PCET's Pimpri Chinchwad College of Engineering

PCCOE

"Knowlege Brings Freedom"

Progress Credibility Confidence
Optimism Excellence

Since 1999

FA-2 PROJECT

ACCELEROMETER-BASED SIGNAL ANALYSIS USING ARDUINO AND SIMULINK

BY Mannat Raina 124B1E174

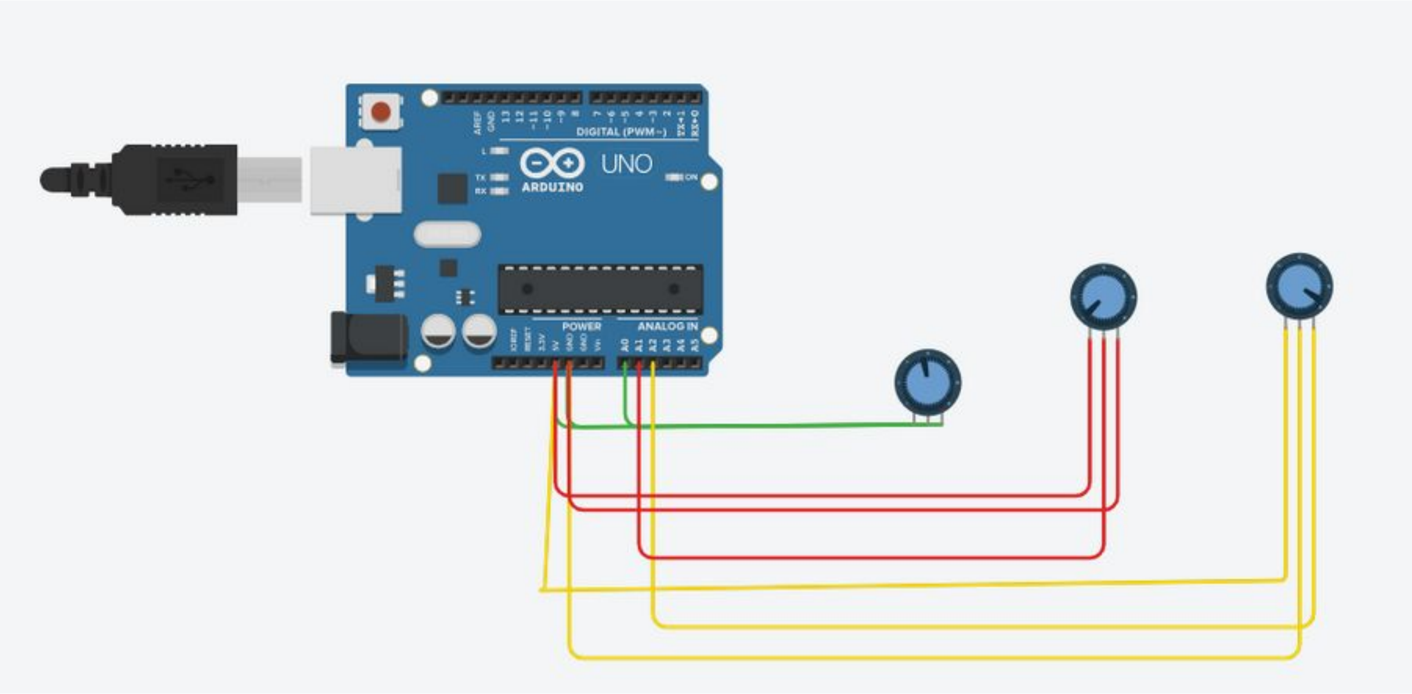
Mita Patil 124B1E177

Eshal Sayed 124B1E191

OBJECTIVE

To interface an accelerometer sensor with Arduino, capture real-time acceleration data, and analyze it in Simulink using Fourier and Laplace transforms to understand frequency and system response behavior.

