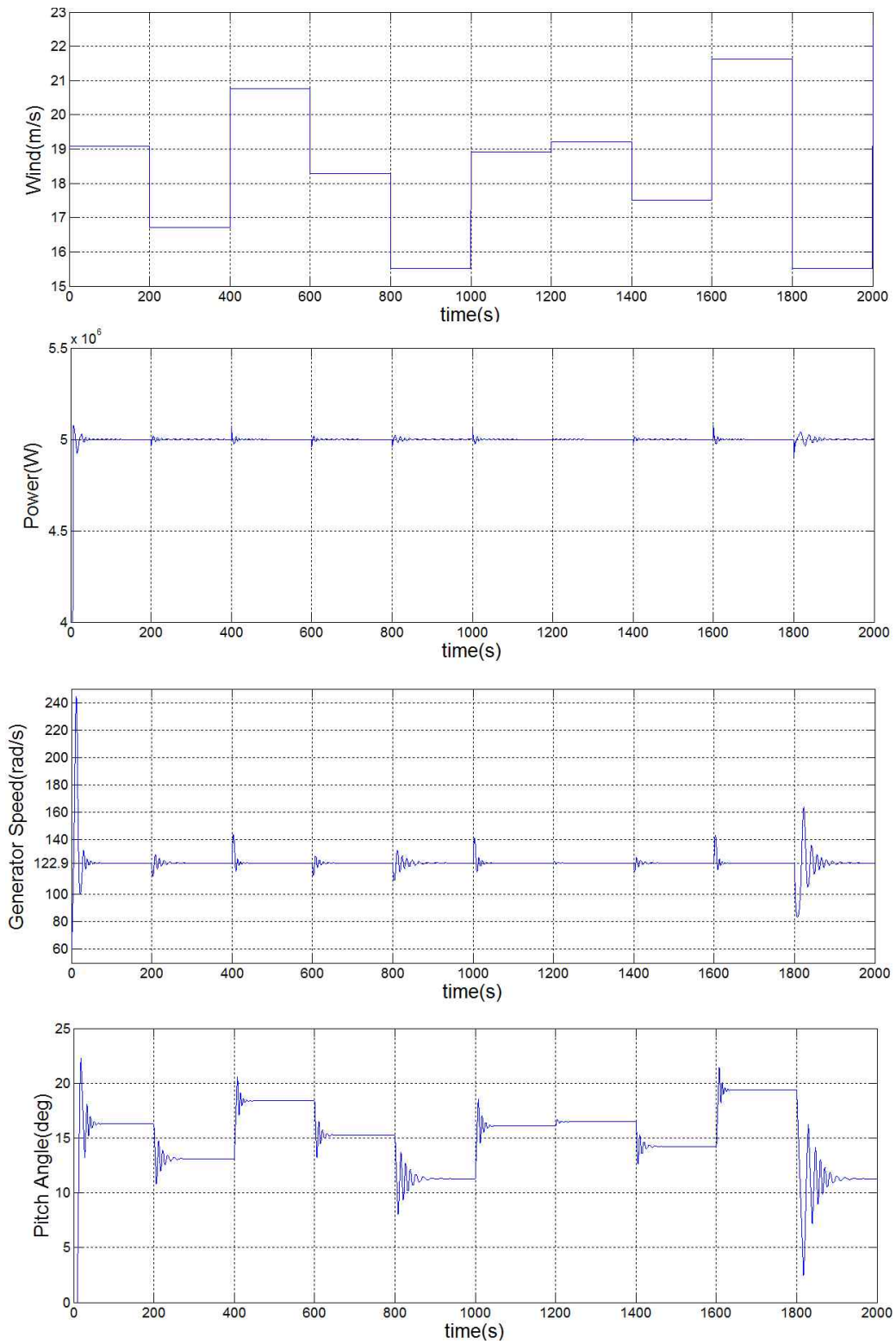


Fig. 25 Simulation results of wind turbine system with NNC consider wind and generator speed (random wind speed)

### 4.3.2 발전기의 회전속도 변화율을 고려한 신경망 피치제어시스템

발전기의 회전속도 변화율을 고려한 신경망 제어를 풍력발전기에 적용한 Matlab/Simulink 시스템을 Fig. 26에 나타내었다.

