

신경망을 이용한 풍력 발전시스템의 피치제어

홍민호¹, 고승윤¹, 김호찬², 허종철³, 강민제^{1*}

¹제주대학교 전자공학과, ²제주대학교 전기공학과, ³제주대학교 기계공학과

Pitch Angle Controller of Wind Turbine System Using Neural Network

Min-Ho Hong¹, Seung-Youn Ko¹, Ho-Chan Kim², Jong-Chul Hur³
and Min-Jae Kang^{1*}

¹Department of Electronic Engineering, Juju National University

²Department of Electrical Engineering, Juju National University

³Department of Mechanical Engineering, Juju National University

요 약 풍력발전시스템은 정격풍속미만에서는 토크를 제어하여 바람의 에너지를 최대로 하고 정격풍속이상에서는 피치를 제어하여 발전량을 정격으로 유지한다. 본 논문에서는 풍력발전시스템의 피치제어를 신경망을 이용하여 제어하는 방안을 제시한다. 피치제어의 목적은 정격풍속 이상에서 발전기의 회전속도를 일정하게 제어하여, 결과적으로 발전기의 출력을 정격전력으로 유지한다. 이 논문에서는 신경망 피치제어기의 성능을 향상시키기 위하여 발전기의 정격회전속도와 현재 회전속도 차이를 풍속과 함께 신경망의 입력으로 사용하는 방법을 제안하였다. 신경망의 훈련 알고리즘은 오류역전파(error back-propagation) 방법이 사용되었고, Matlab/Simulink를 사용하여 제어가 원활하게 되는 것을 확인하였다.

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 a method for wind turbine pitch controller using neural networks. The purpose of the pitch control is to control generator speed and power in the above rated wind speed. To improve the neural network pitch controller, the difference between a rated and current speed of generator has been used for another input of neural networks as well as wind speed. Error back-propagation algorithm is used for training the neural network pitch controller and simulation and Matlab/Simulink is used for verifying that this system is controlled well.

Key Words : Error back-propagation, Neural network controller, Pitch control, Wind turbine

1. 서론

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 관심이 되었고 앞으로 계속 증가할 전망이다. IEA(국제에너지기구)의 “World Energy Outlook 2012“ 보고서의 향후 25년간의

「본 연구는 2009년도 지식경제부의 재원으로 한국에너지기술평가원(KETEP) 풍력특성화대학원 인력양성사업의 지원을 받아 수행한 연구과제입니다.(NO.20094020200020)」

*Corresponding Author : Min-Jae Kang(Jeju National Univ.)

Tel: +82-64-754-3666 email: minjk@jejunu.ac.kr

Received January 6, 2014

Revised February 4, 2014

Accepted February 5, 2014

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 에너지를 개발하기 위하여 많은 노력을 기울이고 있고 풍력은 이러한 신재생에너지 중에서도 가장 기대되는 에너지원이다. 풍력 터빈(wind turbine)은 바람이 갖고 있는 에너지를 전기적 에너지로 변환하는 장치이다. 바람이 가지고 있는 에너지는 풍속이 3승에 비례한다. 마찬가지로 이 바람이 가지고 있는 에너지를 전기 에너지로 변환하는 풍력 터빈의 출력은 풍속의 3승에 비례하여 커진다. 하지만 고풍속 영역에서 풍속의 3승에 비례하는 출력을 얻기 위해서는 풍력 터빈 발전기의 용량이 커져야 하므로 그 무게도 엄청 늘어날 뿐만 아니라 비용이 효율적이지 못하다[2].

풍력을 효율적으로 사용하기 위해서 다양한 설계방법과 제어 방법들이 제시되고 있으며 제어방법에는 풍속에 따라 크게 두 가지로 나뉜다. 정격풍속보다 작은 풍속의 영역에서는 피치각(pitch angle)을 고정시키고 최대한 많은 양의 바람에너지를 회수하기 위해 발전기 토크(torque)를 제어하여 최적의 상태로 유지하고, 정격풍속보다 큰 풍속의 영역에서는 피치각을 제어하여 로터의 회전속도와 발전기의 발전량을 각각 정격 회전속도, 정격 발전량이 되도록 유지해야 한다[3].