TINKERCARD

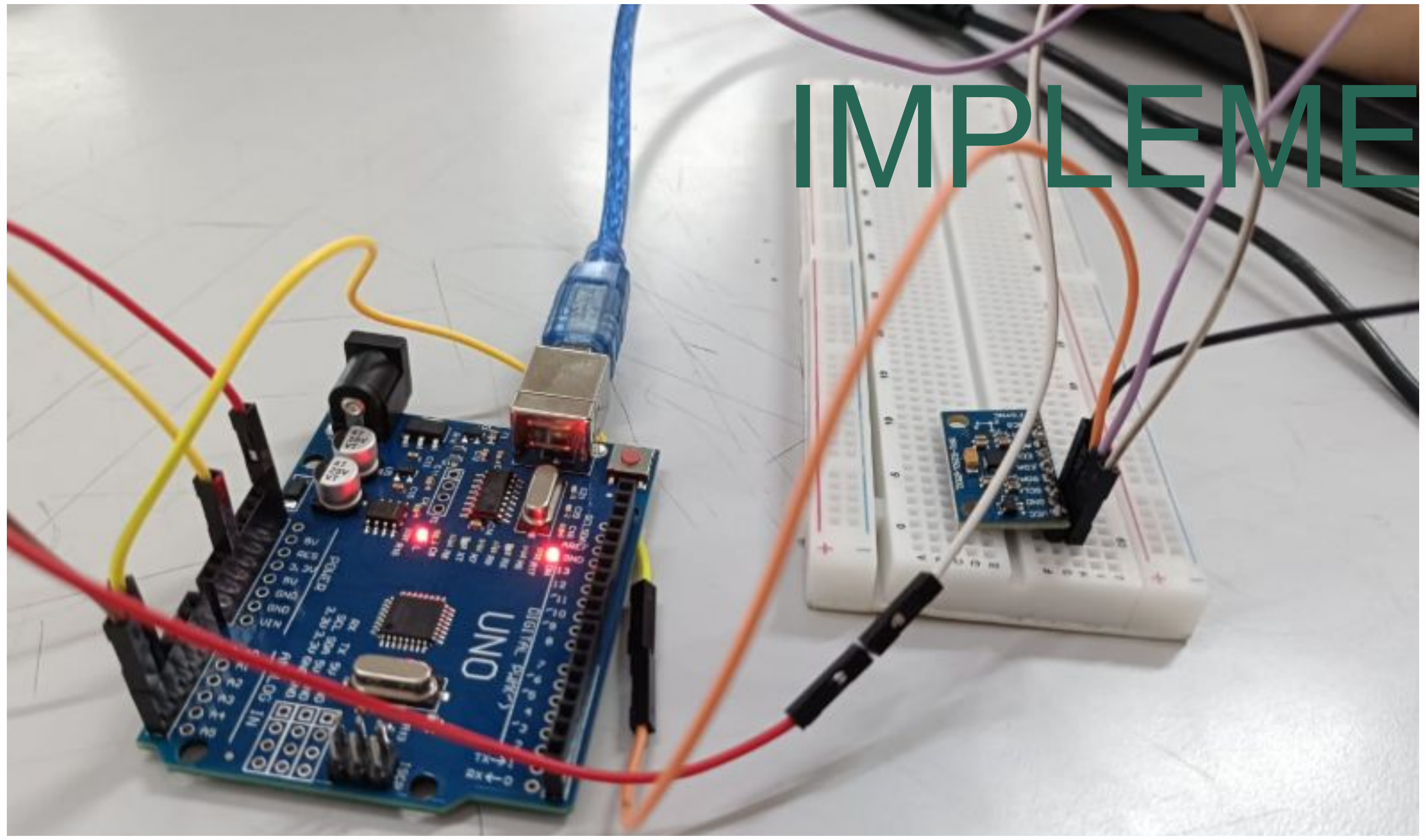


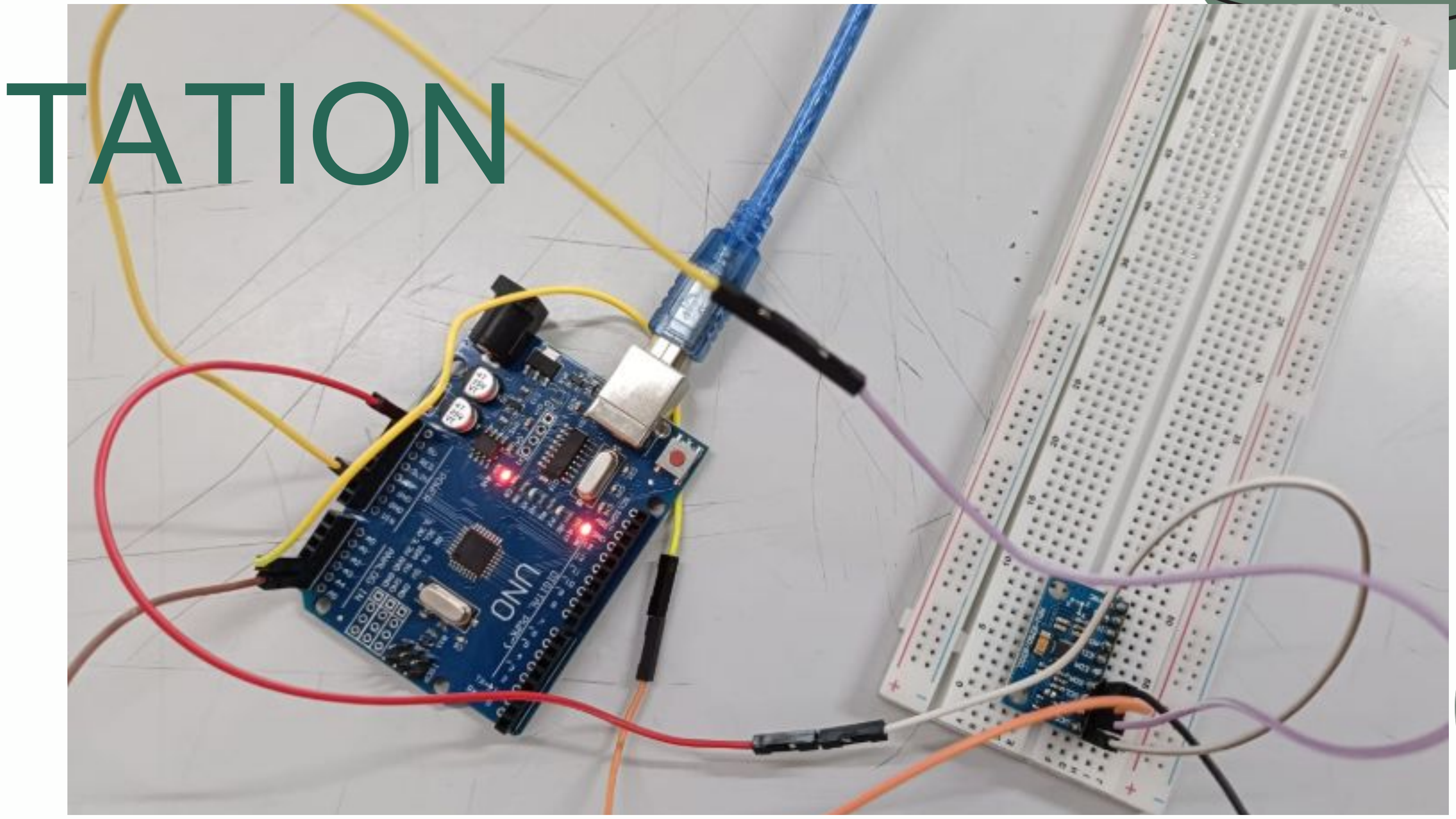
- Circuit was first simulated on Tinkercad to verify connections and sensor readings.
- The accelerometer outputs analog data to Arduino for further processing

