

도플러 레이더와 유한상태기계를 이용한 풍력 블레이드의 회전 속도 추출 방법

최영재*, 최인식**

Rotation Speed Extraction Method of Wind Turbine Blade Using Doppler Radar and Finite State Machine

Young-Jae Choi*, In-Sik Choi**

본 연구는 한국전력공사의 사외공모 기초연구(개별)에 의해 지원되었음(과제번호: R19X001-50).
이 논문은 2018년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2018R1D1A1B07041496).

요 약

도플러 레이더로 관측한 풍력 블레이드의 데이터는 풍력 블레이드의 회전 속도에 대한 정보를 가지고 있다. 본 논문에서는 도플러 레이더로 회전하는 풍력 블레이드를 관찰할 때 수신되는 데이터를 이용하여 풍력 블레이드의 회전 속도를 재귀적인 절차(Recursive process)로 계산하는 방법을 제안한다. 제안된 방법은 도플러 빈 생성기, 도플러 빈 패턴 검출기, 시간 패턴 검출기로 구성되어 있다. 시간 패턴 검출기는 유한상태기계 (Finite state machine)를 기반으로 설계 하였으며, 서포트 벡터 머신(Support vector machine)으로 구현 하였다. 제안된 방법을 검증하기 위해 합성 데이터와 수치해석 데이터에 대하여 시뮬레이션을 수행하였다. 시뮬레이션 결과로부터 제안된 방법이 성공적으로 회전 속도를 검출함을 확인할 수 있다.

Abstract

The wind turbine blade data observed by the Doppler radar contains information on the rotational speed of that. In this paper, we propose a recursive process that extracts the rotational speed of a wind turbine blade using the data of the rotating wind turbine blade observed by a Doppler radar. The proposed method consists of a Doppler bin generator, a Doppler bin pattern detector and a time pattern detector. The time pattern detector is designed based on a finite state machine and implemented using a support vector machine. In order to verify the proposed method, simulations are performed on the synthetic data and the numerical data. the simulation result shows that the proposed method successfully detects the rotational speed.

Keywords

dynamic radar cross section, wind turbine blade, finite state machine, doppler radar, support vector machine

* 한남대학교 전기전자공학과 박사과정
- ORCID: <https://orcid.org/0000-0002-8632-5265>
** 한남대학교 전기전자공학과 교수(교신저자)
- ORCID: <https://orcid.org/0000-0002-8210-0843>

• Received: Jan. 21, 2020, Revised: Feb. 21, 2020, Accepted: Feb. 24, 2020
• Corresponding Author: In-Sik Choi
Dept. of Electrical and Electronic Engineering, Hannam University, 70
Hannam-ro, Daeduk-gu, Daejeon, 34430, Korea
Tel.: +82-42-629-8568, Email: recog@hnu.kr

1. 서 론

풍력 발전기는 바람으로부터 전력을 얻는 방식의 친환경 발전 장비이다. 바람의 힘을 회전력으로 전환하는 풍력 발전의 원리와 풍력 발전기가 가지고 있는 거대한 블레이드의 크기 때문에, 풍력 발전기는 야외에 다수의 발전기를 설치하여 운영하는 경우가 많다[1]. 특히 해상 풍력 발전의 경우 풍력 발전기가 바다에 설치되므로 접근성이 좋지 않아 원격지에서 풍력 발전기의 동작 상태를 확인할 수 있는 기술이 필요하다[2]. 도플러 레이더의 스펙트로그램은 표적의 모션에 대한 정보를 가지고 있으며 이를 분석하여 표적의 움직임에 대한 정보를 얻는 연구들이 진행되어 왔다[3]-[5]. 참고문헌 [3]은 스펙트로그램의 전처리 이후 특성벡터를 추출한 뒤 구분기에 전달하는 방식을 사용하고 참고문헌 [4]는 합성곱신경망(Convolutional neural networks)을 사용하여 도플러 레이더의 스펙트로그램으로부터 사람의 움직임을 분류하였다. 참고문헌 [5]는 스펙트로그램으로부터 사람의 움직임을 분류하기 위한 새로운 구분기를 제안하였다.

이러한 연구들에서는 특성의 추출이 까다로운 스펙트로그램 영상을 분석하기 위하여 영상 전체를 분류에 활용해야만 했다. 이러한 방식은 일괄처리(Batch process) 방법으로 볼 수 있다. 그러나 풍력 블레이드와 같이 동작이 단순하고 주기성을 갖는 표적의 경우 스펙트로그램이 매우 정형화 된 형태로 나타나며 동작의 특성 역시 스펙트로그램 영상으로부터 곧바로 추출이 가능하다[6].

그러나 매번 데이터를 샘플링할 때마다 스펙트로그램 다시 계산하는 일괄 처리 방법으로 특성을 추출할 경우에는 지나치게 많은 메모리 공간과 긴 계산 시간을 필요로 한다. 따라서 이러한 일괄처리 방법 대신 현재 시간에 수집되는 데이터로 내부의 상태 변수를 업데이트하고 이러한 상태변수로부터 특성을 추출하는 방식의 재귀적인 처리절차(Recursive process)로 동작하는 알고리즘을 사용함으로써 이러한 문제를 개선할 수 있다.

본 논문에서는 현재 시간에 수집되는 도플러 빈(Doppler bin)만을 활용하는 재귀적인 처리 방법으로

스펙트로그램상의 특성을 추출하고 이를 이용하여 풍력 블레이드의 회전 속도를 추출하는 방법을 제안한다.

본 논문에서 제안한 방법은 데이터를 취득하는 시점에서는 해당 시점의 도플러 빈만을 계산하고 이 도플러 빈의 판별 결과를 이용하여 내부의 상태 변수를 업데이트 하는 방식으로 스펙트로그램을 직접 구하지 않고도 스펙트로그램 상의 특성을 추출할 수 있다. 제안된 방법을 검증하기 위해 신호 모델과 수치해석을 사용하여 회전하는 풍력 블레이드의 스펙트로그램을 생성하고 계산된 스펙트로그램에 제안된 방법을 적용하여 풍력 블레이드의 회전 속도를 추정하는 시뮬레이션을 수행하였다.