본 논문에서는 NREL에서 제공하는 5MW급 풍력 터빈을 이용하여 정격풍속 이상에서의 출력을 제어하는 제어를 설계하였다. 정격풍속 이상의 영역에서 풍력 터빈의 출력을 일정하게 유지하기 위하여 피치를 제어하는 제어를 설계하였다. 제어기는 오류역전파 신경망(Error Back-Propagation Neural Network)을 사용하였으며, 이를 Matlab/Simulink에서 구현하여 정격풍속이상에서 발전기의 회전속도가 정격속도로 제어되고 있음을 확인하였다.

2. 풍력 터빈 시스템모델

2.1 풍력 터빈

풍력 터빈시스템은 바람의 운동 에너지를 전기 에너지로 변환시키는 장치이다. 여기서 풍력 터빈의 에너지원은 바람의 흐름으로 바람의 시간당 운동 에너지는 다음과 같다[4,5].

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

여기서 ρ 는 공기밀도[kg/m^3], A 는 블레이드(blade)의 단면적[m^2], v 는 풍속[m/s]이다. 본 논문에서 공기 밀도는 $1.2231[kg/m^3]$ 를 사용하였다. 위의 P_t 는 바람의 운동에너지이고 이 바람이 풍력 터빈을 통과하면서 풍력 터빈의 로터(rotor)로 전달되는 에너지는 다음과 같다.

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

위의 식(2)의 $C_p(\lambda, \beta)$ 는 전력계수라 하며, λ 와 β 의 두 변수를 가지는 비선형 함수로 λ 는 주속비(TSR: tip speed ratio)라 하고 현재 불어오는 바람에 대한 블레이드 끝단 속력의 비이며 다음과 같이 표현된다.

$$\lambda = \frac{\Omega_r R}{v} \quad (3)$$

여기서 Ω_r 은 로터의 각속도[rad/s]이고, R 은 로터의 반경[m]을 의미한다. β [deg]는 블레이드의 피치각이다. 또한 로터로 들어온 바람과 $C_p(\lambda, \beta)$ 를 이용하여 구한 P_r [W]을 로터의 각속도 Ω_r [rad/s]로 나누면 로터의 토크 T_r (N)값을 다음과 같이 구할 수 있다.

$$T_r = \frac{1}{2\Omega_r} \rho A v^3 c_p(\lambda, \beta) \quad (4)$$

본 논문에서 동력전달 장치 모델은 2-mass 모델을 사용하였으며 아래의 식(5)와 식(6)을 이용하여 발전기 회전속도 Ω_g [rad/s]을 구할 수 있다[6].

$$J_r \frac{d\Omega_r}{dt} = T_r - K_s(\theta_r - \frac{1}{N_g}\theta_g) - B_r(\Omega_r - \frac{1}{N_g}\Omega_g) \quad (5)$$

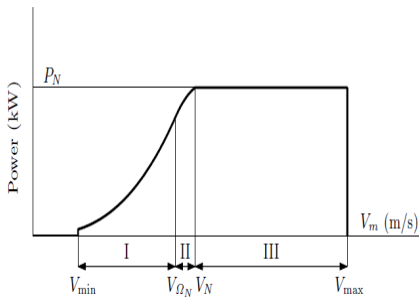
$$J_g \frac{d\Omega_g}{dt} = \frac{K_s}{N_g}(\theta_r - \frac{1}{N_g}\theta_g) + \frac{B_r}{N_g}(\Omega_r - \frac{1}{N_g}\Omega_g) - T_g \quad (6)$$

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, θ_r, θ_g 는 각각 저속 회전축(LSS)와 발전기축의 고속 회전축(HSS)의 각도를 의미한다. 또한 Ω_r, Ω_g 는 저속 회전축과 고속 회전축의 각속도를 의미하며, J_r 은 3개의 블레이드, 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.

