JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

VOICE-ORIENTED ANTENNA AUTO-ALIGNMENT SYSTEM USING LORA RSSI FEEDBACK

An Automatic Signal-Reception Optimisation Approach with Voice Triggering and Real-Time Monitoring

[1]Praneeth Niranjan, [2]Thrishar R G, [3]Vijeth Shanubaug, [4]Satish Kumar B

[1,2,3]UG Student, [4]Assistant Professor

[1,2,3,4]Department of Electronics and Communication Engineering,

[1,2,3,4]K.S. Institute of Technology, Bengaluru, India

Abstract : Directional antennas give a large improvement in received signal strength, but only when they point at the transmitter. Manual alignment is slow, needs a skilled operator and must be repeated whenever the link geometry changes. This paper presents a Voice-Oriented Antenna Auto-Alignment System that points a directional receiving antenna automatically using the Received Signal Strength Indicator (RSSI) and Signal-to-Noise Ratio (SNR) reported by a 433 MHz LoRa SX1278 transceiver. An Arduino microcontroller drives a servo-actuated antenna mount, performs a coarse 10° scan of the azimuth range, refines the search around the strongest bearing and estimates the true peak with a least-squares quadratic fit of the averaged RSSI samples. Alignment is started by a voice command routed through the Blynk IoT platform, after which the process is fully automatic, with no manual positioning mode. A Flask WebSocket dashboard displays live RSSI, SNR and antenna angle. Monte Carlo simulation over 1000 random transmitter bearings shows that, with 2 dB RSSI noise and five-packet averaging, the proposed method reaches a mean pointing error of 1.65° using about 25 antenna positions (≈16 s), compared with 5.25° and 181 positions (≈118 s) for an exhaustive 1° sweep. The system recovers an average of about 10.9 dB lost to random initial orientation.

IndexTerms - Antenna alignment, LoRa, SX1278, RSSI, servo control, voice control, Blynk, IoT, Arduino.

I. INTRODUCTION

Wireless links that use directional antennas, such as point-to-point telemetry, rural IoT backhaul, amateur radio and ground stations, depend heavily on correct antenna orientation. A directional antenna with a half-power beamwidth (HPBW) of 50° loses about 3 dB at 25° of misalignment and more than 10 dB at 45°. In practice the antenna is usually aligned by hand: an operator rotates the mount while watching a signal meter. This is slow, depends on the operator's skill and must be repeated whenever the transmitter moves, the mount is disturbed by wind, or the installation is relocated.

Long Range (LoRa) transceivers provide a convenient way to automate this task. Every received LoRa packet carries an RSSI and SNR measurement made by the radio itself [1], [7], so the receiver can measure link quality without any extra RF hardware. When these measurements are combined with a low-cost servo mount and a microcontroller, the antenna can search for the direction of maximum received power on its own.

This work develops such a system and adds two features aimed at ease of use. First, alignment is triggered by voice through the Blynk IoT platform, so the user does not need physical access to the antenna. Second, a web dashboard built on Flask and WebSockets shows the received signal and antenna position in real time. Once triggered, the alignment runs fully automatically; the user never positions the antenna manually.

The main contributions of the paper are: (i) a complete hardware and software architecture for voice-triggered, RSSI-driven antenna alignment using an Arduino and an SX1278 LoRa module; (ii) a three-stage search algorithm (coarse scan, local refinement and quadratic peak estimation) that is robust to RSSI noise; and (iii) a simulation study comparing the proposed search with an exhaustive sweep in terms of pointing accuracy and alignment time.

II. LITERATURE SURVEY

LoRa is a chirp spread-spectrum modulation designed for long-range, low-power links in the sub-GHz ISM bands. Augustin et al. [1] describe the LoRa physical layer and show how the spreading factor trades data rate for sensitivity, while Centenaro et al. [2] and Adelantado et al. [9] discuss the capabilities and limits of LoRa and LoRaWAN for IoT and smart-city deployments. Petäjäjärvi et al. [3] measured LoRa coverage in urban and open environments and proposed a log-distance channel attenuation model, which motivates the path-loss model used in this study. Bor et al. [4] examined how LoRa networks behave as the number

of nodes grows and highlighted the importance of link quality for reliable reception.