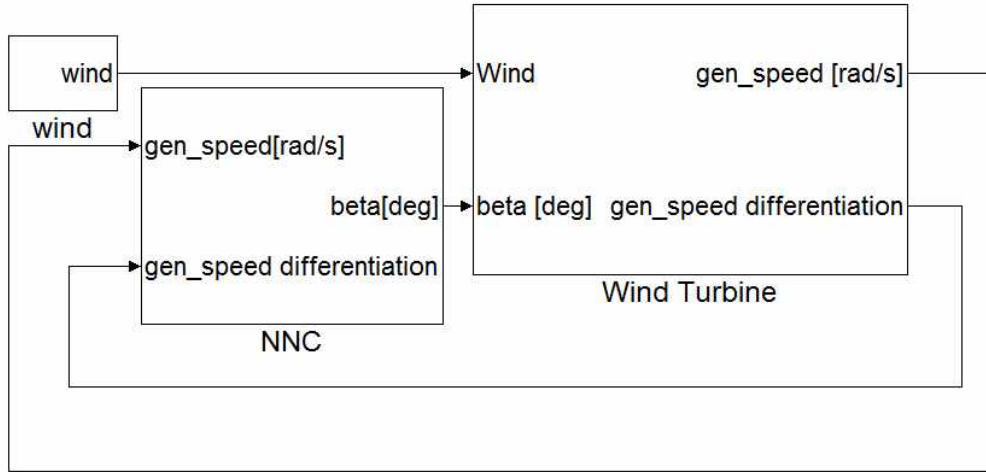


Fig. 26 Block Diagram of Matlab/Simulink for wind turbine system with NNC consider changing rate of generator speed.

Fig. 26의 풍력 발전기 모델에서 바람이 들어가면 풍력발전기모델에서 회전속도인  $\Omega_g$ [rad/s]를 구한 후 신경망 제어기의 입력인 회전속도의 변화율을 구하고 이를 신경망 제어기로 보낸다. 신경망 제어기에서는 발전기의 정격 회전속도와 현재 회전속도의 차이를 계산하고 두 개의 입력에 맞는 피치각  $\beta$ [deg]의 증감인  $\Delta\beta$ [deg]를 계산하여 풍력발전기로 보낸다. 그리고 현재 풍속과 현재 피치각에서 신경망 제어기로부터 받은  $\Delta\beta$ [deg]를 합한  $\beta$ [deg]를 이용하여 다시  $\Omega_g$ [rad/s]를 구한다.

Fig. 27은 풍속을 일정하게 증가시키는 계단함수형식의 풍속을 입력으로 했을 때의 풍력발전기의 출력, 현재 발전기 회전속도, 피치각의 변화를 보여준다. 저풍속 구간에서는 피치각의 빠른 진동이 있지만 앞의 풍속과 발전기 회전속도를 이용한 피치제어기의 시스템에서 발견된 문제점은 발견되지 않았다. 풍속과 발전기 회전속도를 이용한 피치제어기에서는 발전기 회전속도가 정격보다 높으면 무조건 피치각을 증가시키고 발전기 회전속도가 정격보다 낮으면 무조건 피치각을 감소시켰지만 발

전기 회전속도 변화율을 고려한 신경망 제어기는 여기에 발전기 회전속도의 변화율을 함께 고려함으로써 피치각이 변화가 발전기 회전속도에 바로 영향을 주지 않는 문제를 해결 할 수 있었다. 이로 인하여 발전기 회전속도는 크게 진동하지 않고 122.9[rad/s]를 잘 유지하고 있으며 출력 또한 눈에 띄게 향상된 출력을 볼 수 있다.