2.2 풍력 터빈 시스템의 제어기모델

Fig. 1은 풍력 터빈의 풍속에 대한 파워의 출력 곡선을 나타낸다. 이 그래프는 시동풍속인 V_{min} 과 중단풍속 V_{out} 사이에서만 풍력 터빈이 동작한다는 것을 보여준다. 풍속이 V_{min} 이하에서는 운전비용에 대한 효율이 너무 적기 때문에 정지시키고 V_{out} 이상에서는 풍력터빈 시스템의 안전을 위하여 정지시킨다.

또한 V_{min} 과 V_{out} 사이에서 풍력 터빈의 작동 상태에 따라 위와 같이 3개의 영역으로 나누게 된다.



[Fig. 1] Ideal power curve of wind turbine systems

영역-I에서는 발전기의 정격 전력보다 낮은 부분으로 풍력 터빈으로 들어오는 바람을 최대한 많이 회수하도록 제어해주어야 한다. 그렇기 때문에 영역-I에서의 그래프는 바람의 3승에 비례하는 그래프가 보여 지며 이 영역에서는 피치각을 0으로 고정시키고 $C_p(\lambda, \beta)$ 의 값이 최대가 될 수 있도록 주속비 λ 를 일정한 값으로 유지시켜야 한다. 주속비 λ 를 일정하게 유지시키기 위해서 제어기에서 발전기 토크 $T_{g(ref)}$ [N]를 이용하여 제어하며 최대한의 에너지를 얻을 수 있도록 풍력 터빈을 제어한다.

영역-III에서는 풍력 터빈의 출력을 정격출력으로 일정하게 유지시켜야 한다. 이 영역에서는 바람에 따른 출력이 발전기의 정격 출력을 넘어서기 때문에 발전기의 정격 출력 만큼 바람의 에너지를 받을 수 있도록 피치각 β [deg]를 이용하여 제어하며 로터의 회전속도가 일정하도록 한다.

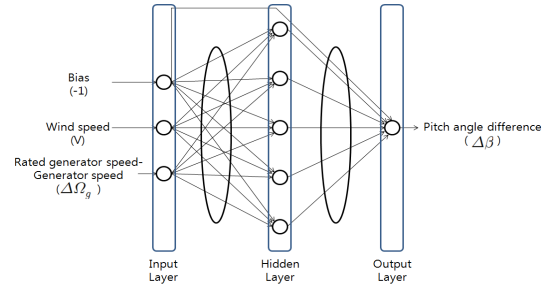
영역-II는 영역-I과 영역-III을 이어주는 부분으로 소음문제도 해결하고 로터의 하중을 줄여주기 위하여 부드럽게 넘어가도록 토크제어를 이용하여 제어한다[7,8].

본 논문에서는 정격풍속 이상에서 풍속의 변화에 따라 피치각 β [deg]를 제어하여 발전기의 회전속도와 발전량을 정격으로 유지하는 신경망 컨트롤러(Neural Network Controller 이하 NNC)의 설계방법을 제안하려고 한다.

3. 신경망을 이용한 피치제어

3.1 피치제어 신경망

본 논문에서 피치제어를 위해 오류역전파 신경망이 사용되었다[9]. Fig. 2에서 보여주듯이 풍속과 발전기의 회전속도 차이가 입력값으로 사용되고, 피치각의 변화가 출력값으로 나타난다. 그리고 은닉층에는 5 개의 뉴론을 사용하였다.



[Fig. 2] Structure of neural network controller

오류역전파 신경망에서 학습하는 과정을 효율과 안정적인 수렴을 위해 바이어스 뉴론이 입력단에 종종 사용되는 데, 여기서도 -1의 값을 갖는 바이어스 뉴론이 입력단에 사용되었다. 여기서 w_{ip} 는 입력층의 p 번째 뉴론과 은닉층의 i 번째 뉴론을 연결하는 가중치를 나타내고 w_i 는 은닉층의 i 번째 뉴론이 뉴론이 하나 뿐인 출력층과 연결하는 가중치를 나타낸다.