The theory of antenna radiation patterns, gain and beamwidth is covered in detail by Balanis [5], and the effect of path loss and shadowing on the received signal is treated by Rappaport [6]. Classical tracking systems in radar and satellite ground stations use monopulse or conical-scan techniques to keep an antenna pointed at a target [12]. These methods are accurate but require multiple feeds or dedicated RF comparators, which makes them too costly and complex for low-cost IoT links. GPS-based trackers avoid RF hardware but need the position of both ends to be known and shared.

RSSI-based alignment needs only a single receiver and therefore suits low-cost systems, but RSSI readings are noisy because of multipath, shadowing and receiver quantisation. Simple "move to the largest reading" strategies are easily misled by a single noisy sample. This work addresses that weakness by averaging several packets at each position and estimating the peak from a fitted curve rather than from a single maximum reading.

III. RESEARCH METHODOLOGY

The methodology covers the system architecture, the hardware design, the signal model, the alignment algorithm and the voice-control and monitoring software. The details are as follows.

3.1 System Architecture

The system has two LoRa nodes. The *transmitter node* is placed at the remote location and sends short beacon packets at a fixed interval. The *receiver node* carries the directional antenna on a servo-driven pan–tilt mount. Its Arduino reads the RSSI and SNR of every beacon from the SX1278 module and moves the servos to search for the bearing of maximum signal. The Arduino is connected to a host computer over USB serial; a Python gateway on the host links the Arduino to the Blynk cloud (for voice commands) and to the Flask WebSocket server (for the dashboard). Fig. 1 shows the overall architecture.

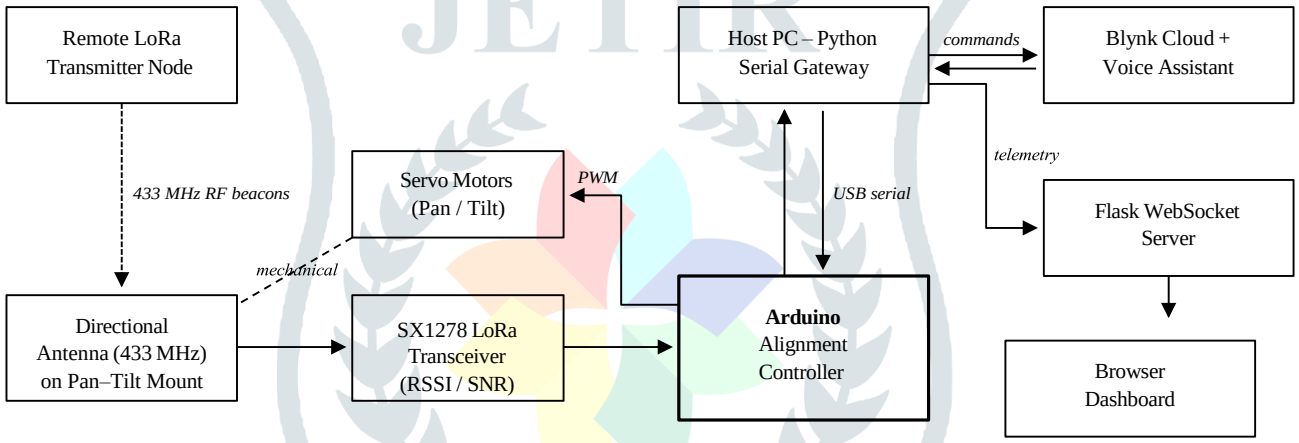

Fig. 1: Block diagram of the voice-oriented antenna auto-alignment system

3.2 Hardware Design

The hardware components and their roles are listed in Table 1. The SX1278 module communicates with the Arduino over SPI (NSS on D10, MOSI on D11, MISO on D12, SCK on D13, RESET on D9 and DIO0 interrupt on D2). Because the SX1278 is a 3.3 V device, it is powered from the 3.3 V rail and its inputs are driven through level shifting. The pan and tilt servos receive PWM signals from two PWM-capable pins and are powered from a separate 5 V supply that shares a common ground with the Arduino, so that servo current surges do not reset the microcontroller or disturb the radio.