II. 풍력 블레이드의 스펙트로그램

회전하는 블레이드의 Ideal point scatterer model은 식 (1)과 같이 나타낼 수 있다[7].

$$s(t) = \sum_{k=1}^N xp \left\{ -j \frac{4\pi}{\lambda} [R_0 + z_0 \sin \beta] \right\} \times \text{sinc}(\Phi_k(t)) \exp\{-j\Phi_k(t)\} \quad (1)$$

여기서 L 은 블레이드의 길이, R_0 는 안테나와 블레이드 중심 사이의 거리, z_0 는 블레이드와 지면 사이의 높이, β 는 안테나가 표적을 바라보는 방향과 풍력 블레이드가 회전하는 평면 사이의 각도, λ 는 레이더 송신 신호의 파장, N 은 블레이드의 개수이다. $\Phi_k(t)$ 는 식 (2)와 같이 정의된다.

$$\Phi_k(t) = \frac{4\pi}{\lambda} \frac{L}{2} \cos \beta \cos \left(\Omega t + \phi_0 + k \frac{2\pi}{N} \right) \quad (2)$$

여기서 Ω 는 블레이드의 회전 각속도이고 ϕ_0 는 블레이드의 초기 회전각이다.

식 (2)에서 블레이드의 회전 각속도 Ω 는 시간에 따라 변화할 수 있다. 이러한 각속도의 변화가 1차 함수로 근사될 수 있다고 가정하고 Ω 를 식 (3)과 같이 정의한다.

$$\Omega = \frac{1}{2} at + \Omega_0 \quad (3)$$

여기서 a 는 블레이드의 회전 각 가속도이고, Ω_0 는 블레이드의 초기 회전 각속도이다.

식 (3)을 식 (2)에 대입하여 식 (4)를 구한다.

$$\Phi_k(t) = \frac{4\pi}{\lambda} \frac{L}{2} \times \cos\beta \cos\left(\frac{1}{2}at^2 + \Omega_0t + \phi_0 + k\frac{2\pi}{N}\right) \quad (4)$$

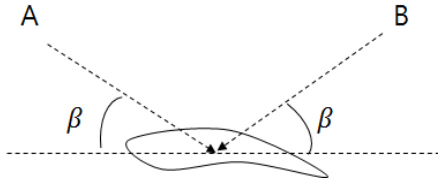


그림 1. 풍력 블레이드의 단면 형상

Fig. 1. Shape of wind turbine blade cross section

그림 1은 풍력블레이드의 단면의 형상을 보여준다. 그림 1에서 블레이드가 레이더를 향해 다가오는 경우 레이더가 표적을 A 방향에서 관측한다고 가정하면 블레이드가 레이더에서 멀어지는 경우에는 레이더는 블레이드를 B 방향에서 관측하게 된다. A와 B 방향에서 레이더가 바라보는 풍력 블레이드의 형상은 서로 다르기 때문에 해당 방향으로 반사되는 신호의 강도도 다르다. 이러한 영향을 반영하기 위하여 식 (1)을 식 (5)와 같이 수정한다.

$$s(t) = \sum_{k=1}^N xp\left\{-j\frac{4\pi}{\lambda}[R_0 + z_0\sin\beta]\right\} \times [\sigma(\beta)\text{sinc}(\Phi_k(t))\exp\{-j\Phi_k(t)\}] \quad (5)$$

여기서 $\Phi_k(t)$ 는 식 (4)를 사용하고, $\sigma(\beta)$ 는 풍력 블레이드에서 반사되는 신호의 강도로 식 (6)과 같이 정의한다.

$$\sigma(\beta) = \begin{cases} \sigma_h(\beta), & 0 < \text{mod}(\theta(t), 2\pi) \leq \pi \\ \sigma_l(\beta), & \text{others} \end{cases} \quad (6)$$

여기서 $\text{mod}(x,y)$ 는 x 를 y 로 나눈 나머지를 구하는 연산이고, $\theta(t) = \frac{1}{2}at^2 + \Omega_0t + \phi_0 + k\frac{2\pi}{N}$ 이다.

그림 2는 표 1의 매개변수와 식 (5)를 사용하여 구한 $s(t)$ 의 스펙트로그램의 예시를 보여준다.

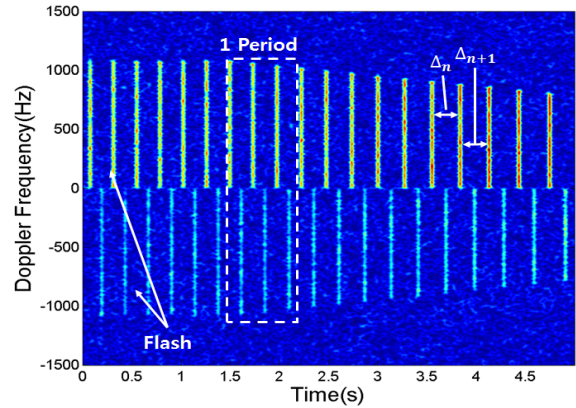


그림 2. 신호 모델로 생성된 스펙트로그램

Fig. 2. Spectrogram generated by signal model

표 1. 그림 2의 매개변수

Table 1. Parameters in figure 2

Parameter	Value
L	1
N	3
R_0	1000 (m)
λ	0.0125(m)
sampling rate	3000 (sps)
β	40 (degree)
Ω_0	2π (rad/sec)
z_0	80 (m)
ϕ_0	50 (degree)
a	0.6981 (rad/sec)
σ_h	1
σ_l	0.3
observation time	2.5 (sec)
SNR (Signal to noise ratio)	10 dB

그림 2의 스펙트로그램은 풍력 블레이드의 회전 속도가 일정하게 증가하는 경우에 해당한다. 풍력 블레이드의 스펙트로그램은 블레이드가 안테나에 가장 넓은 면적을 보여줄 때 발생하는 플래쉬(Flash)가 주기적으로 반복되는 형태로 구성되어 있다.