피치제어 신경망은 풍속의 변화와 현재 발전기의 회전속도에 맞는 피치각의 변화를 찾아서 발전기가 안정적으로 정격출력을 낼 수 있도록 한다. 신경망의 학습은 이런 조건을 만족할 수 있도록 뉴론들을 연결하는 가중치 w_{ip} 와 w_i 의 값을 찾아야 한다. 이를 위해서 임의의 값으로 가중치 w_{ip} 와 w_i 를 초기화하고 그 값을 통해 피치제어 신경망 출력의 오차를 계산하고 그 오차가 줄어들도록 가중치 w_{ip} 와 w_i 를 수정한다. 오차가 줄어들도록 가중치를 수정하는 식은 다음과 같다[10].

$$w_{ip}(t+1) = w_{ip}(t) + \frac{\alpha}{l} \sum_{n=1}^l x_{pn} f_1'(net_{i-}) E_{in} w_i \quad (7)$$

$$w_i(t+1) = w_i(t) + \frac{\alpha}{l} \sum_{n=1}^l E_{in} f_2'(net) Z_{in} \quad (8)$$

$$E(n) = f_2(n) - \hat{Y}(n) \quad (n = 1, 2, \dots, l) \quad (9)$$

$$E_{total} = \frac{\sum(E(n))^2}{l} \quad (n = 1, 2, \dots, l) \quad (10)$$

여기서 $X(n)$ 은 피치제어 신경망 입력들의 벡터이고 $Z(n)$ 은 은닉층의 뉴론들의 출력 벡터이다. $f_2(n)$ 는 출력값의 변환함수, $\hat{Y}(n)$ 은 목표값, f_1 은 은닉층의 변환함수 나타낸다. net_i 와 net 는 각각 은닉층 i 번째 뉴론의 입력값과 출력뉴론의 입력값을 나타낸다. 그리고 l 은 훈련 사이즈, α 는 훈련 비율을 나타낸다. 은닉층 뉴론의 변환함수는 -1에서 1 사이의 값을 갖는 시그모이드 (sigmoid) 함수로 식(11)과 같고 출력뉴론의 변환함수는 식(12)와 같이 선형함수이다.

$$f_1(u) = \frac{1 - e^{-u}}{1 + e^{-u}} \quad (11)$$

$$f_2(u) = u \quad (12)$$

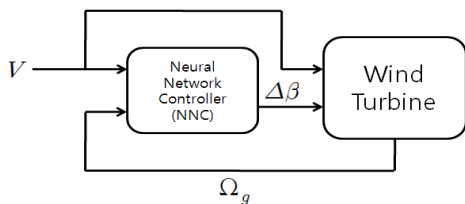
이와 같은 방법으로 전체적인 에러인 식(10)이 원하는 값 이하가 될 때 까지 가중치의 수정과정을 반복한다.

3.2 피치제어를 위해 제안된 알고리즘

앞 장에서 설명 했듯이 영역-III에서는 풍력 터빈의 출력을 정격출력으로 일정하게 유지되어야 한다. 또한 풍력 터빈의 하중을 줄이고 안전하게 동작하기 위하여 로터의 회전속도 또한 정격 속도로 유지시켜야 한다.

입력을 현재 들어오는 풍속 하나만 놓아서 제어할 경우에는 NNC의 훈련 에러는 훨씬 줄어든다. 하지만 이렇게 훈련된 가중치 값들을 NNC에 넣고 풍력 터빈에 적용시킨다면 정격풍속 이상에서의 피치각은 풍속에 비례하여 나온다. 이 결과는 그저 들어오는 풍속에 대한 Look-up table이 될 뿐이고 결국 로터의 회전속도는 일정속도를 유지할 수 없게 된다.

본 논문에서 제안하는 알고리즘은 현재 들어오는 풍속을 하나의 입력으로 받고 현재 발전기의 회전속도와 정격 발전기 회전속도의 차이를 또 하나의 입력으로 받아 정격 발전기 회전속도보다 현재 발전기 회전속도가 높으면 피치각을 증가시켜 발전기의 회전속도를 낮추고 정격 발전기 회전속도보다 현재 발전기 회전속도가 낮으면 피치각을 감소시켜 발전기의 회전속도를 높인다. 이런 제어 알고리즘을 기반으로 신경망 모델을 훈련시키고 풍력 터빈에 적용하여 제어하는 방식을 제안한다.