Table 1: Hardware components of the receiver node

Component	Specification	Function
Microcontroller	Arduino (ATmega328P, 16 MHz)	Runs the alignment algorithm and servo control
LoRa transceiver	SX1278, 433 MHz, SPI interface	Receives beacons; reports packet RSSI and SNR
Actuators	Servo motors, 0–180°, PWM	Pan (azimuth) and tilt (elevation) positioning
Antenna	Directional, 433 MHz	Provides gain toward the transmitter
Power supply	5 V regulated (servos), 3.3 V (radio)	Isolates servo load from logic and RF supply
Host interface	USB serial, 115200 baud	Link to Python gateway, Blynk and dashboard

3.3 Signal Model

The received power at servo angle θ, when the transmitter lies at bearing θ_T, is modelled by the link budget

$$P_r(\theta) = P_t + G_t + G_r(\theta - \theta_T) - PL(d) + X_\sigma \tag{3.1}$$

where P_t is the transmit power, G_t and G_r are the transmit and receive antenna gains in dBi, $PL(d)$ is the path loss at distance d, and X_σ is a zero-mean Gaussian term of standard deviation σ that represents shadowing, multipath and RSSI measurement noise [6]. Following the log-distance model used for LoRa coverage studies [3], [6],

$$PL(d) = 20 \log_{10}(4\pi d_0/\lambda) + 10\, n \log_{10}(d/d_0) \tag{3.2}$$

where λ is the wavelength, $d_0 = 1$ m is the reference distance and n is the path-loss exponent. The directional receive antenna is modelled with a $\cos^m$ power pattern [5],

$$G_r(\Delta) = G_0 + \max\{10 \log_{10}(\cos^m \Delta), -20\}, \quad m = \ln 0.5 / \ln \cos(HPBW/2) \tag{3.3}$$

where Δ is the pointing error, G_0 is the peak gain and the −20 dB floor represents side and back lobes. To reduce the effect of noise, the controller averages the RSSI of N packets at each antenna position:

$$\bar{R}(\theta) = (1/N) \sum_{k=1}^{N} RSSI_k(\theta) \tag{3.4}$$

3.4 Alignment Algorithm

The alignment algorithm, shown in Fig. 2, has three stages.

Stage 1 – Coarse scan: The pan servo steps from 0° to 180° in 10° increments (19 positions). At each position the servo is allowed to settle and N beacons are averaged using Eq. 3.4. The bearing θ_0 with the largest averaged RSSI is taken as the coarse estimate.

Stage 2 – Local refinement: The antenna is moved to the intermediate 5° positions within ±30° of θ_0 that were not visited in Stage 1, which adds about six positions.

Stage 3 – Peak estimation: All samples within ±30° of θ_0 are fitted with a second-order polynomial by least squares,

$$\bar{R}(\theta) \approx c_2\theta^2 + c_1\theta + c_0, \quad \theta^* = -c_1 / (2c_2) \quad (c_2 < 0) \tag{3.5}$$

and the servo is commanded to the rounded vertex θ^*. If the fitted curve is not concave ($c_2 \geq 0$), the controller falls back to θ_0. The fit uses many samples on both sides of the main lobe, so a single noisy reading cannot move the final estimate far. The same procedure is then applied to the tilt servo over its range.

After alignment the controller enters a monitoring state and keeps tracking the averaged RSSI. If the signal drops by more than 3 dB below the value recorded at alignment, it automatically starts a new search. The total alignment time for K visited positions is

$$T = K\,(t_s + N\, t_p) \tag{3.6}$$

where t_s is the servo settling time and t_p is the beacon interval.

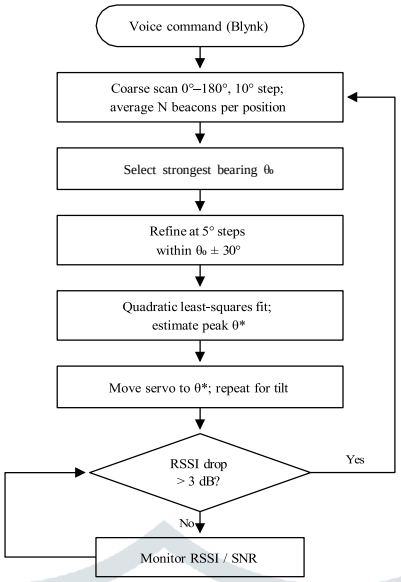

Fig. 2: Flowchart of the automatic alignment algorithm

3.5 Voice Control and Monitoring Dashboard

Voice commands are handled through the Blynk IoT platform [10]. A phrase such as "align antenna" spoken to a voice assistant linked with Blynk sets a virtual pin in the Blynk cloud. The Python gateway on the host computer watches this pin and sends an *ALIGN* command to the Arduino over serial. The voice command only starts the process; scanning, peak estimation and re-alignment are carried out entirely by the controller, and the system offers no manual positioning mode.