플래쉬는 풍력 블레이드의 스펙트로그램에서 가장 강건한 특성들 중 하나이다. 양의 도플러 주파수에 있는 플래쉬와 음의 도플러 주파수에 있는 플래쉬의 신호 강도는 서로 다르며 풍력 블레이드가 한 바퀴 회전하는 동안 이 두 종류의 플래쉬가 번갈아가며 3쌍이 나타난다. 이러한 특성은 회전하는 블레이드의 스펙트로그램을 측정하는 다른 연구들에서도 공통적으로 나타난다[6][8][9].

이 두 종류의 플래쉬들 중 신호 강도가 강한 플래쉬를 선택했을 때 n 번째 플래쉬와 $n-1$ 번째 플래쉬 사이의 시간 간격을 $\Delta_n(\text{sec})$ 라 하고 $\Delta_n(\text{sec})$ 동안 풍력 블레이드는 반드시 120° 회전한다. 따라서 $\Delta_n(\text{sec})$ 사이의 평균 회전 각속도 Ω_n 은 식 (7)과 같은 관계를 갖는다.

$$\Omega_n = \frac{2\pi}{3\Delta_n} \quad (7)$$

식 (7)은 스펙트로그램의 Δ_n 으로부터 해당 시간 범위에서의 평균 회전 각속도 Ω_n 을 구할 수 있음을 보여준다.

III. 제안하는 방법

3.1 개요

본 장에서는 풍력블레이드의 스펙트로그램에서 Δ_n 시간 동안의 평균 회전 각속도 Ω_n 을 구하는 방법을 설명한다. 그림 3은 본 논문에서 제안하는 방법의 전체 블록도를 보여준다.

그림 3의 도플러 빈 생성기(Doppler bin generator) 블록에서는 레이더 수신 데이터 $s(t)$ 를 입력받아 해당 시간 샘플에서의 도플러 빈 \vec{D} 를 생성한다. 여기서 W 는 도플러 빈의 생성을 위한 윈도우의 크기이다. 이렇게 생성된 \vec{D} 를 도플러 빈 패턴 검출기(Doppler bin pattern detector)에서 플래쉬를 검출하기 위해 필요한 몇 가지 유형으로 분류한다. 시간 패턴 검출기는 이러한 결과값으로부터 플래쉬의 검출 여부를 판단한다. 플래쉬가 검출되는 경우 이전 플래쉬가 검출된 시점부터 현 시점까지의 시간 샘플의 개수를 세고 여기에 샘플링 주기 d 를 곱하여 Δ 를 구한다. 여기서 n 번째로 검출된 플래쉬에 대한 Δ 를 Δ_n 이라 한다.

3.2 도플러 빈 생성기

스펙트로그램의 재귀적인 분석의 첫 번째 단계는 도플러 빈을 생성하는 것이다. 도플러 빈은 그림 4와 같이 스펙트로그램에서 현재 시간 샘플에 대한 도플러 스펙트럼을 의미한다.

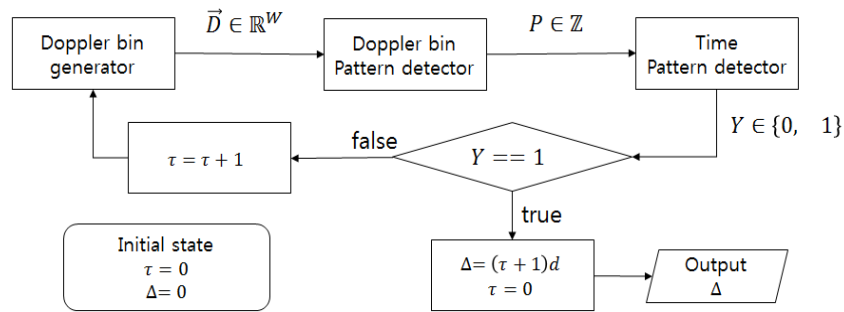


그림 3. 제안한 방법의 블록도
Fig. 3. Block diagram of proposed method

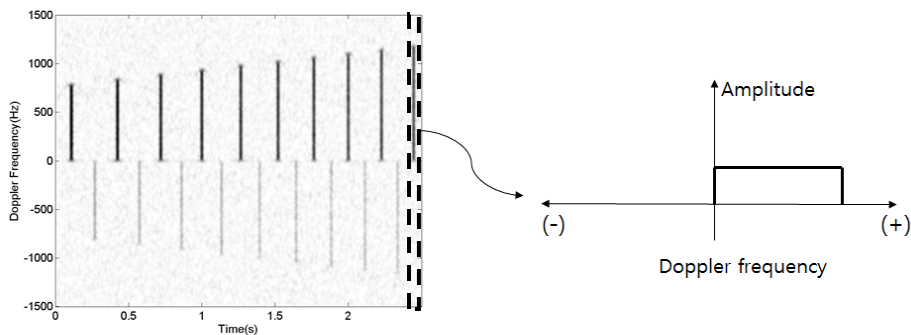


그림 4. 도플러 빈의 개념
Fig. 4. Concept of doppler bin

도플러 빈 \vec{D} 는 식 (8)과 같은 방법으로 구할 수 있다.

$$\vec{D} = |FT(\vec{W} \circ \vec{B})| \quad (8)$$

여기서 $FT(\cdot)$ 는 푸리에 변환(Fourier transform) 연산이며 \vec{B} 는 식 (9)과 같이 정의한다.

$$\vec{B} = [s(t_p) s(t_{p-1}) \dots s(t_{p-W+1})]^T \quad (9)$$

식 (8)에서 $\vec{W} \in \mathbb{C}^W$ 는 윈도우 함수(Window function)이며 본 논문에서는 가우시안 윈도우를 사용하였다. \circ 는 행렬의 원소끼리 곱을 하는 아다마르 곱(Hadamard product)이다[10]. W 는 \vec{D} 와 \vec{W} 의 길이이다. 이러한 방법을 이용하여 새로운 시간 샘플에서의 데이터 $s(t_p)$ 가 주어질 때마다 새로운 도플러 빈을 얻을 수 있다.

3.3 도플러 빈 패턴 검출기

도플러 빈 \vec{D} 는 W 차원의 벡터로써 매우 많은 경우의 수를 가지고 있다. 그러나 스펙트로그램에서 특정 패턴을 검출하기 위해 필요한 도플러 빈의 패턴은 극히 일부이다. 이해를 돕기 위하여 그림 5와 같은 임의의 패턴 검출 문제를 생각해 보자.