[Fig. 3] Block diagram of pitch controller system with NNC

Fig. 3은 정격풍속 이상에서 NNC을 이용한 피치제어 시스템의 블록다이어그램을 나타내며 이를 순차적으로 표현하면 아래와 같다.

- I) 풍속 V [m/s]가 풍력 터빈으로 들어가 현재 발전기 회전속도인 Ω_g [rad/s]를 구한다.
- II) 풍속 V [m/s]와 발전기 회전속도 Ω_g [rad/s]가 NNC의 입력으로 들어가고 NNC는 정격 발전기 회전속도와 현재 발전기 회전속도의 차이를 계산하고 이에 맞는 피치각 β [deg]의 증감인 $\Delta\beta$ [deg]를 계산하여 풍력 터빈으로 보낸다.
- III) 현재 풍속 V [m/s]와 현재 피치각 β [deg]에서 NNC로부터 받은 $\Delta\beta$ [deg]를 합한 β [deg]를 가지고 다시 Ω_g [rad/s]를 구한다.
- IV) 위 과정을 계속 반복하며 정격발전기 회전속도와 현재 발전기 회전속도의 차이를 줄이도록 풍력 터빈의 피치각을 제어한다.

3.3 신경망 훈련요소

앞 절에서 제시한 알고리즘을 따라 NNC를 훈련시켜야 하는데 NNC의 훈련을 위하여 고려해야할 요소들은 첫째 전체 훈련 개수, 둘째 풍속의 범위, 셋째 발전기 회전속도 차이의 범위 그리고 넷째 피치각의 변화 정도이다.

[Table 1] Inputs and desired output of NNC

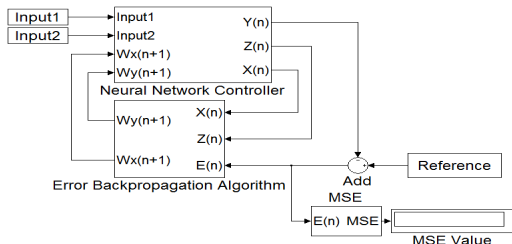
Input1	Input2	Reference	Input1	Input2	Reference
11	-20	1	17	-7	1
11	-15	1	17	-6	1
11	-5	0.7	17	-3	0.7
11	-2	0.1	17	-2	0.1
11	0	0	17	0	0
11	2	-0.1	17	2	-0.1
11	5	-0.7	17	3	-0.7
11	15	-1	17	6	-1
11	20	-1	17	7	-1
13	-15	1	19	-6	1
13	-10	1	19	-5	1
13	-4	0.7	19	-2	0.7
13	-2	0.1	19	-1	0.1
13	0	0	19	0	0
13	2	-0.1	19	1	-0.1
13	4	-0.7	19	2	-0.7
13	10	-1	19	5	-1
13	15	-1	19	6	-1
15	-13	1	21	-5	1
15	-8	1	21	-4	1
15	-3	0.7	21	-2	0.7
15	-2	0.1	21	-1	0.1
15	0	0	21	0	0
15	2	-0.1	21	1	-0.1
15	3	-0.7	21	2	-0.7
15	8	-1	21	4	-1
15	13	-1	21	5	-1

전체 훈련 개수는 너무 적으면 그만큼 훈련오차가 줄겠지만 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 NNC의 입력풍속은 1[m/s]부터 21[m/s]까지 정하였다. 그리고 정격 발전기 회전속도와 현재 발전기 회전속도의 차이범위는 -20에서 +20사이에서 결정하였다. 실험을 통해 다양한 범위를 적용해 보았고 이 정도의 범위가 적당함을 알게 되었다. 피치각의 변화정도는 -1과 1 사이의 값을 가지도록 훈련시켰으며 훈련풍속 중에도 저풍속일 때는 정격 발전기 회전속도와 현재 발전기 회전속도의 차이에 따라 비교적 피치각의 변화를 작게, 고풍속에서는 피치각의 변화를 크게 하도록 훈련시켰다.