Fig. 28은 고풍속에서 불규칙적으로 바뀌는 풍속을 입력으로 했을 때의 풍력발전기의 출력, 현재 발전기 회전속도, 피치각의 변화를 보여준다. 시뮬레이션 결과를 통해서 알 수 있듯이 풍속이 급격하게 줄어들거나 크게 불 때의 진동이 보이긴 하지만 앞의 풍속과 발전기 회전속도를 고려한 피치제어기보다 발전기 회전속도와 피치각의 변화를 비교했을 때 과도상태의 시간이 길지 않고 풍력발전기의 출력 또한 눈에 띄게 향상된 성능을 보이고 있다.

이 두 가지 방법으로 풍속을 변화시키며 살펴본 결과, 입력이 계단함수 형식으로 들어간 부분에서의 저풍속 영역에 진동하는 부분과 입력이 불규칙적인 형식으로 들어간 부분에서의 풍속이 급격하게 떨어질 때 피치각이 크게 흔들린 부분을 제외하고는 발전기의 출력과 발전기 회전속도의 정격을 유지하는 면에서 앞의 풍속과 발전기 회전속도를 고려한 피치제어기의 시스템보다 좋은 성능을 보이고 있음을 볼 수 있다.

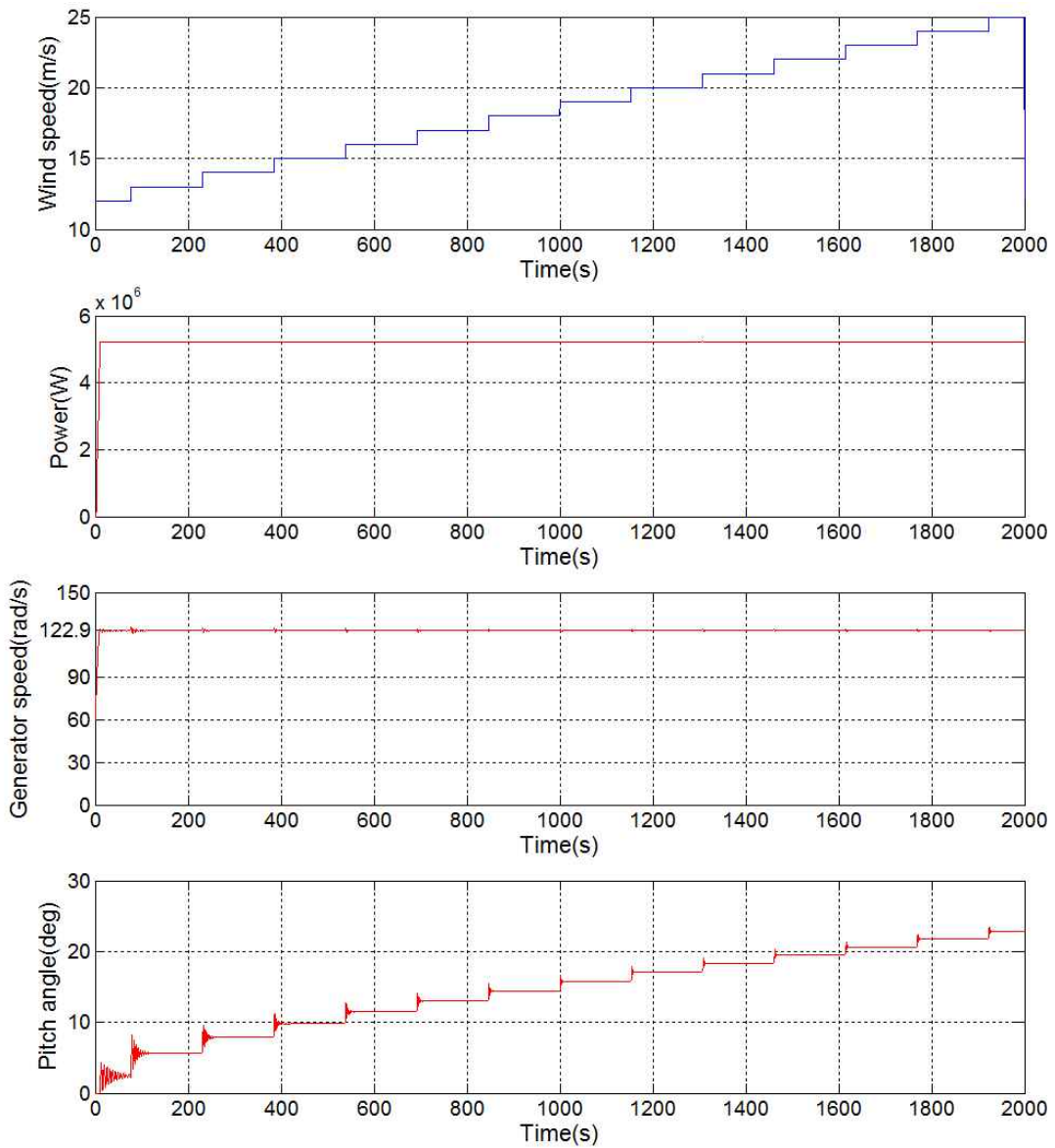


Fig. 27 Simulation results of wind turbine system with NNC changing rate of generator speed (step wind speed)

