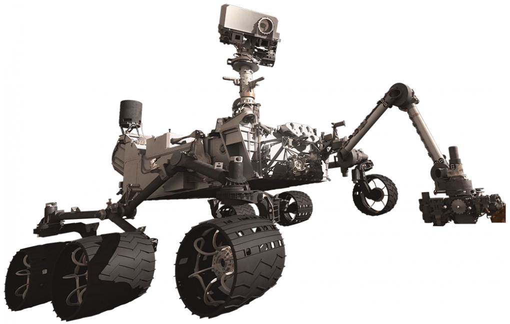
NASA Intelligent Industrial Machine Monitoring

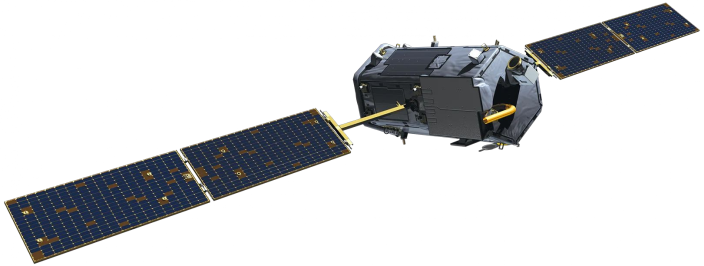
Group 1: Nick Bui, Mohamed Homsi, Aashrith Beesabathuni, Vincent Duong, Carter Chen