During operation the Arduino streams the current pan and tilt angles, RSSI, SNR and algorithm state to the gateway as comma-separated serial frames. The gateway forwards these frames to a Flask server, which pushes them to connected browsers over WebSockets [11]. The dashboard plots RSSI and SNR against time, shows the antenna bearing on a polar indicator and displays the current stage (scanning, refining, aligned or monitoring). This lets the user confirm the link quality remotely without opening the serial monitor.

3.6 Simulation Setup

Before the hardware tests, the algorithm was validated by Monte Carlo simulation in Python using the model of Section 3.3. The parameters are listed in Table 2. For each case, 1000 transmitter bearings were drawn uniformly between 10° and 170°. The proposed method was compared with an exhaustive sweep that visits every 1° position from 0° to 180° and selects the largest averaged reading.

Table 2: Simulation parameters

Parameter	Value	Parameter	Value
Carrier frequency	433 MHz	Path-loss exponent, n	2.7
Transmit power, P_t	17 dBm	Link distance, d	500 m
Transmit antenna gain, G_t	2.15 dBi	RSSI noise, σ	1, 2, 4 dB
Receive peak gain, G_0	7 dBi	Packets averaged, N	1, 5
Receive antenna HPBW	50°	Servo settling time, t_s	150 ms
Side/back-lobe floor	−20 dB	Beacon interval, t_p	100 ms

IV. RESULTS AND DISCUSSION

4.1 RSSI Profile and Alignment Example

Fig. 3 shows a single simulated alignment with the transmitter at 112°, σ = 2 dB and N = 5. The coarse-scan samples follow the main lobe of the antenna pattern but scatter by about ±1 dB around it. The highest single sample does not coincide with the true bearing, yet the quadratic fit over the main lobe places the antenna at 110°, within 2° of the transmitter. At 500 m the aligned RSSI is about −72 dBm, while an antenna pointing 90° away receives about −92 dBm, a difference of 20 dB.

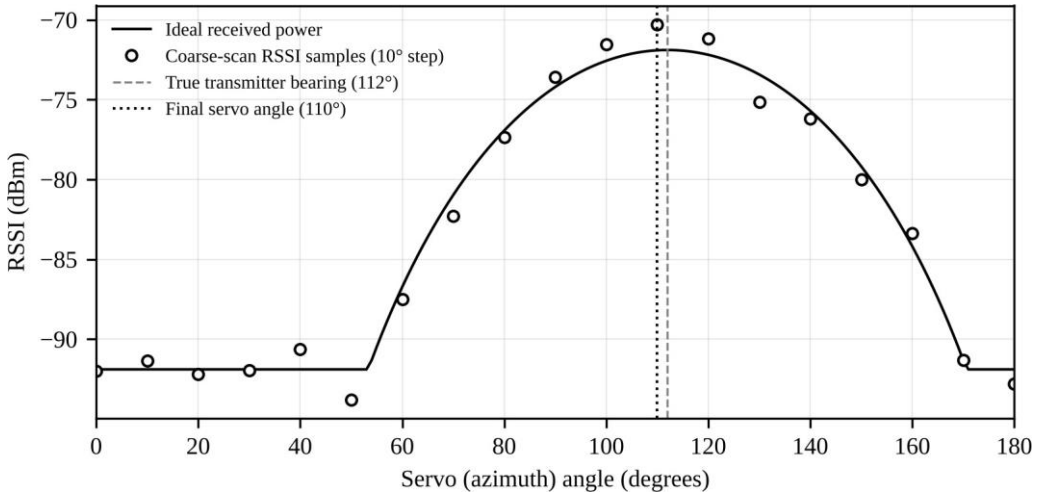

Fig. 3: Simulated RSSI versus servo angle for a transmitter at 112° (σ = 2 dB, N = 5)

4.2 Pointing Accuracy and Alignment Time

Table 3 summarises the Monte Carlo results. In every case the proposed method gives a lower mean absolute pointing error than the exhaustive 1° sweep while visiting only about 25 positions instead of 181, which shortens alignment by a factor of about seven. The exhaustive sweep performs worse despite its finer step because it takes the single largest reading out of 181 noisy samples; near the flat top of the main lobe, noise rather than antenna gain decides which sample is largest. The quadratic fit uses the shape of the whole main lobe and therefore averages the noise out.