그림 5에서 스펙트로그램 상의 임의의 패턴을 검출하는데 필요한 도플러 빈은 해당 패턴이 가지고 있는 도플러 빈의 패턴 뿐 이다.

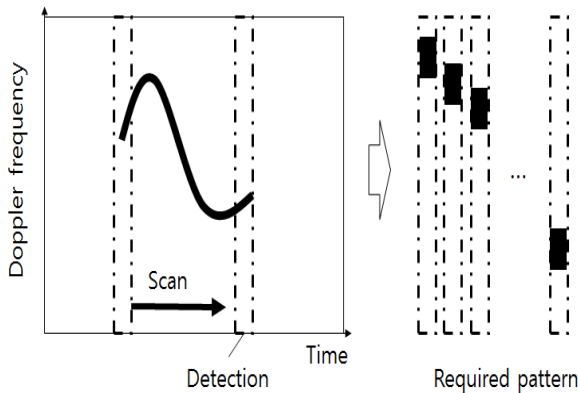


그림 5. 패턴 검출 예시

Fig. 5. Pattern detection example

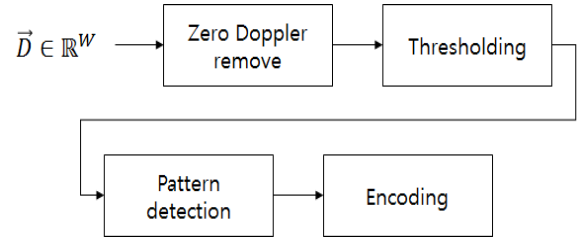


그림 6. 도플러 빈 패턴 검출기의 블록도

Fig. 6. Block diagram of doppler bin pattern detector

본 절에서 설명하는 도플러 빈 패턴 검출기는 주어진 \vec{D} 로부터 스펙트로그램의 패턴 검출에 필요한 패턴만을 검출하는 역할을 한다. 그림 6은 본 논문에서 제안하는 도플러 빈 패턴 검출기의 블록도이다.

도플러 레이더를 통해 획득한 \vec{D} 에는 클러터로부터 반사된 영 도플러(Zero doppler) 성분이 강하게 존재한다. 따라서 신호처리를 수행하기 전 \vec{D} 에서 관심 있는 도플러 주파수 이하의 낮은 도플러 주파수 성분에 대하여 제로 패딩(Zero-padding)을 수행한다. 제로 패딩 된 \vec{D} 를 임계치 G 를 기준으로 이진화한다.

$$b_j = \begin{cases} 1, & d_j > G \\ 0, & \text{others} \end{cases} \quad (10)$$

여기서 임계치는 $G = c\mu_p$ 와 같이 결정하며 여기서 c 는 상수이다. $\vec{D}_{bin} = [b_1 b_2 \dots b_W]^T$ 이고 $\vec{D} = [d_1 d_2 \dots d_W]^T$ 이다. μ_p 는 식 (11)과 같이 저주파 필터를 사용하여 적응시킨다.

$$\mu_p = \alpha\mu_{p-1} + (1-\alpha)\frac{1}{W}\sum_{w=1}^W d_w \quad (11)$$

여기서 α 는 $0 \leq \alpha \leq 1$ 인 상수이다.

\vec{D}_{bin} 의 관심영역에 신호가 존재하는지를 확인하기 위한 마스킹 벡터 $\vec{M}_i = [m_1 m_2 \dots m_W]^T$ 를 그림 7과 같이 정의하자.

\vec{M}_i 에서 관심 영역에 해당하는 원소는 0 ~ 1 사이의 실수를 대입하고 나머지 영역은 0을 대입한다.

이때 $\sum_{w=1}^W m_w = 1$ 이 되도록 원소의 값을 결정한다.

이와 같이 정의한 \vec{M}_i 를 사용하여 i 번째 마스킹 벡

터에 대한 검출 점수 x_i 를 식 (12)과 같이 계산한다.

$$x_i = \begin{cases} 1, & (\overrightarrow{M_i^T D_{bin}}) > \epsilon \\ 0, & \text{others} \end{cases} \quad (12)$$

여기서 ϵ 은 0과 1 사이의 실수이다.

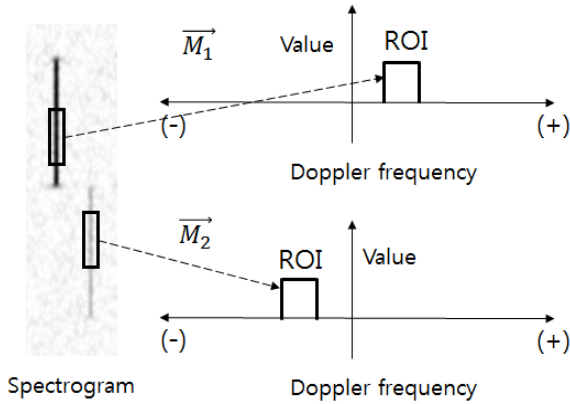


그림 7. 패턴 검출을 위한 마스크 벡터의 디자인
Fig. 7. Design of masking vector for pattern detection

본 논문에서는 그림 7과 같은 2개의 마스크 벡터를 사용하였다. 따라서 x_1 과 x_2 로부터 표 2와 같이 도플러 빈 패턴 검출기의 출력 값 P 를 결정한다.

표 2. 출력 표
Table 2. Output table

x_1	x_2	P
0	0	0
0	1	1
1	0	2
1	1	0

3.4 시간 패턴 검출기

도플러 빈 패턴 검출기의 출력을 수집함으로써 표 3과 같은 조건으로 스펙트로그램의 패턴을 검출할 수 있다.