ARDUINO CODE

```
void setup() {
 Serial.begin(38400); // Open Serial Monitor at 38400
}

void loop() {
 int ax = analogRead(A0);
 int ay = analogRead(A4);
 int az = analogRead(A5);

 // Map analog values (0–1023) to -1g to +1g
 float ax_g = map(ax, 0, 1023, -100, 100) / 100.0;
 float ay_g = map(ay, 0, 1023, -100, 100) / 100.0;
 float az_g = map(az, 0, 1023, -100, 100) / 100.0;

 Serial.print("Accel simulated -> X: "); Serial.print(ax_g);
 Serial.print(" Y: "); Serial.print(ay_g);
 Serial.print(" Z: "); Serial.println(az_g);
```

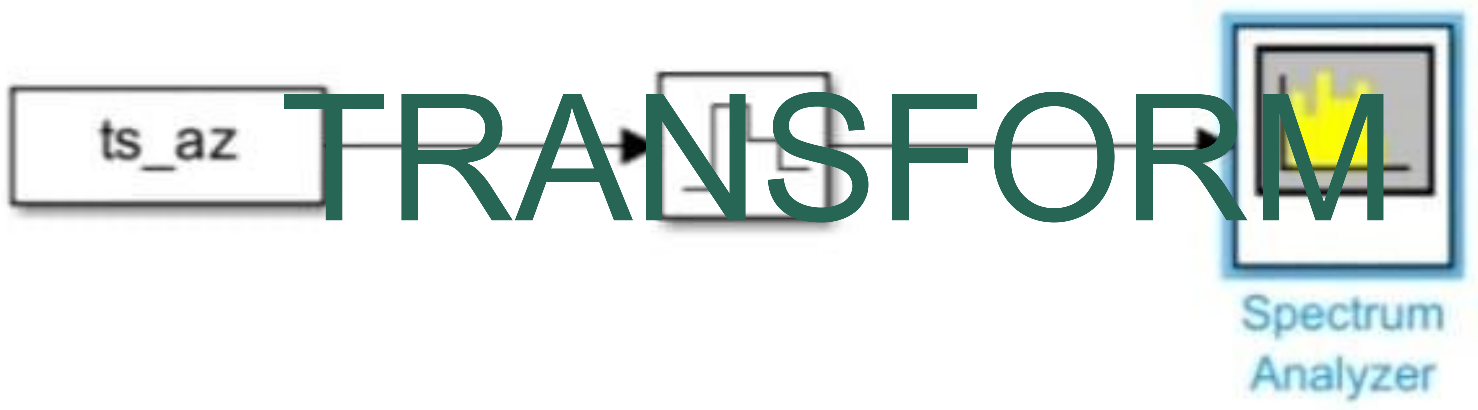
HARDWARE IMPLEMENTATION