Table 1은 NNC훈련의 두 개의 Input과 하나의 Reference값을 보여준다. Input1은 풍속을 나타내고 Input2는 정격 발전기 회전속도에서 현재 발전기 회전속도를 뺀 값이다. 만약 Table 1과같이 Input1을 11m/s부터 21m/s까지 6개의 풍속으로 잡고 각 풍속당 Reference를 9개로 잡는다면 전체 훈련개수는 54개가 된다. 이때의 Reference는 각각 -1, -1, -0.7, -0.1, 0, 0.1, 0.7, 1, 1 로 고정시키고 발전기 회전속도의 차이는 각 풍속마다 차이를 두고 훈련시켰다.

4. 시뮬레이션 및 결과

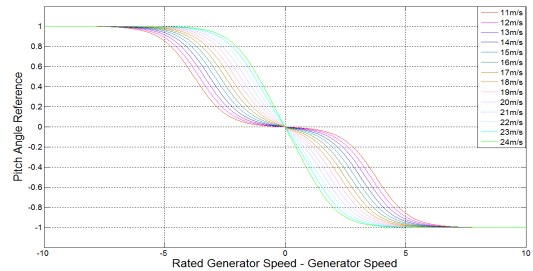
Fig. 4는 3.1에서 설명한 피치제어 신경망의 학습을 위해 Matlab/Simulink에서 구현한 것이다. 부수적으로 학습 과정에서 실시간으로 전체에러가 변하는 모습을 보기 위하여 Scope를 달았다.



[Fig. 4] Block diagram of Matlab/Simulink for error back-propagation algorithm

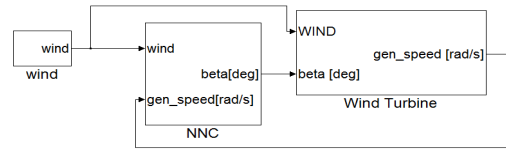
이렇게 구현된 시스템에서 학습하여 목표한 에러에 만족하는 뉴런 연결 가중치들을 구하였다. 이렇게 하여 구한 뉴런 연결 가중치들이 Table 1에 포함되지 않은 풍속

에서도 제대로 작동하는지를 알기 위하여 학습된 가중치를 NNC에 넣고 테스트하였다. 풍속은 11[m/s]부터 24[m/s]까지 1[m/s]씩 증가하고 발전기 회전속도의 차이는 -10에서 10까지 변하는 값을 입력으로 넣어서 실험하였다. Fig. 5에서 보면 훈련되지 않은 풍속의 결과에도 문제가 없는 것을 알 수 있다.



[Fig. 5] Test outputs of NNC for untrained region

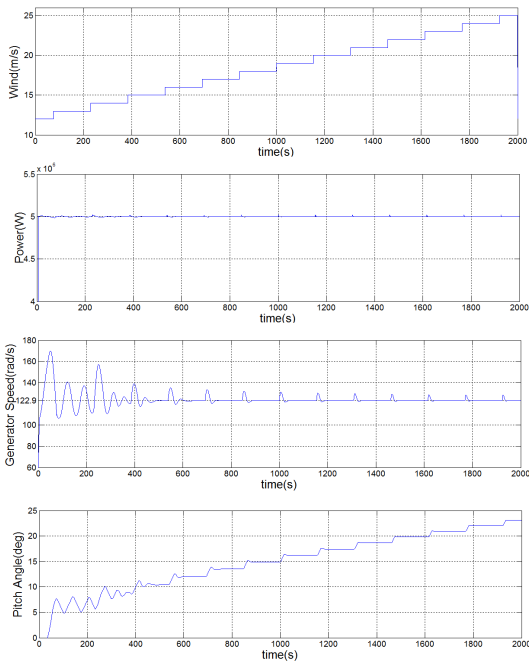
Fig. 4에서 학습된 NNC를 Fig. 3의 피치제어시스템에 적용하였으며, 이것을 Matlab/Simulink로 구현한 것이 Fig. 6에 나타나 있다.