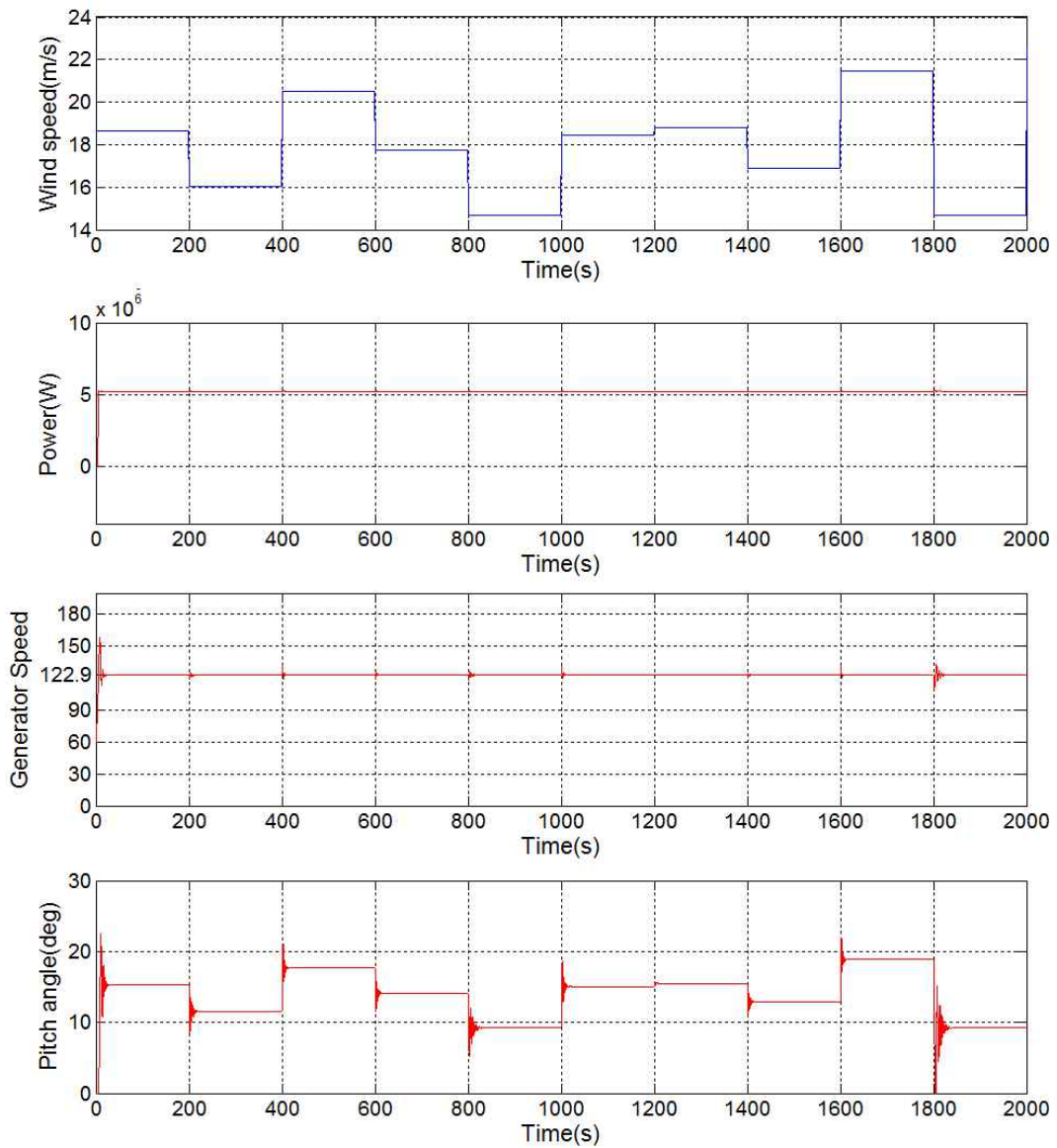


Fig. 28 Simulation results of wind turbine system with NNC changing rate of generator speed (random wind speed)

## 제 5 장 결 론

풍력발전기는 정격풍속에서 그 발전기의 용량에 맞는 발전량을 생산해 낸다. 하지만 그 이상으로 풍속이 강하게 불 경우에 발전기의 용량을 초과하는 전력을 생산해 낸다면 발전기의 과부하가 걸리게 될 뿐만 아니라 로터의 회전속도 또한 너무 증가하게 되어 안전적으로도 문제가 발생 할 수 있다. 이를 방지하기 위하여 블레이드(blade)의 회전속도를 제어할 필요가 있으며 가장 일반적으로 사용되는 방법은 피치각을 이용해 로터의 회전속도를 일정하게 유지시키고 발전량 또한 정격을 유지하도록 제어하는 것이다.

본 논문에서는 신경망 제어를 이용하여 피치각을 제어하는 방법에 대하여 연구하고 그 결과를 알아보았다. 풍력발전기의 모델로는 NREL에서 제공하는 5MW의 풍력발전기를 연구에 맞게 변형하여 사용하였다. 본 논문에서 가장 주의 깊게 봐야 할 부분은 목표값의 설정이다. 이는 적절한 값을 임의대로 설정하여 실험하고 그 중 가장 최선을 선택한 것으로 이를 어떻게 설정하느냐에 따라 결과가 천차만별로 변할 수 있다.

우선 풍속과 발전기 회전속도를 이용한 방법에서 발전기의 정격 회전속도와 현재 회전속도의 차이가 양수일 때는 무조건 피치각을 감소시키고 발전기의 정격 회전속도와 현재 회전속도의 차이가 음수일 때는 피치각을 증가시키는 신경망 피치제어를 설계하여 그 결과를 알아보았는데, 피치각을 제어하여 발전기 회전속도와 출력은 정격을 유지하지만 제어가 원활하게 된다고 보기에는 문제가 있어 보였다. 