표 3. 시간 패턴의 검출 조건
Table 3. Detection condition of time pattern

P_p	P_{p-1}	P_{p-2}	Y
'T'	'T'	'T'	1
others			0

표 3은 시간 방향에서 플래쉬를 강건하게 검출하기 위하여 동일한 P 가 3회 반복 될 때 플래쉬로 검출하도록 조건을 설정한 것이다. 여기서 P_p 는 시간 샘플 t_p 일 때, 출력된 P 이다. 양의 도플러 주파수에 있는 플래쉬를 사용할 경우에는 'T'=1을 대입하고 음의 도플러 주파수에 위치한 플래쉬를 사용할 경우에는 'T'=2를 대입한다. 그러나 이러한 단순한 조건에 의한 검출법은 몇 가지 문제를 내포하고 있다. 첫 째로 이전 시간 샘플에서의 P 를 내부 공간에 저장해야 한다. 이는 패턴 판독에 필요한 P 의 개수가 많아질수록 요구되는 버퍼와 계산량이 증가함을 의미한다. 두 번째 문제는 시간 영역에서의 패턴의 검출 조건이 가변적일 수 있다는 것이다. 이러한 문제를 해결하기 위하여 그림 8과 같은 유한상태기계(FSM, Finite State Machine)을 생각할 수 있다[11].

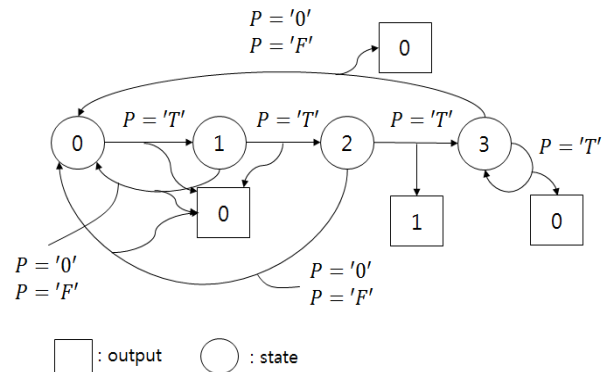


그림 8. 유한상태기계(FSM)
Fig. 8. Finite state machine

그림 8의 FSM은 입력이 P 가 들어올 때마다 상태의 변화와 출력을 갖는다. 이러한 FSM의 출력을 $FSM(P)$ 라고 정의하고 $Y=FSM(P)$ 와 같이 대입하여 시간 패턴 검출기의 출력으로 사용한다.

3.5 SVM을 이용한 FSM의 구현

FSM은 입력에 따른 상태 변화와 출력만 정확하게 보장하면 되기 때문에 디지털 회로, 컴퓨터 프로그래밍 등 다양한 방법으로 구현이 가능하다[11]. 본 논문에서는 서포트 벡터 머신(SVM, Support Vector Machine)을 사용하여 그림 8의 FSM을 구현하였다[12].

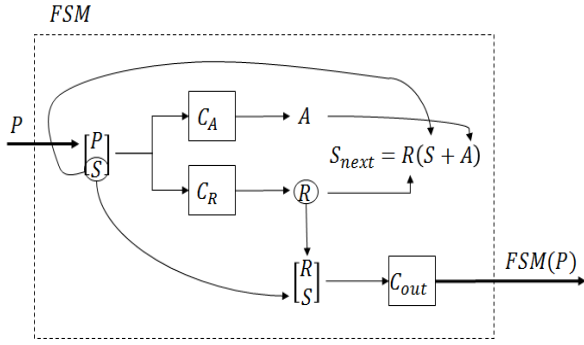


그림 9. SVM으로 구현된 FSM의 구조
Fig. 9. Structure of FSM implemented by SVM

그림 9는 SVM을 이용하여 구현한 FSM의 구조를 보여준다. 그림 9에서 C_A , C_R , C_{out} 은 SVM으로 구현된 이진분류기이고 $S \in \mathbb{Z}$ 는 상태변수이며 $A \in \{0,1\}$ 와 $R \in \{0,1\}$ 는 C_A 과 C_R 의 출력으로써 상태천이를 위한 변수이다.

입력 P 가 주어졌을 때, 입력 P 와 현재의 상태변수 S 를 사용하여 C_A , C_R 의 입력벡터를 $[PS]^T$ 의 형태로 정의한다. C_A 와 C_R 의 출력인 A , R 과 현재의 상태변수 S 를 사용하여 다음 세대의 상태변수 $S_{\neq xt}$ 를 $S_{\neq xt} = R(S+A)$ 와 같이 계산한다. 마지막으로 현재의 상태변수 S 와 C_R 의 출력인 R 를 사용하여 C_{out} 의 입력 벡터를 $[RS]^T$ 와 같이 정하고 C_{out} 의 출력을 해당 FSM의 최종 출력으로 사용한다. 이후 $S = S_{\neq xt}$ 와 같이 상태변수를 업데이트 한다.

그림 8의 FSM을 구현하기 위한 C_A , C_R , C_{out} 의 훈련 데이터는 표 4, 5, 6과 같다. 이와 같이 기계학습 기반의 이진분류기를 이용하여 FSM을 구현할 때의 장점은 C_A , C_R , C_{out} 의 훈련 데이터만을 변경하여 다양한 패턴의 검출이 가능한 FSM을 만들 수 있다는 점이다.

표 4. C_A 의 훈련 데이터

Table 4. Training data of C_A

S	P	A
0	'T'	1
1	'T'	1
2	'T'	1
3	'T'	0
Others		0

표 5. C_R 의 훈련 데이터

Table 5. Training data of C_R

S	P	R
0	'T'	1
1	'T'	1
2	'T'	1
3	'T'	1
Others		0

표 6. C_{out} 의 훈련 데이터

Table 6. Training data of C_{out}

S	R	FSM(P)
2	1	1
Others		0

IV. 시뮬레이션

4.1 합성데이터에 대한 시뮬레이션

본 논문에서 제안한 알고리즘의 성능을 검증하기 위하여 2장에서 소개한 풍력블레이드의 신호 모델을 사용하여 그림 10과 같이 다양한 속도가 변화하는 풍력 블레이드에 대한 스펙트로그램을 합성하였다.