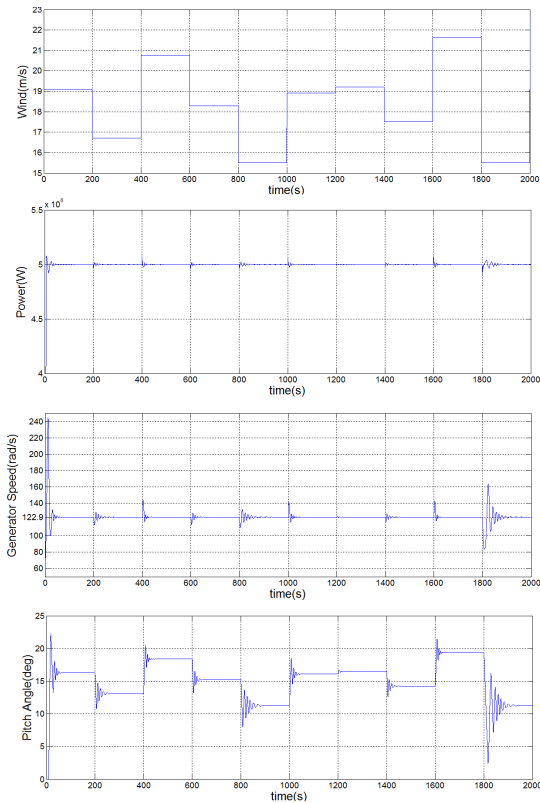
[Fig. 6] Block diagram of Matlab/Simulink for wind turbine system with NNC

여기서 풍력 터빈의 모델은 NREL에서 제공하는 5MW급 풍력 터빈 Matlab/Simulink모델을 수정하여 사용하였다[11]. 가장 왼쪽의 블록에서 풍속을 생성하여 보내주면 이 풍속이 각각 NNC와 풍력 발전기로 들어가며 NNC에서는 첫 입력으로 들어온 풍속과 풍력 발전기에서 계산된 현재 발전기 회전속도가 NNC의 두 번째 입력으로 들어가 피치각의 증감의 값을 계산하고 피치각을 제어하여 발전기의 회전속도를 일정한 속도로 유지한다.

Fig. 7은 풍속을 일정하게 증가시키며 살펴본 결과이다. 풍속은 계단함수 형식으로 12부터 1씩 25까지 증가하는 형식을 입력으로 하여 발전량과 현재 발전기의 속도, 그리고 피치각의 변화를 나타내었다. 이 제어 시스템은 피치각의 증감을 이용하여 풍력발전기의 발전기 회전속도가 정격풍속 이상에서도 일정하게 유지되도록 제어하는 것을 목표로 하며 위의 그림을 보았을 때 풍속이 증가하였을 때 발전기의 속도는 약간의 과도상태를 보이지만 결국 정격속도인 122.9[rad/s]가 되도록 제어 되는 것을 볼 수 있다.



[Fig. 7] Simulation results of wind turbine with step wind speed



[Fig. 8] Simulation results of wind turbine with random wind speed

Fig. 8은 고풍속에서 200초마다 불규칙적으로 바뀌는 풍속을 입력으로 했을 때의 풍력 터빈의 출력, 현재 발전기 회전속도, 피치각의 변화를 보여준다. 풍속의 변화가 급격할수록 발전기 회전속도의 오버슈트가 커지고 과도상태의 시간이 길어지긴 하지만 곧 정격 발전기 회전속도인 122.9[rad/s]로 제어되는 것을 볼 수 있다.

5. 결론

본 논문에서는 신경망을 이용하여 정격풍속이상에서 풍력 터빈의 피치각을 제어하여 정격회전속도와 출력을 제어하는 시스템을 제안하였다. 신경망은 오류역전파 신경망을 사용하였으며 입력단에는 두 개의 입력과 한 개의 바이어스, 은닉층에는 5개의 뉴런 그리고 출력은 피치각의 변화를 나타내는 1개의 뉴런으로 구성하였다.