DISCLAIMER: Government Shutdown

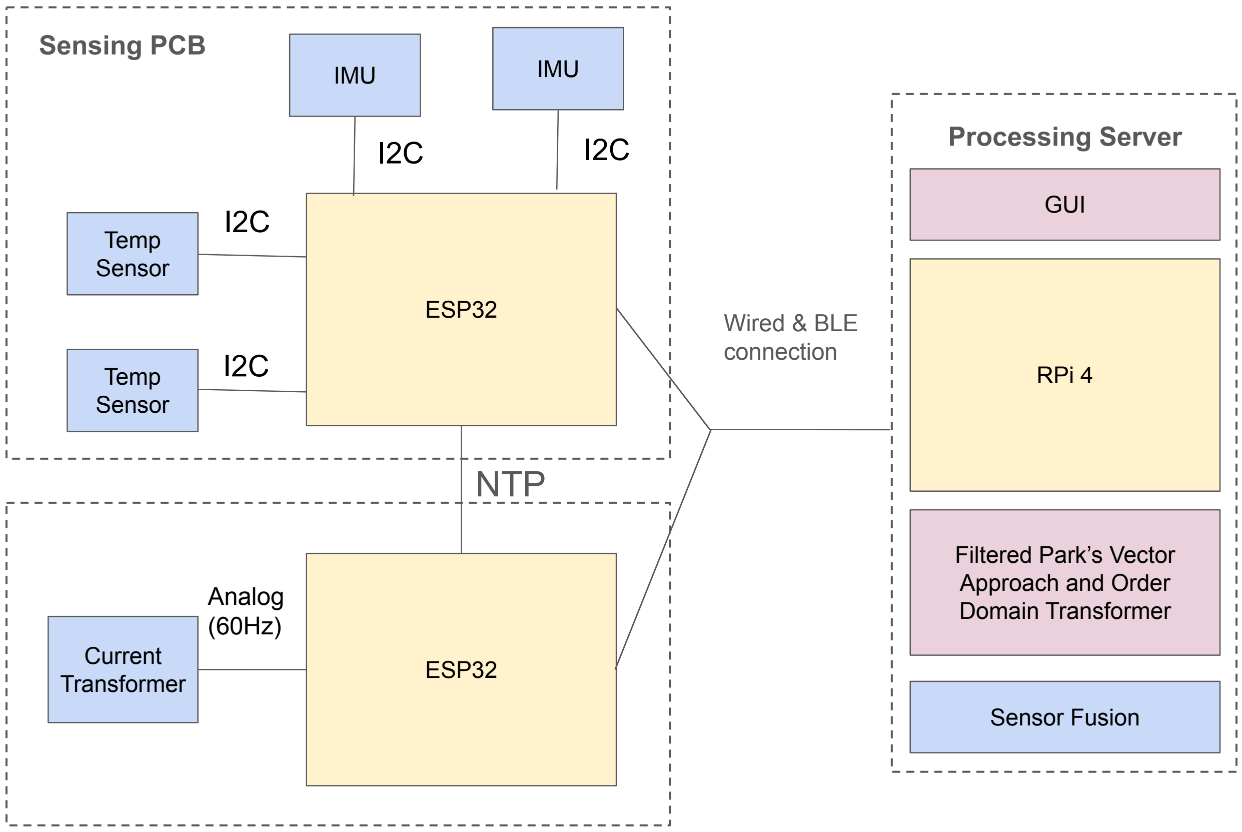
Motivation

- Industrial motors used **everywhere**: manufacturing robots, CNC machines, conveyor systems, etc.
    - (90% of industrial machines)
- NASA uses them to control rovers, satellites, drones, fuel pumps, cryogenic valves, etc.
- Typically induction motors: convert electrical energy into mechanical energy using electromagnetic induction
    - Produce vibration and heat → indicate motor health and efficiency
    - Faults (broken gear teeth, misaligned shafts) → excess power draw, increased heat, shift in vibration spectrum
- Problem Statement: induction motor-based industrial machines can develop faults such as misalignment, bearing wear, or gear damage that often go unnoticed until failure occurs. Therefore, we aim to create an intelligent monitoring system that continuously analyzes motor current, vibration, and temperature to detect faults early and improve overall system reliability.





