

## CIMES 2.5 kW Desktop Prototype Package v1 US Patent 11,799,400 B2 May 19, 2026

### Overview

This package provides everything needed to build a scaled 2.5 kW desktop version of CIMES. It uses the same core principles as the 10 kW design (frustum-cone rotor, staggered N52 magnets, axial compression throttle) but is smaller and easier to construct for testing and validation.

### Target Performance

- Power: 2.5 kW
- Rotational speed: ~1,000 RPM
- Axial compression range: 5–15 mm
- Safety factor: >8
- Thermal rise: <60 °C with basic air cooling

### Bill of Materials (BOM)

- Rotor and stator frames: CNC-machined aluminum or 3D-printed with metal reinforcement
- Neodymium magnets: N52 grade, 3 layers × 24 magnets per layer (scaled down)
- Linear actuator or high-precision stepper motor for axial compression
- Flywheel: ~2 kg·m<sup>2</sup> for ripple smoothing
- Shaft and high-quality bearings
- Torque sensor (optional but recommended for validation)
- Arduino Uno or ESP32 microcontroller
- Power supply and wiring
- Fasteners and structural supports

### Geometry & CAD Notes

- Rotor: Frustum cone, base radius scaled to ~0.12 m, 28° half-angle
- Stagger pattern: Use the new optimized angles [-42.3°, 17.8°, 61.2°]
- STEP files and full CAD will be provided upon request via the Starter Pack.

### Control Code (Arduino/ESP32)

C++

```
// Basic axial compression control for CIMES desktop prototype
#include <Stepper.h>

const int stepsPerRevolution = 200;
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);

void setup() {
  Serial.begin(9600);
```

```
    myStepper.setSpeed(60);  
}  
  
void loop() {  
    // Example: compress to 11 mm gap  
    Serial.println("Compressing to 11 mm...");  
    myStepper.step(200); // adjust steps for your actuator  
    delay(5000);        // hold and log data  
}
```

### Data Logging Script (Python)

Python

```
import serial  
import time  
  
ser = serial.Serial('/dev/ttyUSB0', 9600) # adjust port  
  
while True:  
    line = ser.readline().decode('utf-8').strip()  
    print(f"Timestamp: {time.time()} | Data: {line}")  
    # Add torque, RPM, gap logging here  
    time.sleep(0.1)
```

### Assembly & Validation Notes

1. Assemble rotor and stator with the new stagger pattern.
2. Connect the linear actuator to the microcontroller for precise gap control.
3. Run at 1,000 RPM and log torque/ripple at different gap settings.
4. Compare results to the published 2.5 kW target.

**Starter Pack v2** (including full CAD files) is available by emailing [danbucklives@gmail.com](mailto:danbucklives@gmail.com) with subject "CIMES Challenge Starter Pack".