풍속이 계단함수 형식으로 들어갔을 때에서 자세하게 확인 할 수 있었는데, 풍력발전시스템에서 피치각의 변화가 즉시 발전기 회전속도에 영향을 주는 것이 아니기 때문에 피치각이 제대로 수렴하지 않고 진동하는 것을 볼 수 있다. 예를 들면 피치각이 필요이상으로 증가하게 되어 발전기 회전속도는 갑작스레 줄어들고 피치각이 필요이상으로 감소하게 되어 발전기 회전속도가 갑작스레 증가하며 결국 제대로 된 제어를 할 수가 없게 된다.

이를 보완하기 위하여 발전기의 정격 회전속도와 현재 회전속도의 차이를 하나의 입력으로 하고 발전기 회전속도의 변화율을 첨가한 신경망 피치제어를



설계하였다. 발전기 회전속도의 변화율을 첨가함으로써 발전기의 정격 회전속도와 현재 회전속도의 차이가 양수일 때에도 무조건 피치각을 감소시키지 않고 발전기 회전속도의 변화율까지 고려하여 다음 피치각의 변화율을 결정하는 신경망 피치제어기를 설계하여 그 결과를 살펴보니 피치각의 변화가 발전기 회전속도에 바로 영향을 주지 않아 생겼던 문제를 해결 할 수 있었다. 이로 인하여 발전기 회전속도는 크게 진동하지 않고 122.9[rad/s]를 잘 유지하고 있으며 발전기의 출력 또한 눈에 띄게 향상된 것을 볼 수 있다. 하지만 계단함수형식의 풍속이 입력으로 들어갔을 때의 빠른 진동을 하는 부분과 불규칙적인 형식의 풍속이 입력으로 들어갔을 때 풍속이 급격하게 떨어질 때 피치각이 순간적으로 0까지 떨어진 부분은 향후 연구가 더 필요한 부분이라 생각된다.