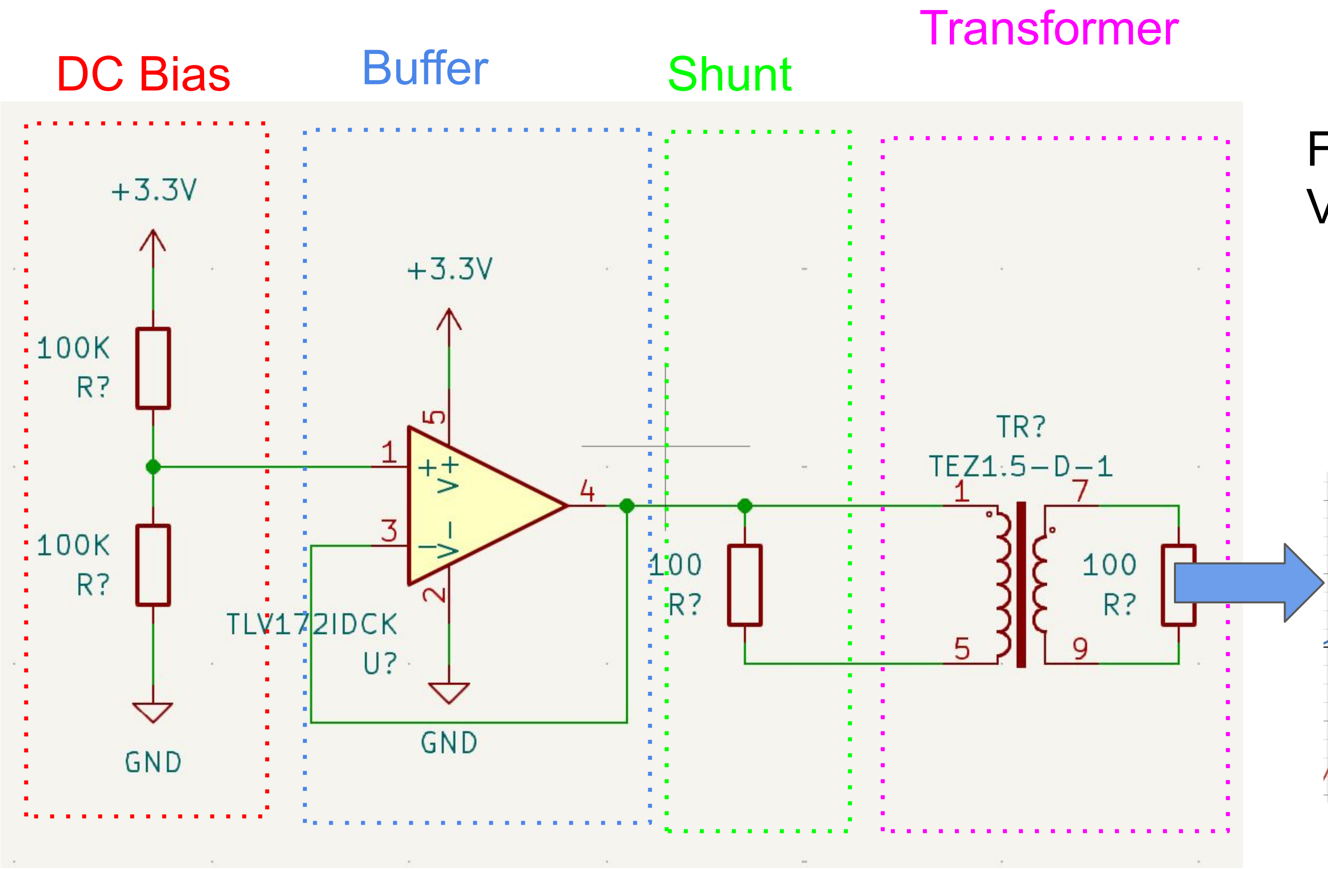
System Architecture



Course Topics

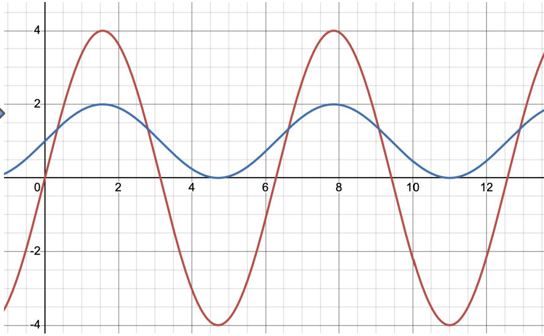
- Memory-mapped IO: read data from memory addresses associated with GPIO pins where sensors are connected to; use serial communication protocols such as I2C
- Concurrency: simultaneously measuring, processing, transmitting, and visualizing sensor data to and from the PCB and Raspberry Pi
- Sensor Calibration: use affine sensor models to calibrate our accelerometers, temperature sensors
- Networking: transmitting sensor data measured by the PCB to the Raspberry Pi via BLE and synching clocks across ESP32s using NTP
- Sensor Fusion: Marzullo’s algorithm or Kalman-based approach to extract meaningful measurements from noisy sensor data across multiple sensors
- Signal Conditioning: filter unwanted noise using LPF, downsampling to common sampling frequency

Current Transformer Schematic

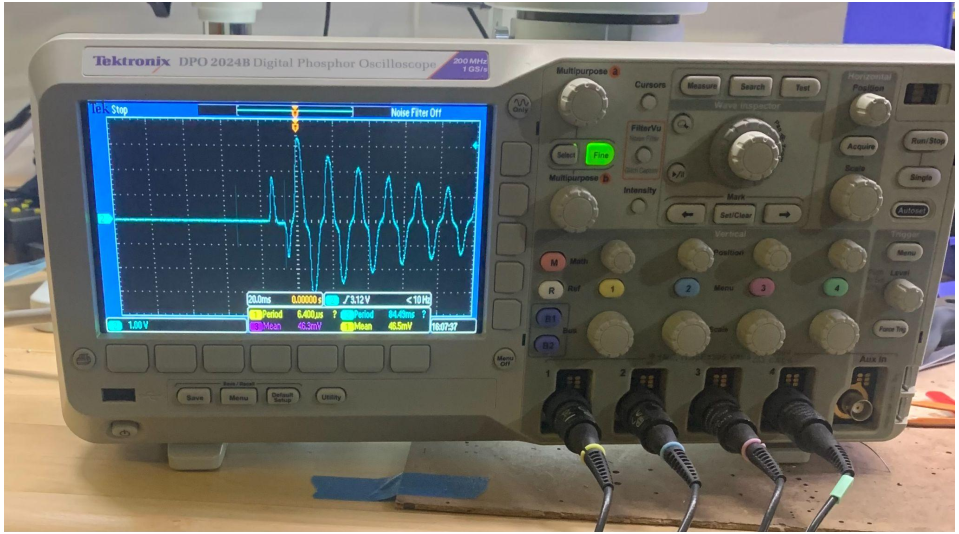


Find turns ratio:
V=IR

$$\text{Current ratio} = \frac{I_S}{I_P} = \frac{N_P}{N_S}$$



DEMO: 1 phase current



Evaluation

PCB:

1. Sensor Calibration:
    a. Current Transformer:
        i. Assemble CT circuit with 8.5A Hand Drill + Stripped Power Strip
        ii. Measure V across shunt resistor using 1-wire analog read on ESP32
        iii. Compare values read by ESP32 to Oscilloscope
    b. IMU and Temperature:
        i. Attach to ESP32 via I2C bus
        ii. Apply known heat and vibration to IMU and temperature sensor
        iii. Compare values read by ESP32 to known temperature and vibration
    c. Calibrate as needed
2. Breadboard Prototype Validation:
    a. Integrate ESP32s, CT, IMU, and Temp. sensor into breadboard prototype
    b. Make sure all readings are being measured correctly on ESP32 side

Processing Server:

1. Make sure readings from ESP32 are being received on RPi side with sub-20 ms latency (wired and BLE)
2. Test filtered park’s vector approach and order domain transformer on provided dataset
    a. Compare ground truth fault detection results with our predictions

Necessary Resources

- Different types of AC industrial motors (Dutta suggested using Cory motors)
  - Hand Drill
  - Brushless Motor
- Raspberry Pi 4 (RPi4)
- 2x ESP32
- Software
  - C for sensor calibration + read/transmit sensor data on ESP32
  - Python for sensor data processing and GUI on RPi4
- Sensors (accelerometers, temperature)
- Current transformers (mix of wired and wireless)
- Algorithms to process sensor data (order domain transformer, Park's vector approach, Marzullo's, N-point Moving Average, etc.)
- Advice/context from NASA stakeholders/mentors such as Dr. Rodney Martin

Timeline

Hardware:

1) Finalize specific electronic components (All) [ASAP]
2) Order electronics components (Vincent) [ASAP]
3) PCBs schematic and layout completed (Mohamed, Vincent, Aashrith) [11/14]
4) Breadboard prototype and sensor calibration [11/14]
    a) Temperature sensor and accelerometer (Mohamed)
    b) Current transformer (Vincent, Aashrith)
5) Breadboard prototype integration and testing (Mohamed, Vincent, Aashrith) [11/21]
6) PCBs ordered (Vincent) [11/21]
7) PCBs integrated and tested (Mohamed, Vincent, Aashrith) [11/28]

Software:

1) Write firmware to read sensor data and send to processing server (Nick and Carter) [11/14]
    a) Wired data transmission (Nick)
    b) Wireless data transmission (Carter)
    c) NTP networking protocol to sync data between MCUs (Carter)
2) Write sensor fusion and filtering algorithms to pre-process sensor data (Nick) [11/14]
3) Implement filtered park's vector approach algorithm (Carter) [11/21]
4) Implement order domain transformer algorithm (Nick) [11/21]
5) Test filtered park's vector approach and order domain transformer on provided dataset (Nick and Carter) [11/21]
6) Develop GUI (Carter) [11/28]

Major Risks

- Government Shutdown (unable to reach NASA mentors at this time)
  - Affected the (lack of) guidance we received, resulting in project goal and timeline shift
- High Voltage working with AC power and current transformers
- Clock Sync of MCUs, to accurately measure and sample our 3 sensors
- PCB Design: ensuring low-noise analog design
- Employing software to process error and faults in motors
- Adaptability of system to other motors

Contacts:

Vincent Duong (vincent_duong@berkeley.edu)

Nick Quach Bui (nicholasqbui@berkeley.edu)

Aashrith Beesabathuni (aashrith.b24@berkeley.edu)

Carter Chen (carter16891@berkeley.edu)

Mohamed Tarek Homsi (mohamedhomsi@berkeley.edu)