TASK 2/5/2025

Tuesday, February 11, 2025 9:33 PM

```python
# Generates Gaussian noise with proper variance
noise_variance = power / 10                              # Adjusting the noise variance
noise = np.random.normal(0, np.sqrt(noise_variance), num_bits)  # Proper scaling of noise
```

Understanding Noise Variance and Its Role in Signal-to-Noise Ratio (SNR)

The code snippet you provided deals with **Gaussian noise generation**, an essential aspect of real-world communication systems. Let's break it down **step by step** and explain **why** you are doing this, how it affects the **signal-to-noise ratio (SNR)**, and its real-world applications.

---

1. What Does This Code Do?

python

```python
noise_variance = power / 10  # Adjusting the noise variance
noise = np.random.normal(0, np.sqrt(noise_variance), num_bits)  # Proper scaling of noise
```

- **Step 1:** Defines the **variance of the noise** as:

$$\sigma^2 = \frac{\text{power}}{10}$$

  - Since `power = 10^{-7}`, the noise variance is:

$$\sigma^2 = 10^{-7}/10 = 10^{-8}$$

  - This means the **standard deviation** of the noise (σ) is:

$$\sigma = \sqrt{10^{-8}} = 10^{-4}$$

- **Step 2:** Generates **Gaussian (random) noise** with:
  - **Mean** = **0** (centered around zero, which is common in thermal noise models).
  - **Standard deviation** = $\sqrt{\text{noise variance}}$ to ensure correct power scaling.
  - **Size** = `num_bits` **(1000 values)**, meaning **each transmitted bit gets affected by independent noise.**

2. Why Are You Doing This?

A. Modeling Real-World Noise (Additive White Gaussian Noise - AWGN)

- In **real-world communication channels**, signals experience **random fluctuations** due to:
  - Thermal noise from circuit components.
  - Atmospheric interference in wireless systems.
  - Cosmic background noise in deep-space communication.
- The most common model for this noise is **Additive White Gaussian Noise (AWGN)**:

$$y = x + n$$

  where:

  - x = transmitted signal (e.g., 0 or $\sqrt{10^{-7}}$).
  - n = Gaussian noise with mean 0 and variance σ^2.
  - y = received signal after noise addition.

By defining `noise_variance = power / 10`, you are **controlling the intensity of noise** in your simulation.

B. Impact on Signal-to-Noise Ratio (SNR)

The **Signal-to-Noise Ratio (SNR)** is a key performance measure in communication systems, defined as:

$$SNR = \frac{\text{Signal Power}}{\text{Noise Power}}$$

- **Signal Power** is given as `power = 10^{-7}`.
- **Noise Power** is set to `noise_variance = power / 10 = 10^{-8}`.
- Thus, the **SNR (linear scale)** is:

$$SNR = \frac{10^{-7}}{10^{-8}} = 10$$

- In **decibels (dB)**, SNR is computed as:

$$SNR_{dB} = 10\log_{10}(SNR)$$

$$SNR_{dB} = 10\log_{10}(10) = 10\text{ dB}$$

This means that your system is designed with an **SNR of 10 dB**, which is a reasonable level for moderate-quality digital communication.

Does `noise_variance` Control SNR?

Yes! If you **increase** `noise_variance`, the **noise power increases**, reducing the SNR (worse performance). If you **decrease** `noise_variance`, the noise power decreases, improving the SNR.

By choosing `power / 10`, you are setting **a controlled SNR level** for your system.

---

3. How Does This Apply to Real Life?

Your noise model is a **fundamental** part of any real communication system.

A. Wireless Communication (Wi-Fi, Bluetooth, 5G)

- **Higher noise variance** (low SNR) leads to more **errors**, requiring **stronger error correction codes.**
- **Example:** A Wi-Fi signal in a **noisy environment (many devices)** has higher noise variance, meaning lower SNR and possible **packet loss.**

B. Space Communication (NASA, Satellites, Deep Space Network)

- Space probes like **Voyager, Mars Rovers, and Starlink satellites** must deal with **extreme noise.**
- Engineers **adjust SNR** using:
  - **Power control** (increase transmitter power).
  - **Error correction techniques** (e.g., Reed-Solomon codes).
  - **Antenna design** (higher gain antennas increase signal strength).

C. Audio & Video Transmission

- **Digital TV and radio signals** have optimized SNR to **reduce static and distortion.**
- If noise variance is **too high**, your **TV signal might pixelate** or **audio might cut out.**

D. Radar & Military Systems

- **Stealth aircraft (F-35, B-2 bomber)** use **low radar SNR** to remain undetectable.
- Military communication is optimized to **maximize SNR**, reducing enemy jamming effects.

1. Effect of Changing `noise_variance = power / X`

Current Case (`X = 10`):

```python
noise_variance = power / 10
```

- **Signal Power** = 10^{-7}
- **Noise Power** = 10^{-8}
- **SNR (Linear Scale):**

$$SNR = \frac{10^{-7}}{10^{-8}} = 10$$

- **SNR (dB Scale):**

$$SNR_{dB} = 10\log_{10}(10) = 10 \text{ dB}$$

If You Change `X = 20`:

```python
noise_variance = power / 20
```

- **Signal Power** remains 10^{-7}.
- **Noise Power decreases** to:

$$\frac{10^{-7}}{20} = 5 \times 10^{-9}$$

- **SNR (Linear Scale):**

$$SNR = \frac{10^{-7}}{5 \times 10^{-9}} = 20$$

- **SNR (dB Scale):**

$$SNR_{dB} = 10\log_{10}(20) \approx 13.01 \text{ dB}$$

So, by increasing the divisor from **10 to 20**, you are **reducing the noise power**, which leads to a **higher SNR (from 10 dB to 13 dB)** and makes the **signal easier to detect accurately.**

2. What Happens if You Use a Larger X (e.g., 50, 100)?

X Value	Noise Variance	SNR (Linear)	SNR (dB)
10	10^{-8}	10	10 dB
20	5×10^{-9}	20	13 dB
50	2×10^{-9}	50	17 dB
100	10^{-9}	100	20 dB

As `X` increases, **noise power decreases, SNR increases, and fewer errors occur.**

2. What Happens if You Use a Larger X (e.g., 50, 100)?

X Value	Noise Variance	SNR (Linear)	SNR (dB)
10	10^{-8}	10	10 dB
20	5×10^{-9}	20	13 dB
50	2×10^{-9}	50	17 dB
100	10^{-9}	100	20 dB

As `X` increases, **noise power decreases, SNR increases, and fewer errors occur.**

3. What Happens if You Make X Smaller?

Example: Setting `X = 5`

```python
noise_variance = power / 5
```

- **Noise variance increases** to:

$$\frac{10^{-7}}{5} = 2 \times 10^{-8}$$

- **SNR decreases** to:

$$SNR = \frac{10^{-7}}{2 \times 10^{-8}} = 5$$

- **SNR (dB Scale):**

$$SNR_{dB} = 10\log_{10}(5) \approx 7dB$$

Now, your signal is **harder to distinguish** because the noise is stronger, leading to **more bit errors.**