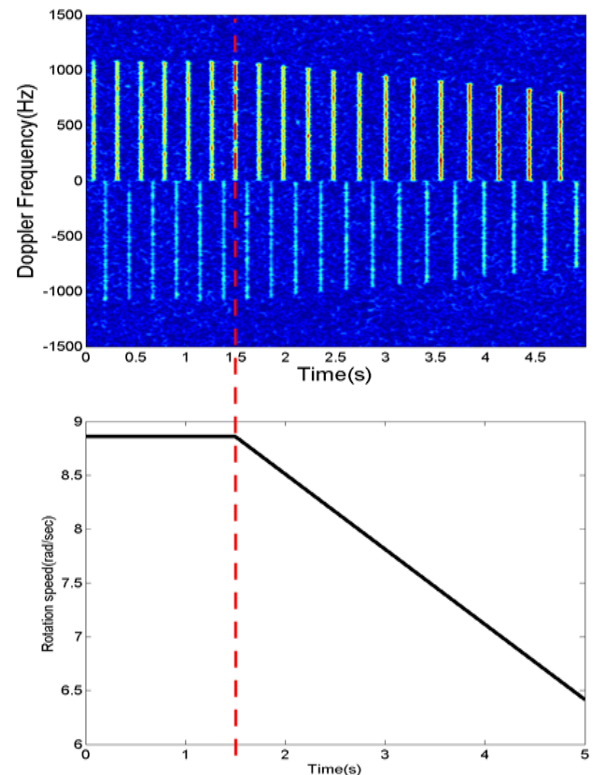


그림 10. 합성된 풍력 블레이드의 스펙트로그램
Fig. 10. Generated spectrogram of wind turbine blade

표 7. 그림 9의 매개변수

Table 7. Parameters in figure 9

Parameter	Value
L	1
N	3
R_0	1000 (m)
λ	0.0125(m)
sampling rate	3000 (sps)
β	40 (degree)
Ω_0	8.8593 (rad/sec)
z_0	80 (m)
ϕ_0	50 (degree)
$a(0 \leq t < 1.5)$	0 (rad/sec)
$a(1.5 \leq t < 5)$	-0.6981 (rad/sec)
σ_h	1
σ_l	0.5
observation time	5 (sec)
SNR	5 dB

그림 10의 스펙트로그램의 파라미터는 표 7과 같다. 그림 10의 스펙트로그램에서 풍력 블레이드는 1.5초까지 초기 속도를 유지하다가 1.5초 지점부터 감속을 시작한다. 감속이 시작되는 1.5초 지점부터는 플래쉬가 나타나는 최대도플러 주파수가 감소하면서 플래쉬 사이의 간격이 증가함을 확인할 수 있다.

또한 잡음 환경을 재현하기 위하여 가우시안 노이즈를 추가하여 5dB의 SNR의 신호를 만들어 주었다. 여기에 영 도플러 제거를 위한 제로 패딩 및 임계치를 적용하여 구한 이진화된 도플러 빈 \vec{D}_{bin} 를 수집하여 만든 영상은 그림 11과 같다. 이때 임계치를 위한 상수 α 는 0.999를 사용하였다.

그림 11에서 임계치가 수렴하는 0.8초 지점까지는 스펙트로그램에서 플래쉬가 정상적으로 식별되지 않는다. 따라서 이 시뮬레이션에서는 0.8초 이후에서만 알고리즘이 정상적으로 회전 속도를 추출할 수 있다.

그림 12는 그림 11로부터 0.8초 이후의 블레이드의 회전 속도를 추출한 결과이다. 표 8은 해당 시뮬레이션에 대한 SNR 별 RE(Relative Error)를 100회의 몬테카를로 시뮬레이션을 통해 계산한 결과이다. RE는 식 (13)과 같이 정의 된다.

$$RE = \frac{\|\vec{T} - \vec{Y}\|}{\|\vec{T}\|} \times 100 \quad (13)$$

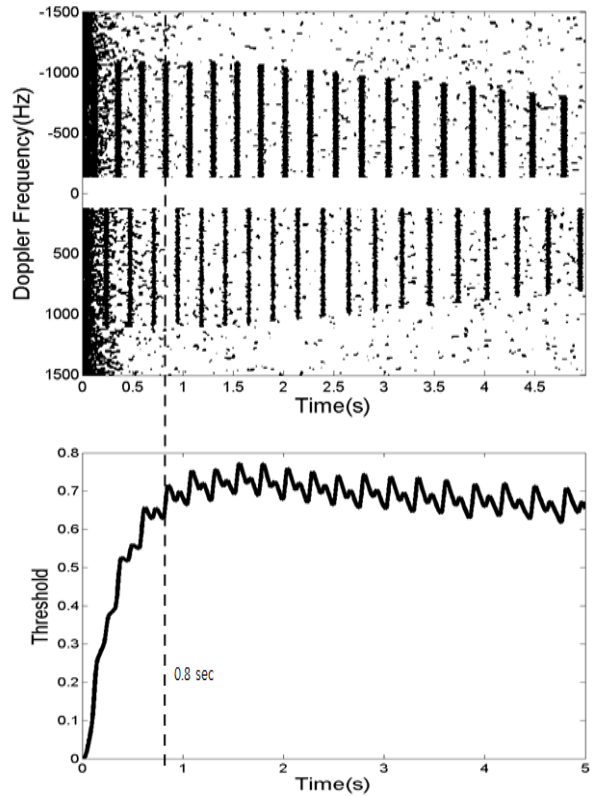


그림 11. 임계치가 적용된 스펙트로그램
Fig. 11. Spectrogram with threshold applied

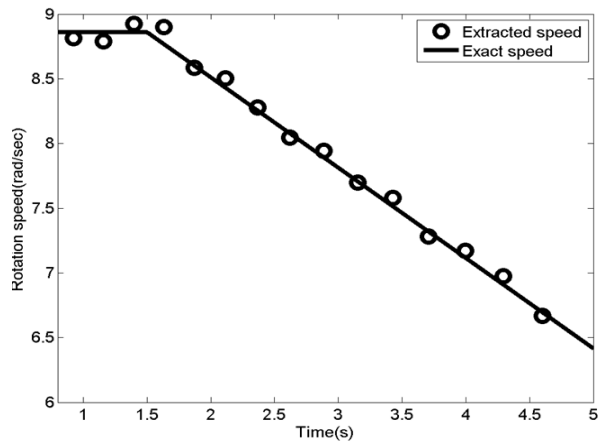


그림 12. 추출된 블레이드 회전 속도
Fig. 12. Extracted blade rotational speed

식 (13)에서 \vec{Y} 는 제안된 알고리즘으로 추출된 풍력 블레이드의 시간에 따른 회전 속도 값을 원소로 갖는 벡터이고 \vec{T} 는 회전 속도가 추출된 시점에서의 실제 속도 값을 원소로 갖는 벡터이다. $\|\cdot\|$ 는 2-norm이다. 표 8에서 SNR 5dB까지는 높은 정확도를 유지하다 0dB부터는 급격히 정확도가 낮아짐을 확인할 수 있다.