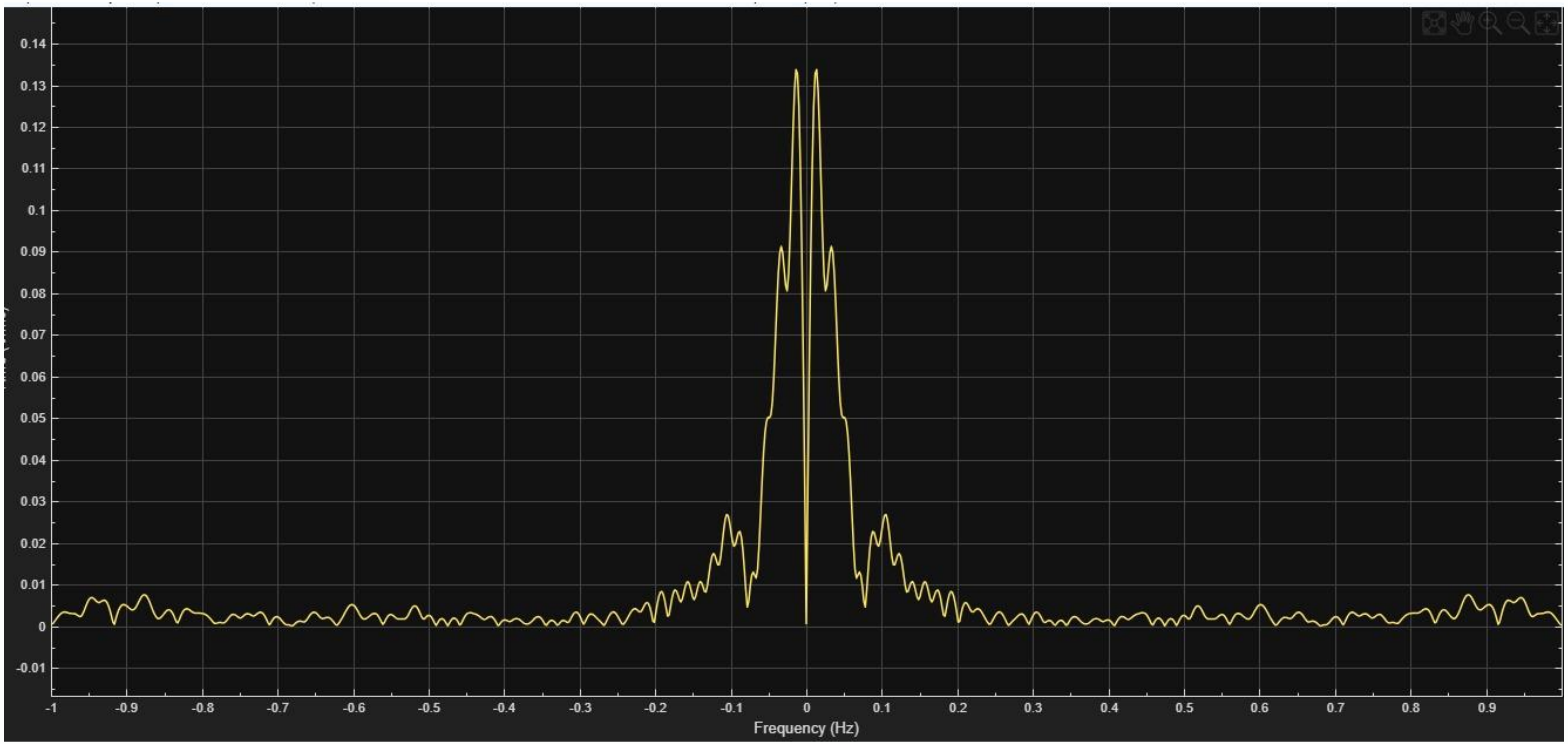




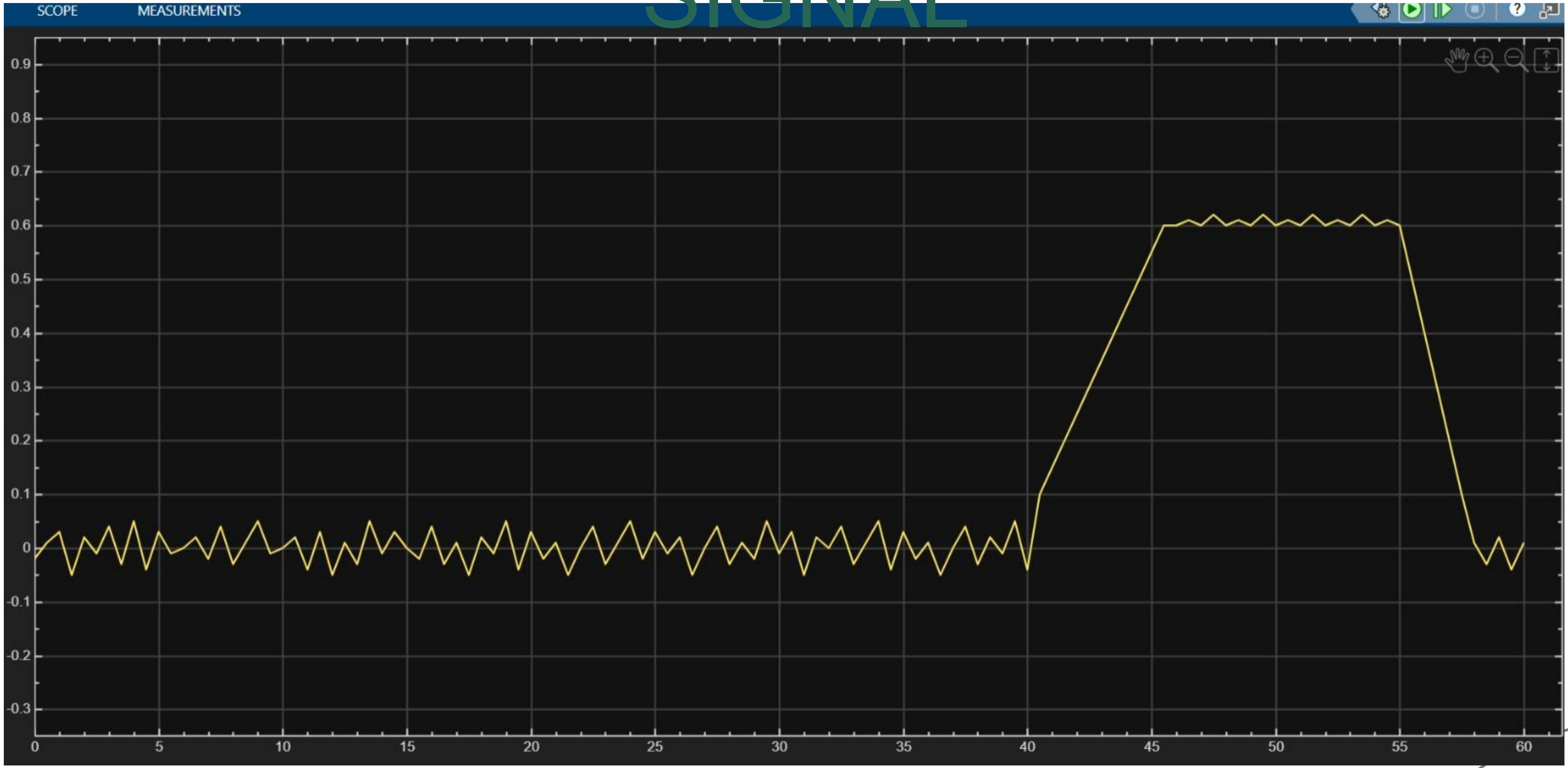
- The circuit was implemented using Arduino Uno and an accelerometer sensor (MPU6050).
- The accelerometer detects front and back, lateral and bumps(up and down) corresponding analog voltages on its X, Y, and Z pins.
- These analog signals are read by the Arduino’s analog input pins (A4, A5).
- The Arduino processes and sends this real-time data to the serial monitor or MATLAB for further analysis.
- USB connection powers the circuit.
- The setup was tested successfully on a breadboard, confirming proper sensor response to movement.

