<Project> Lever Weighing Device with Gyroscope Sensor

Author

XD

Introduction

General Requirements

Design a single-object mass measurement system based on Arduino UNO with various types of sensors to measure the actual mass of an object with a range of about 75g~750g as accurately as possible without the limitation of using strain-type sensors.

Detailed Requirements

a) The system must use some type of sensor (other than strain gauge).

b) Range: about 75g~750g.

c) The error is within ±5g.

Experimental Instruments

1. Arduino UNO.

2. Computer.

3. DuPont cable.

4. Student power supply.

5. Roller screw linear module (with stepper motor).

6. Stepper motor driver board.

7. Aluminum profile bracket.

8. two flange seat bearings.

9. 3D printing parts.

10. two 500g weights.

11. GY25 gyroscope module with MPU6050.

Implementation and analysis

Implementation of quality measurement

A linear module is used as a lever, and the lever pivot point is connected to a support that can be freely rotated. Flange housing bearings are used to minimize the friction generated during rotation. The weight and pallet are placed at one end of the module, and a counterweight of known mass is fixed on the module slider. The mass of the weight is measured by driving a stepper motor that controls the slider and the displacement of the counterweight on it to change the torque on this side.

In the process of estimating the mass of an object from known moments, we use the principle of leverage. The ratio between the output force and the input force is equal to the inverse ratio of the vertical distance between the two forces and the fulcrum respectively, which is called the "leverage principle" and expressed in the equation:

$$F_2 : F_1 = D_1 : D_2$$

Define moment M for:

$$M \triangleq FD$$

where F is the applied force and D is the vertical distance between the applied force and the fulcrum.

Then the input torque is equal to the output torque:

$$M_1 = M_2$$

Through this principle, we can calculate the mass of the object to be measured by sensing the displacement of the counterweight of known weight.

Implementation of balanced perception

A GY25 gyroscope is fixed on the lever to detect the angle between the lever and the level in real time. By the positive or negative angle, you can judge which end of the lever is heavier.

Figure 1 GY-25 Module

GY-25 MPU6050 3-Axis Gyroscope Sensor Module [3V-5V] is a high-resolution tilt angle (YAW, ROLL, and PITCH) sensor module which is based on MPU6050 MotionTracking chip and an MCU chip. The MPU-60X0 series chips are small motion-tracking chips, that integrates a tri-axis gyroscope, tri-axis accelerometer, a Digital Motion Processor™ (DMP) and two sets of three 16bits-ADC for converting the tri-axis gyroscope and tri-axis accelerometer analog output to digital signals.

The GY25 module has an internal MCU that compile and process output data of the MPU6050 sensor chip to achieve angle tilt quantity. The process of calculating the attitude angles from the MPU6050 chip typically involves the following steps:

First, the magnitudes of the object's accelerations along the three spatial axes are computed from the accelerometer measurements. Then, using these acceleration magnitudes, the pitch and roll angles of the object are calculated. This is done by applying inverse trigonometric functions (such as the arctan2 function) to obtain the desired angles.

The gyroscope measurements represent the object's angular velocities along the three spatial axes. These angular velocity measurements can be used to estimate the object's yaw angle. Typically, integration of the angular velocities over time can yield changes in the yaw angle.

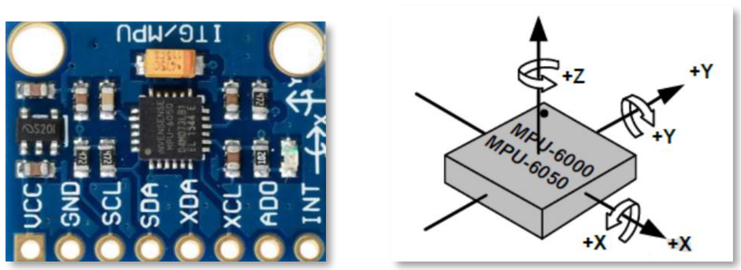

Figure 2 MPU-6050

Finally, by fusing the accelerometer and gyroscope measurements, more accurate estimates of the attitude angles can be obtained. Common methods for this fusion include complementary filters or Kalman filters. These filtering algorithms weight the accelerometer and gyroscope measurements to provide more accurate and stable estimates of the attitude angles. The data obtained from the MPU6050 chip is affected by significant noise, and even when the chip is in a stationary state, the data exhibits fluctuations exceeding 2%. Additionally, there are offsets present in the readings across all data axes.

Kalman filter

To address the issue of significant noise and offset in the data obtained from the MPU6050 chip, a two-step process can be implemented: data calibration for offset correction and noise reduction using the Kalman filter algorithm.