표 8. 시뮬레이션 결과의 Relative error
Table 8. Relative error of simulation result

SNR (dB)	RE (%)
15	0.5354
10	0.6076
5	0.6073
0	65.6468

4.2 수치해석 데이터에 대한 시뮬레이션

합성데이터에 대한 시뮬레이션을 통하여 본 논문에서 제안한 알고리즘이 블레이드의 회전 속도가 변화하는 상황에서도 속도를 추정할 수 있으며 5dB 이상의 잡음 환경에서도 잘 동작함을 확인하였다. 합성데이터는 블레이드의 형상을 직선으로 모델링한 것이므로 실제 풍력 블레이드의 형상에 의한 영향은 반영하지 못하는 한계가 있다. 따라서 표적의 형상의 영향을 반영한 수치해석 데이터에 의한 검증이 필요하다.

그림 13은 수치해석 시뮬레이션을 위해 작성한 풍력 블레이드의 3D CAD 모델이다. 블레이드의 길이는 1.05 m 이고 중앙에 3개의 블레이드를 연결하기 위한 허브가 위치한다. 표 9는 시뮬레이션을 위한 매개변수이다. 수치해석 방법은 물리광학법 (Physical optics)을 사용하였다.

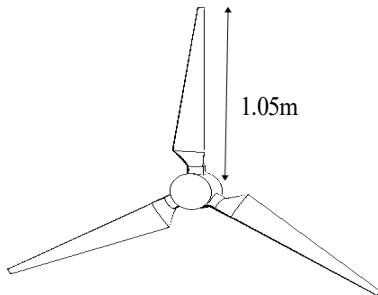


그림 13. 풍력 블레이드의 3D CAD 모델
Fig. 13. 3D CAD model of wind turbine blade

표 9. 수치해석 시뮬레이션을 위한 매개변수
Table 9. Parameters for numerical simulation

Parameter	Value
λ	0.0125(m)
sampling rate	3600 (sps)
β	0 (degree)
Ω	6.28 (rad/sec)
observation time	5 (sec)

그림 14와 그림 15는 각각 수치해석 시뮬레이션을 수행해서 얻은 SNR 30dB와 10dB에서의 스펙트로그램을 보여준다. 그림 14의 스펙트로그램에서는 제로 도플러 부근에서 합성데이터에서는 관찰할 수 없었던 패턴이 보인다. 이는 블레이드와 회전축을 연결하는 허브에 의해 발생하는 성분이다. 그리고 양의 도플러 주파수와 음의 도플러 주파수에서 나타나는 플래쉬의 진폭에 상당한 차이가 있어서 블레이드의 다가오는 면과 멀어지는 면의 RCS의 진폭이 다를 수 있다는 식 (6)의 예측을 다시 한 번 확인할 수 있다.

SNR이 낮아진 그림 15에서는 음의 도플러 주파수에서 플래쉬를 관측할 수 없음을 확인할 수 있다. 그림 16은 그림 14와 그림 15에 해당하는 데이터로부터 블레이드의 회전 속도를 추출한 결과이다.

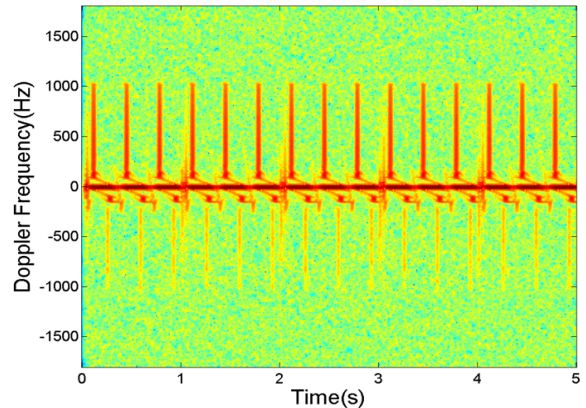


그림 14. 수치해석으로 계산한 스펙트로그램(SNR: 30dB)
Fig. 14. Spectrogram calculated by numerical simulation (SNR: 30dB)

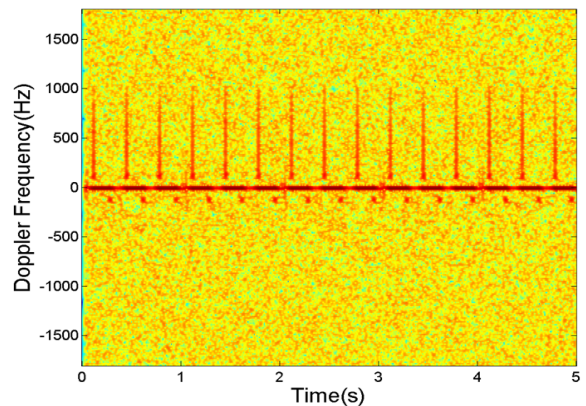


그림 15. 수치해석으로 계산한 스펙트로그램(SNR: 10dB)
Fig. 15. Spectrogram calculated by numerical simulation (SNR: 10dB)

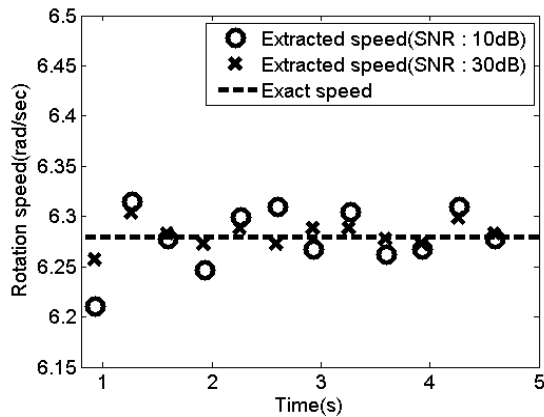


그림 16. 추출된 블레이드 회전 속도
Fig. 16. Extracted blade rotational speed

그림 16에서 제안된 방법에 의하여 추출된 회전 속도가 실제 회전 속도에 근사함을 확인할 수 있다. 또한 신호의 SNR에 비례하여 추출된 회전 속도의 정확성이 증가함을 확인할 수 있다.