## 참고문헌

- [1] International Energy Agency, World Energy Outlook, 2012. OECD/IEA, 2012.
- [2] C-W Lim. "A Study on Properties of Pitch Control for Wind Turbine." Journal of Korean Society of Mechanical Engineers 107-113, 2010.
- [3] K.-N. Ko and J.-C. Hur, An Introduction to Wind Engineering, Munundang, 2007.
- [4] T. Burton, D. Sharpe, N. Jenkins, and E. Bossanyi, Wind Energy Handbook, Wiley, 2001.
- [5] J.-M. Ko, S.-Y. Yang, C.-J. Boo, H.-C. Kim, J.-C. Hur, J.-H. Lee, M.-J. Kang, "Pitch angle control of wind turbine based on variable PID gains," Journal of Korean Institute of Intelligent Systems, vol. 23, no. pp. 1-6, 2013.
- [6] Pena R., Clare J.C., Asher G.M., May 1996. "Doubly fed induction generator using back-to-back PWM converters and its application to variable-speed wind-energy generation". IEEE Proc.- Elec Power Appl. Vol. 143, No.3.
- [7] E. Hau, Wind Turbines: Fundamentals, Technologies, Application and Economics, Springer, 2nd Edition, 2005.
- [8] F. D. Bianchi, H. De Battista, R. J. Mantz, Wind Turbine Control Systems, Springer, 2007.
- [9] A. S. Yilmaz and Z. Özer, "Pitch angle control in wind turbines above the rated wind speed by multi-layer perceptron and radial basis function neural networks," Expert Systems with Applications, vol. 36, pp. 9767-9775, 2009.
- [10] J. M. Zurada, Introduction to Artificial Neural Systems, PWS Publishing Company, 1992.

[11] J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-MW Reference Wind Turbine for Offshore System Development. National Renewable Energy Laboratory, 2009.

## 감사의 글

대학원에 입학한지 어느덧 3년이란 긴 시간이 지나갔습니다. 많은 분들의 노력과 열정으로 풍력에 대하여 좋은 환경에서 많은 것을 배울 수 있었습니다. 긴 시간동안 모자란 저에게 끊임없이 지도와 편달을 아끼지 않아주신 강민제 교수님께 가장 먼저 존경과 감사의 말씀을 전하고 싶습니다. 그리고 바쁘신 중에도 본 논문을 자상하게 심사해주신 김경연 교수님과 김호찬 교수님께 깊이 감사드립니다. 그리고 학사시절 많은 가르침을 주신 전자공학과와 고성택 교수님, 김경식 교수님, 도양희 교수님, 이광만 교수님, 고석준 교수님께도 깊이 감사드립니다.

대학원을 다니는 동안 도움을 아끼지 않으며 풍력특성화 대학원과 학생들의 학업을 위해 오늘도 고생하시는 허종철 교수님께 늘 감사드리며 행정실 선생님들에게도 정말 감사하다는 말씀을 드리고 싶습니다.

전자공학과와 수많은 선,후배님들에게도 정말 고맙다는 말을 전하고 싶고, 같이 석사연구를 함께하는 연구실의 승윤이와 귀여운 후배들 상우, 지연, 지호에게도 항상 감사하고, 매일 같이 먹고 자고 생활하던 공대 고시반 친구와 선,후배님들에게도 감사합니다.

마지막으로 태어나서부터 지금까지 항상 저에 대한 관심과 사랑을 아끼지 않는 우리 아버지, 어머니와 동생에게 정말 감사하며 내 20대의 대부분이었던 희진 이에게도 감사의 마음을 전합니다.