Honda Goldwing Fuel Tank Analysis

Analysis derived from STL scan of fuel tank.

Tank Specifications

Property	Value
Total Capacity	17.34 L (4.58 gallons)
Tank Height	279.2 mm (11.0 inches)
Polynomial Fit R^2	0.999

> **Note:** Volume calculated by voxelizing the STL mesh and filling the interior cavity. The X-axis of the STL appears to be the vertical (fuel level) direction based on cross-sectional analysis.

Polynomial Equations

Volume from Height (for fuel sensors)

Given a fuel height reading `h_mm`, calculate remaining fuel:

```
h = h_mm / 279.2                    (normalize to 0-1)

v_norm = -12.240·h⁵ + 37.229·h⁴ - 41.972·h³ + 19.803·h² - 1.864·h + 0.035

volume_L = v_norm × 17.34
```

Height from Fill Percentage (for gauge calibration)

Given desired fill percentage `p` (0-100), calculate expected fuel height:

```
p_norm = p / 100                    (normalize to 0-1)

h_norm = 12.357·p⁵ - 29.952·p⁴ + 27.078·p³ - 10.813·p² + 2.215·p + 0.041

height_mm = h_norm × 279.2
```

Quick Reference Table

Fill %	Volume (L)	Volume (gal)	Height (mm)	Height (in)
5%	0.87	0.23	35.7	1.40
10%	1.73	0.46	49.8	1.96
15%	2.60	0.69	57.8	2.28
20%	3.47	0.92	62.5	2.46
25%	4.33	1.14	66.2	2.60
30%	5.20	1.37	70.0	2.76
40%	6.93	1.83	80.9	3.18
50%	8.67	2.29	96.1	3.78
60%	10.40	2.75	113.2	4.46
70%	12.14	3.21	130.2	5.13
75%	13.00	3.43	139.3	5.49
80%	13.87	3.66	150.1	5.91
90%	15.60	4.12	184.7	7.27
95%	16.47	4.35	214.6	8.45
100%	17.34	4.58	258.7	10.19

Arduino/ESP32 Implementation

```cpp
```

```
// Fuel tank constants
const float TANK_MAX_HEIGHT_MM = 279.2f;
const float TANK_CAPACITY_ML = 17336.8f;
const float TANK_CAPACITY_L = 17.34f;

// Get volume (L) from height sensor reading (mm)
float getVolumeFromHeight(float height_mm) {
    float h = height_mm / TANK_MAX_HEIGHT_MM;  // normalize 0-1
    float v = -12.240043f * pow(h, 5)
            + 37.229361f * pow(h, 4)
            - 41.972364f * pow(h, 3)
            + 19.803335f * pow(h, 2)
            - 1.863900f * h
            + 0.035199f;
    return constrain(v * TANK_CAPACITY_L, 0, TANK_CAPACITY_L);
}

// Get fill percentage (0-100) from height sensor reading (mm)
float getFillPercent(float height_mm) {
    return (getVolumeFromHeight(height_mm) / TANK_CAPACITY_L) * 100.0f;
}

// Get expected height (mm) for a given fill percentage
float getHeightFromPercent(float fill_pct) {
    float p = fill_pct / 100.0f;  // normalize 0-1
    float h = 12.357231f * pow(p, 5)
            - 29.951945f * pow(p, 4)
            + 27.078483f * pow(p, 3)
            - 10.812860f * pow(p, 2)
            + 2.214996f * p
            + 0.040888f;
    return constrain(h * TANK_MAX_HEIGHT_MM, 0, TANK_MAX_HEIGHT_MM);
}
```

Python Implementation

```
python
```

```python
import numpy as np

# Polynomial coefficients (5th degree, highest power first)
COEFFS_VOL_FROM_HEIGHT = np.array([
    -12.240043, 37.229361, -41.972364, 19.803335, -1.863900, 0.035199
])
COEFFS_HEIGHT_FROM_VOL = np.array([
    12.357231, -29.951945, 27.078483, -10.812860, 2.214996, 0.040888
])

TANK_MAX_HEIGHT_MM = 279.2
TANK_CAPACITY_L = 17.34

def volume_from_height(height_mm):
    """Get volume (L) from fuel height (mm)"""
    h_norm = height_mm / TANK_MAX_HEIGHT_MM
    v_norm = np.polyval(COEFFS_VOL_FROM_HEIGHT, h_norm)
    return np.clip(v_norm * TANK_CAPACITY_L, 0, TANK_CAPACITY_L)

def fill_percent_from_height(height_mm):
    """Get fill percentage (0-100) from fuel height (mm)"""
    return (volume_from_height(height_mm) / TANK_CAPACITY_L) * 100

def height_from_fill_percent(fill_pct):
    """Get expected height (mm) for fill percentage (0-100)"""
    p_norm = fill_pct / 100
    h_norm = np.polyval(COEFFS_HEIGHT_FROM_VOL, p_norm)
    return np.clip(h_norm * TANK_MAX_HEIGHT_MM, 0, TANK_MAX_HEIGHT_MM)
```

Accuracy Notes

- **Polynomial R^2 = 0.999** for volume from height - excellent fit
- **Maximum residual error:** ~300 mL (~2% of total capacity)
- The model assumes the STL scan represents the actual interior volume
- Real tank may have baffles, pickup tubes, or reserve sections not captured in the scan
- Sensor mounting position may require offset calibration
- If the tank orientation assumption is incorrect, swap the axis analysis accordingly