V. 결 론

본 논문에서는 풍력 블레이드의 스펙트로그램에서 나타나는 가장 강한 특성 중 하나인 플래쉬의 간격으로부터 풍력 블레이드의 회전 속도를 재귀적인 절차로 추출하는 알고리즘을 제안하였다.

본 논문에서 제안한 방법은 계산에 데이터를 취득하는 시점에 매번 스펙트로그램을 생성하는 대신 현재 취득된 데이터에 대한 도플러 빈만을 생성하고 이것을 몇 가지 패턴으로 분류한다. 이후 시간에 따른 패턴의 변화를 유한상태 기계를 이용하여 상태변수의 형태로 관리함으로써 전체 스펙트로그램을 생성하지 않고도 시간 방향의 패턴으로부터 풍력 블레이드의 회전 속도를 검출할 수 있다. 이러한 재귀적인 절차를 사용함으로써 계산에 필요한 연산량과 메모리 소모량을 크게 줄일 수 있었다. 또한 제안된 방법을 검증하기 위하여 합성데이터와 수치 해석 데이터에 대하여 시뮬레이션을 수행하였다.

시뮬레이션 결과 각각 5dB 및 10dB 이상의 SNR 환경에서 풍력블레이드의 회전 속도를 추출할 수 있음을 확인하였다. 이러한 결과는 제안된 방법이 잡음에 강건하게 동작함을 보여준다. 이러한 잡음 강건성은 제안된 방법이 사용하는 특성인 플래쉬가 잡음에 매우 강건하기 때문으로 판단된다.

References

- [1] H. Polinder, J. A. Ferreira, B. B. Jensen, A. B. Abrahamsen, K. Atallah, and R. A. McMahon, "Trends in Wind Turbine Generator Systems", in IEEE Journal of Emerging and Selected Topics in Power Electronics, Vol. 1, No. 3, pp. 174-185, Sept. 2013.
- [2] S. J. Park, D. S. Kang, "Implementation of Real-time Monitoring System using the Neural Network for Automatic Failure Diagnosis of Offshore Wind Turbine", The Journal of Korean Institute of Information Technology, Vol. 10, No. 7, pp. 193-198, Jul. 2012.
- [3] B. Jokanovic, M. Amin, and F. Ahmad, "Radar fall motion detection using deep learning", 2016 IEEE Radar Conference (RadarConf), Philadelphia, PA, pp. 1-6, May 2016.
- [4] Y. Kim and T. Moon, "Human Detection and Activity Classification Based on Micro-Doppler Signatures Using Deep Convolutional Neural Networks", in IEEE Geoscience and Remote Sensing Letters, Vol. 13, No. 1, pp. 8-12, Jan. 2016.
- [5] H. Du, T. Jin, Y. Song, and Y. Dai, "Deep Activity: a micro-Doppler spectrogram-based net for human behaviour recognition in bio-radar," in The Journal of Engineering, Vol. 2019, No. 19, pp. 6147-6151, Oct. 2019.
- [6] Y. J. Choi and I. S. Choi, "Dynamic RCS Analysis by Damage Status of Wind Turbine Blade", The Journal of Korean Institute of Information Technology, Vol. 17, No. 7, pp. 193-198, Jul. 2019.
- [7] V. C. Chen, "The Micro-Doppler Effect in Radar, Boston", MA, Artech House, pp. 93-143, 2011.
- [8] M. Kim, K. B. Kang, S. H. Park, and J. H. Jung, "Measurement and Analysis of Radar Shielding and Doppler Effects by the Wind Farm", The Journal of Korean Institute of

Electromagnetic Engineering and Science, Vol. 28, No. 8, pp. 742-749, Aug. 2016.

- [9] K. B. Kang and I. O. Choi, et al., "Analysis of Micro-Motion Characteristics Caused by Maneuvering a Small Unmanned Aerial Vehicle Acquired from X-Band Radar", The Journal of Korean Institute of Electromagnetic Engineering and Science, Vol. 30, No. 7, pp. 573-582, Jul. 2019.
- [10] F. J. Caro-Lopera, V. Leiva, and N. Balakrishnan, "Connection between the Hadamard and matrix products with an application to matrix-variate Birnbaum-Saunders distributions", Journal of Multivariate Analysis, Vol 104, No. 1, pp. 126-139, Jul. 2012.
- [11] H. Lin and D. G. Messerschmitt, "Finite state machine has unlimited concurrency", in IEEE Transactions on Circuits and Systems, Vol. 38, No. 5, pp. 465-475, May 1991.
- [12] J. A. K. Suykens and J. Vandewalle, "Least Squares Support Vector Machine Classifiers", Neural Processing Letters, Vol. 9, No. 3, pp. 293-300, Jun. 1999.

최 인 식 (In-Sik Choi)



1998년 2월 : 경북대학교
전자공학과(공학사)
2000년 2월 : 포항공과대학교
전자전기공학과(공학석사)
2003년 2월 : 포항공과대학교
전자전기공학과(공학박사)
2003년 ~ 2004 : LG전자

선임연구원

2004년 ~ 2007년 : 국방과학연구소 선임연구원

2007년 ~ 현재 : 한남대학교 전기전자공학과 교수

관심분야 : 레이더 신호처리, RCS 해석 및 분석, 레이더 시스템 설계

저자소개

최 영 재 (Young-Jae Choi)



2013년 2월 : 한남대학교

전자공학과(공학사)

2018년 2월 : 한남대학교

전자공학과(공학석사)

2018년 3월 ~ 현재 : 한남대학교

전기전자공학과 박사과정

관심분야 : RADAR 신호처리, RCS

해석 및 분석, 특성 벡터 추출