FOURIER TRANSFORM

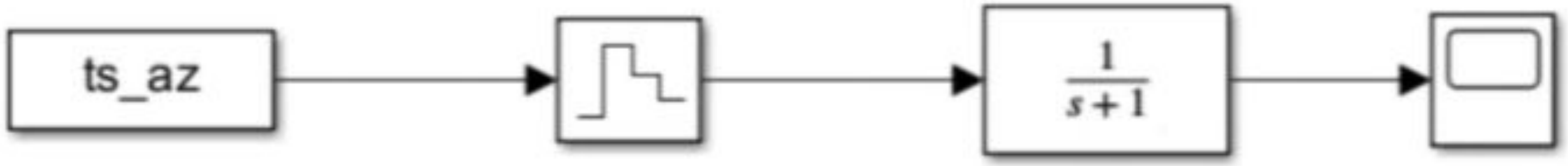


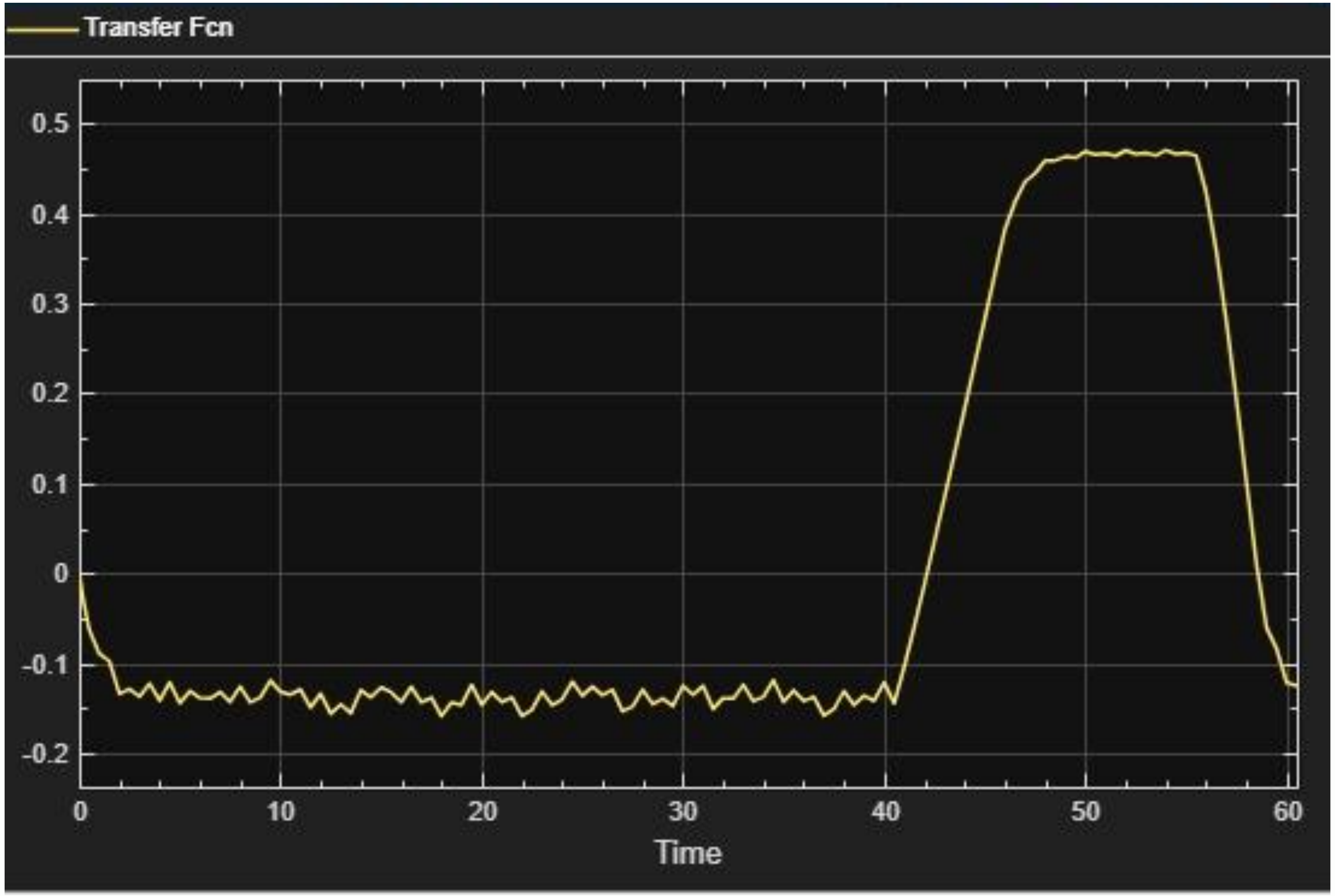


INPUT SIGNAL

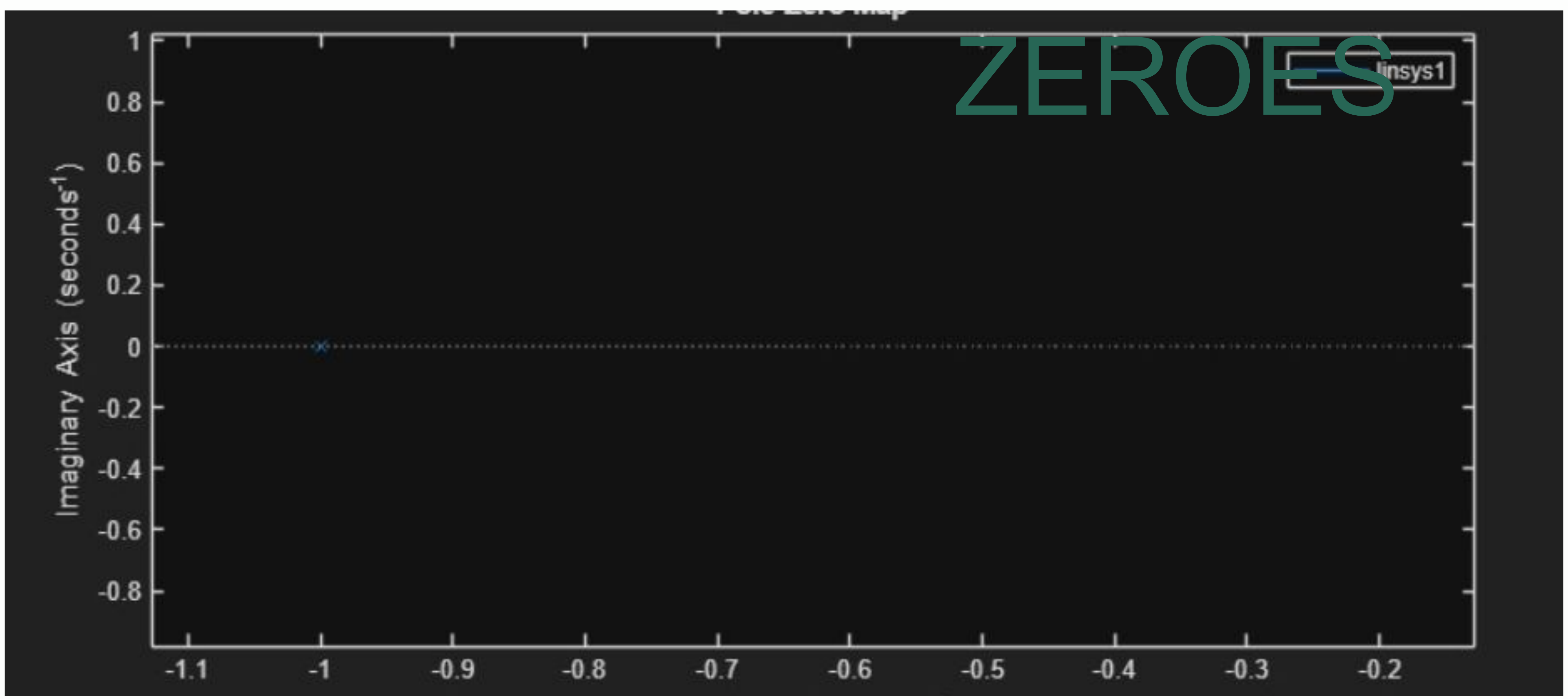


INVERSE LAPLACE





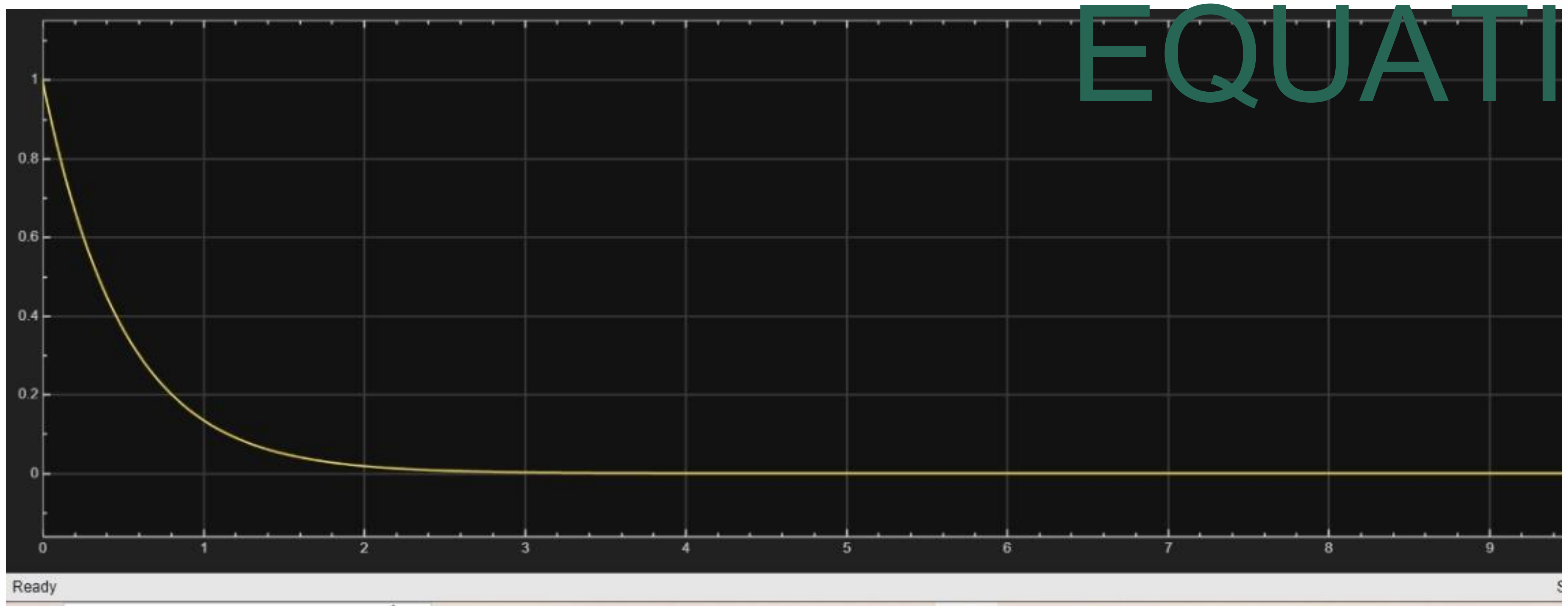
POLES AND ZEROES



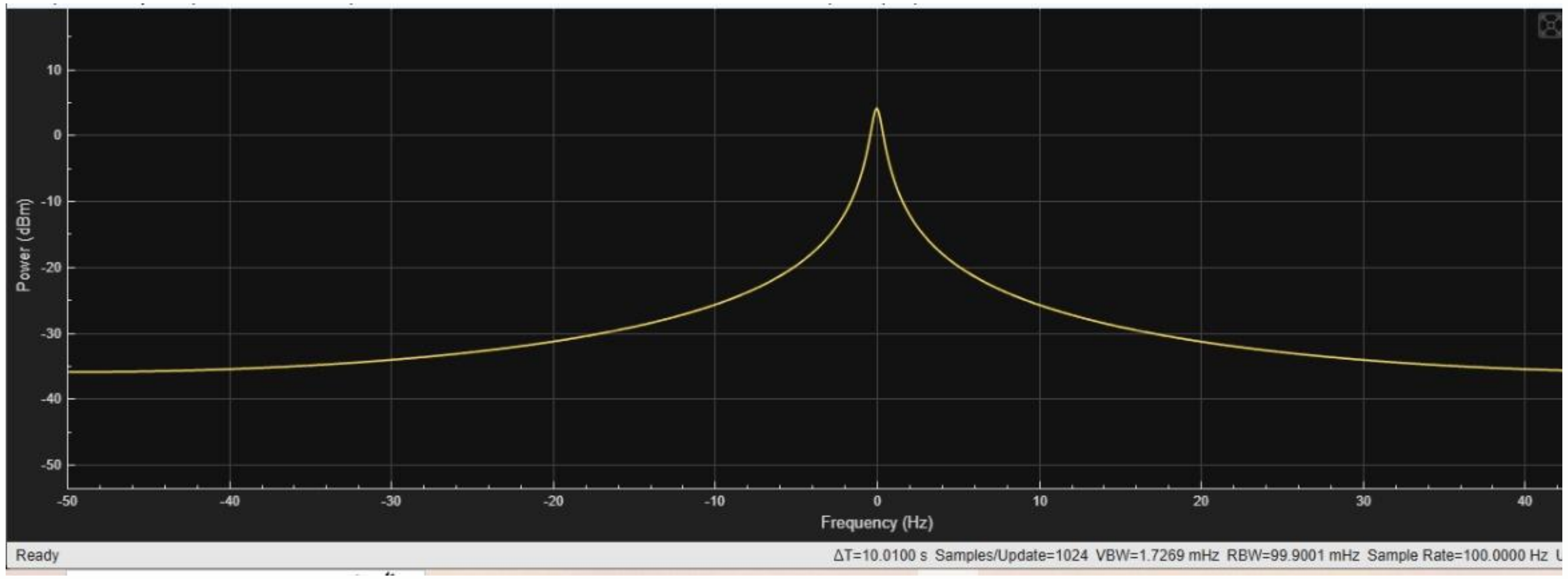
H(s)=1/s+1

- Zeros: None (numerator is constant → no finite zeros)
- Pole: One pole at s = –1
- Stability: Pole lies in Left Half Plane (LHP) → system is stable