Averaging five packets per position reduces the mean error of the proposed method from 3.90° to 1.65° at σ = 2 dB, at the cost of increasing alignment time from about 6 s to 16 s. Even at σ = 4 dB, which represents a strongly multipath environment, the mean error with averaging stays below 4°. Repeating the simulation at a link distance of 2 km (aligned RSSI ≈ −88 dBm, still well above the SX1278 sensitivity [7]) gave a mean error of 1.69°, showing that accuracy depends on the RSSI noise rather than on the absolute signal level.

Table 3: Monte Carlo comparison of alignment methods (1000 trials each, d = 500 m)

σ (dB)	N	Proposed (coarse + refine + fit)				Exhaustive 1° sweep			
		Mean error (°)	95th pct. (°)	Positions	Time (s)	Mean error (°)	95th pct. (°)	Positions	Time (s)
1	1	1.93	5.17	24.7	6.2	5.46	13.45	181	45.2
1	5	0.75	1.95	24.7	16.1	3.78	9.13	181	117.7
2	1	3.90	10.61	24.7	6.2	7.22	17.96	181	45.2
2	5	1.65	4.16	24.7	16.1	5.25	12.55	181	117.7
4	1	8.22	21.75	24.6	6.2	10.24	24.31	181	45.2
4	5	3.70	10.53	24.7	16.1	7.02	16.94	181	117.7

Fig. 4 shows the cumulative distribution of the pointing error for the proposed method with N = 5. At σ = 1 dB almost all trials end within 2° of the transmitter, and at σ = 2 dB about 95 % end within about 4°.

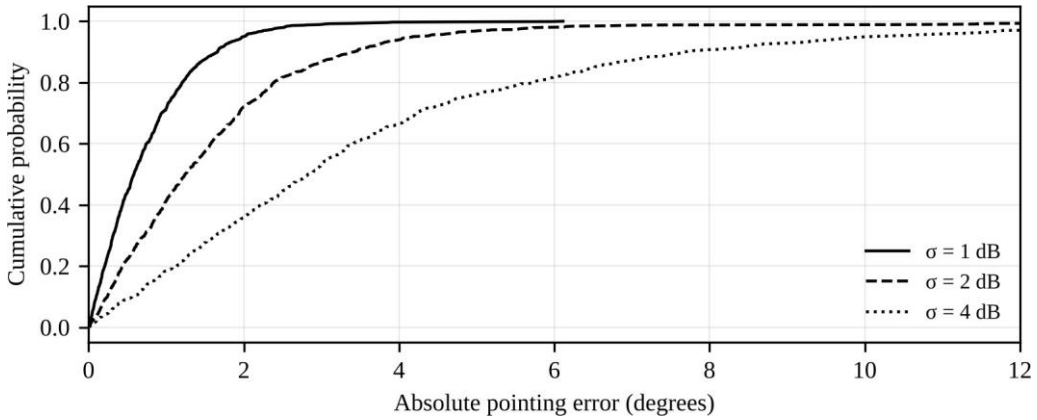

Fig. 4: Cumulative distribution of pointing error for the proposed method (N = 5)

4.3 Effect of Pointing Error on Received Signal

Table 4 converts pointing error into signal loss for the 50° beamwidth antenna. Errors below 5° cost less than 0.2 dB, so the accuracy of the proposed method is more than sufficient for practical links. In contrast, a randomly oriented antenna loses about 10.9 dB on average compared with perfect alignment. The automatic alignment therefore recovers roughly 11 dB of link margin,

which for a LoRa link can be used to extend range, lower the spreading factor for higher data rate, or reduce transmit power.