입력을 현재 들어오는 풍속 하나로만 놓아서 제어를 할 경우에는 정격풍속 이상에서의 피치각은 풍속에 비례하여 나온다. 이 결과는 그저 들어오는 풍속에 대한 Look-up table이 될 뿐이고 현재 풍력터빈의 회전속도상태가 고려되지 않아서 로터의 회전속도는 제어되지 않는다.

본 논문에서 입력으로 풍속과 현재 발전기의 회전속도와 정격 발전기 회전속도의 차이를 또 하나의 입력으로 받아서 피치각을 제어하는 시스템을 제안하였다. 이렇게 하여 훈련된 신경망 피치제어 기는 Fig. 7, Fig. 8과 같이 정격풍속이상에서 발전기 회전속도와 전력이 제어되는 것을 볼 수 있다. 다만 풍속이 변할 때 약간의 과도상태가 생기는 것을 알 수 있으며 특히 12[m/s]~ 13[m/s]의 풍속구간에서 오버슈트가 조금 큰 것은 앞으로 계속 개선해 나가야할 연구과제로 생각한다.

References

- [1] International Energy Agency, World Energy Outlook, 2012. OECD/IEA, 2012.
- [2] H.-C. Kim, A Study on the Control Algorithm Improvement for Small Wind Power Systems, Korea Electrical Engineering & Science Research Institute, 1998.
- [3] J. G. Kim, Power Control of a MW Class Wind Turbine, Master Thesis, Kangwon National University, 2011.
- [4] K.-N. Ko and J.-C. Hur, An Introduction to Wind Engineering, Munundang, 2007.
- [5] T. Burton, D. Sharpe, N. Jenkins, and E. Bossanyi, Wind Energy Handbook, Wiley, 2001.
DOI: <http://dx.doi.org/10.1002/0470846062>
- [6] 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.

[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.
DOI: <http://dx.doi.org/10.1016/j.eswa.2009.02.014>

[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.
DOI: <http://dx.doi.org/10.2172/947422>

홍민호 (Min-Ho Hong) [준회원]



- 2011년 2월 : 제주대학교 전자공학과 (공학사)
- 2011년 3월 ~ 현재 : 제주대학교 풍력 특성화 협동과정 (공학석사)

<관심분야>
풍력 발전 제어, 피치 제어

고승윤 (Seung-Youn Ko) [준회원]



- 2013년 2월 : 제주대학교 전자공학과 (공학사)
- 2013년 3월 ~ 현재 : 제주대학교 풍력 특성화 협동과정 (공학석사)

<관심분야>
풍력 발전 제어, 전기발전 제어

김호찬 (Ho-Chan Kim) [정회원]



- 1989년 2월 : 서울대학교 제어계측공학과 (공학석사)
- 1994년 2월 : 서울대학교 제어계측공학과 (공학박사)
- 2008년 2월 ~ 2009년 2월 : 미국 펜스테이트 방문교수
- 1995년 9월 ~ 현재 : 제주대학교 전기공학과 교수

<관심분야>
신재생에너지, 풍력발전, 접지시스템 설계, 에너지효율

허종철 (Jong-Chul Huh) [정회원]



- 1982년 2월 : 인하대학교 기계공학과 (공학석사)
- 1992년 2월 : 인하대학교 기계공학과 (공학박사)
- 1984년 3월 ~ 현재 : 제주대학교 기계공학과 교수
- 2009년 3월 ~ 현재 : 제주대학교 풍력특성화 대학원 인력양성 사업단 단장

<관심분야>
풍력 발전 시스템

강민제 (Min-Jae Kang) [정회원]



- 1982년 2월 : 서울대학교 전기공학과 (공학사)
- 1991년 2월 : 미국 루이빌대 전기공학과 (공학박사)
- 2003년 2월 ~ 2004년 2월 : 미국 일리노이주립대학 방문교수
- 1992년 3월 ~ 현재 : 제주대학교 전자공학과 교수

<관심분야>
신경회로망, 접지시스템 설계, 풍력 발전 제어