- Time Constant:
- τ=1∣–1∣=1 second\tau = \frac{1}{|–1|} = 1 \text{ second}τ=∣–1∣1=1 second

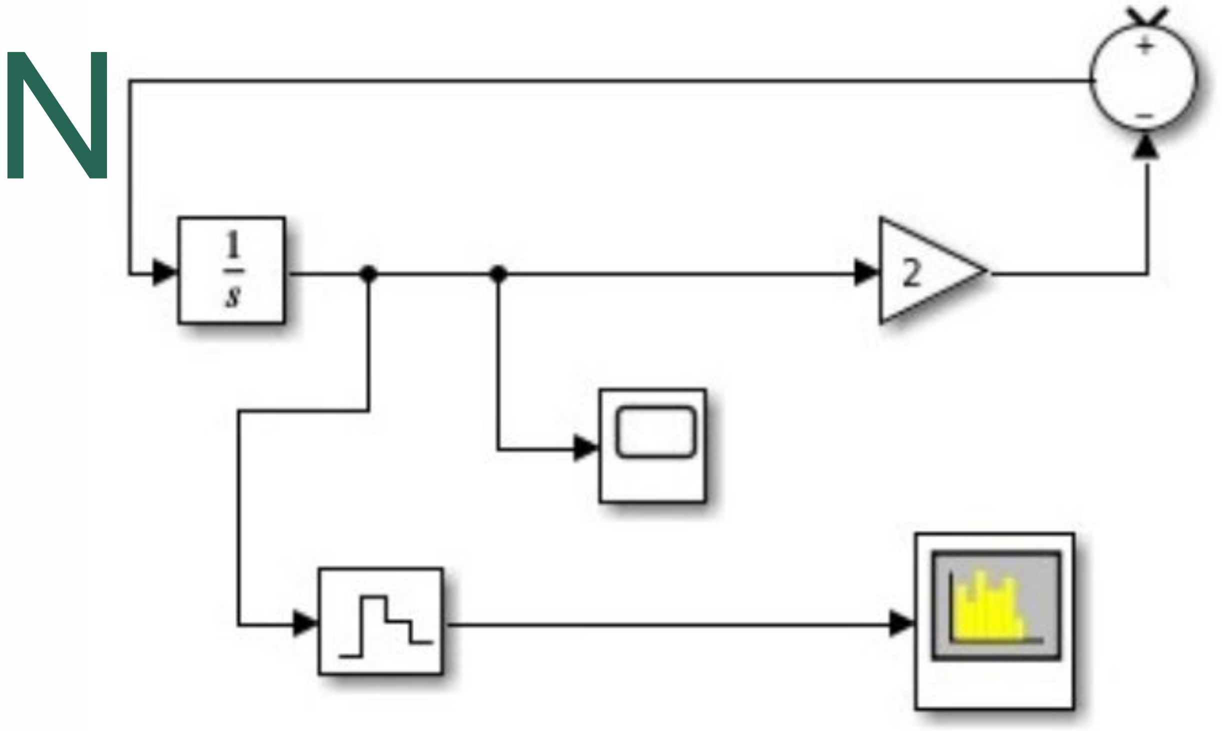
DIFFERENTIAL EQUATION



Ready



Ready ΔT=10.0100 s Samples/Update=1024 VBW=1.7269 mHz RBW=99.9001 mHz Sample Rate=100.0000 Hz



- **The Equation & Solution**: We modeled the natural decay equation $dy/dt=-2y$, and its direct answer is the exponential decay function $y(t)=e-2t$.
- **Time-Domain Plot (The Answer):** The Scope shows this answer as a simple curve that starts at 1 and smoothly falls toward zero over time.
- **Frequency-Domain Plot (The Analysis):** The Spectrum Analyzer graph shows this signal is made of low frequencies, with a strong peak at 0 Hz (DC) because the

CONCLUSION

- The project successfully demonstrated how an accelerometer can be interfaced with Arduino to measure motion and vibration.
- The collected data was analyzed in SIMULINK using Fourier and Laplace transforms to understand the system’s behavior in both time and frequency domains.
- The pole–zero analysis verified that the system is stable and well-behaved.
- Overall, the project combined hardware implementation and simulation analysis, strengthening our understanding of real-time signal processing and system modeling.

THANK YOU