Table 4: Received-signal loss versus pointing error (HPBW = 50°)

Pointing error (°)	0	2	5	10	25	45	90
Signal loss (dB)	0.00	0.02	0.12	0.47	3.01	10.61	20.00

4.4 Hardware Prototype

A working prototype of the receiver node was assembled with the Arduino, the SX1278 module and the servo-driven mount, and the scan–refine–fit routine was verified to run on the microcontroller in real time. The quadratic fit needs only running sums of θ, θZ, $\theta 3$, θ^4, $\bar{R}$, $\theta\bar{R}$ and $\theta Z\bar{R}$, followed by the solution of a 3×3 linear system, so it fits easily within the memory and processing capacity of the ATmega328P. Powering the servos from a separate supply was found to be necessary to keep the radio and microcontroller stable during servo movement. In Phase 2 of the project, live serial and Blynk data replace the simulated RSSI so that field measurements can be compared directly with the simulation results presented above.

V. CONCLUSION AND FUTURE SCOPE

This paper presented a voice-oriented antenna auto-alignment system that uses LoRa RSSI feedback to point a directional antenna automatically. A voice command through Blynk starts the process, after which the Arduino controller performs a coarse scan, a local refinement and a quadratic peak estimate without any manual intervention, and a Flask WebSocket dashboard reports the link in real time. Simulation showed that, with 2 dB RSSI noise and five-packet averaging, the system aligns within 1.65° on average in about 16 s, which is more accurate and about seven times faster than an exhaustive 1° sweep, and it recovers about 11 dB of link margin compared with a randomly oriented antenna.

Future work includes field measurements over longer links, joint pan–tilt search instead of sequential axes, use of SNR together with RSSI for weak signals near the sensitivity limit, continuous tracking of moving transmitters, and replacing the host computer with a Wi-Fi-enabled microcontroller so that the receiver node can connect to Blynk directly.

ACKNOWLEDGMENT

The authors thank Mr. Satish Kumar B, Department of Electronics and Communication Engineering, K.S. Institute of Technology, Bengaluru, for his guidance throughout this project, and the Head of the Department and the Principal of K.S. Institute of Technology for providing the facilities required for this work.

REFERENCES

[1] Augustin, A., Yi, J., Clausen, T. and Townsley, W. M. 2016. A Study of LoRa: Long Range & Low Power Networks for the Internet of Things. Sensors, 16(9): 1466.

[2] Centenaro, M., Vangelista, L., Zanella, A. and Zorzi, M. 2016. Long-Range Communications in Unlicensed Bands: The Rising Stars in the IoT and Smart City Scenarios. IEEE Wireless Communications, 23(5): 60–67.

[3] Petäjäjärvi, J., Mikhaylov, K., Roivainen, A., Hänninen, T. and Pettissalo, M. 2015. On the Coverage of LPWANs: Range Evaluation and Channel Attenuation Model for LoRa Technology. Proc. 14th International Conference on ITS Telecommunications (ITST), Copenhagen, 55–59.

[4] Bor, M. C., Roedig, U., Voigt, T. and Alonso, J. M. 2016. Do LoRa Low-Power Wide-Area Networks Scale? Proc. 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), 59–67.

[5] Balanis, C. A. 2016. Antenna Theory: Analysis and Design, 4th ed. John Wiley & Sons, Hoboken, NJ.

[6] Rappaport, T. S. 2002. Wireless Communications: Principles and Practice, 2nd ed. Prentice Hall, Upper Saddle River, NJ.

[7] Semtech Corporation. SX1276/77/78/79 – 137 MHz to 1020 MHz Low Power Long Range Transceiver, Datasheet.

[8] Microchip Technology Inc. ATmega328P 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash, Datasheet.

[9] Adelantado, F., Vilajosana, X., Tuset-Peiro, P., Martinez, B., Melia-Segui, J. and Watteyne, T. 2017. Understanding the Limits of LoRaWAN. IEEE Communications Magazine, 55(9): 34–40.

[10] Blynk Inc. Blynk IoT Platform Documentation. https://docs.blynk.io

[11] Grinberg, M. 2018. Flask Web Development: Developing Web Applications with Python, 2nd ed. O'Reilly Media, Sebastopol, CA.

[12] Skolnik, M. I. 2001. Introduction to Radar Systems, 3rd ed. McGraw-Hill, New York.