The Kalman filter is a recursive data processing algorithm that effectively filters out noise from sensor measurements and provides a more accurate estimate of the underlying state. It combines the information from the sensor measurements and the system model to estimate the true state while minimizing the impact of noise.

1. Prediction:

In the prediction step, the Kalman filter predicts the current state based on the previous state estimate and the system dynamics model. The prediction equation is as follows:

$$\hat{x}_{k}^{-} = F * \hat{x}_{k-1} + B * u_{k-1}$$

where:

- $\hat{x}_k^-$ is the predicted state at time k

- F is the state transition matrix

- $\hat{x}_{k-1}$ is the previous state estimate

- B is the control-input matrix

- u_{k-1} is the control input

2. Update:

In the update step, the Kalman filter incorporates the sensor measurements to refine the state estimate. The update equations are as follows:

$$y_k = z_k - H * \hat{x}_k^-$$
$$K_k = P_k^- * H^T * (H * P_k^- * H^T + R)^{-1}$$
$$\hat{x}_k = \hat{x}_k^- + K_k * y_k$$
$$P_k = (I - K_k * H) * P_k^-$$

where:

- y_k is the measurement residual

- z_k is the sensor measurement

- H is the measurement matrix

- P_k^- is the predicted error covariance matrix

- R is the measurement noise covariance matrix

- K_k is the Kalman gain

- I is the identity matrix

- P_k is the updated error covariance matrix

Implementing data calibration for offset correction and applying the Kalman filter algorithm can help mitigate the noise and offset issues in the data obtained from the MPU6050 chip, leading to improved accuracy and reliability in the measurements.

Figure 3 Kalman Filter

Implementation of model establishing

According to the above theory, the model is established as follows:

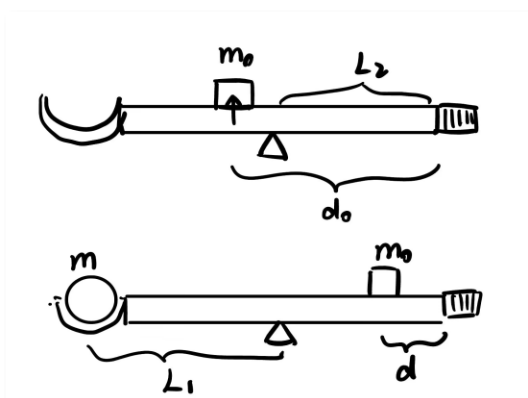

Figure 4 Model Schematic

The meaning of each variable in the figure is as follows:

- m_0: The total mass of slider and counterweight.
- d_0: is the position of the slider when there is no weight in balance.
- M: The mass of the lever itself.
- L_1: is the distance from the center of mass to the fulcrum.
- L_2: The distance from the center of gravity of the lever to the fulcrum.
- M: Amount of analyte.
- d: The position of the slider when there is a heavy object in balance.

When the weight to be measured is not placed, the lever balance relationship is as follows:

$$m_0(d_0 - L_2) = L_2 M$$

When placing the weight to be tested, the lever balance relationship:

$$mL_1 = L_2 M + m_0(d_0 - d)$$

Simultaneously obtain the relationship between mass and slider position:

$$mL_1 = m_0(d_0 - d)$$

The ideal model is a linear model.

Method correction and improvement

The displacement d_1 required for the lever to be lifted from the motor end to the weight end, and the required displacement d_2 from the weight end to the motor end.

Due to the friction of the pulley, d_1 and d_2 are not equal, the relationship is as follows:

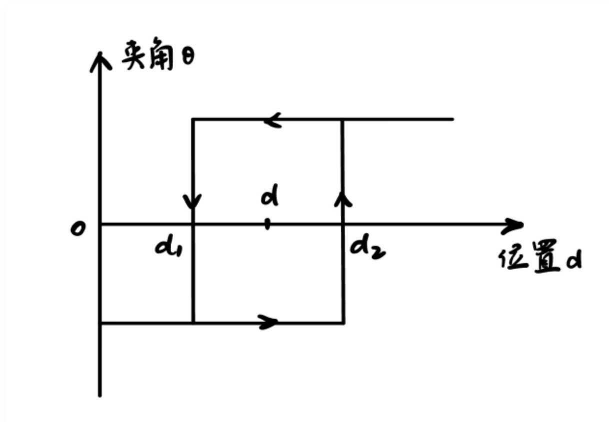

Figure 5 Process Diagram

After testing and analysis, we choose

$$d = \frac{1}{2}(d_1 + d_2)$$

as an observation, to solve the mass m to be measured.

System Design

Hardware design

The entire project instrument takes the linear module as the core component, and is assisted by linear module brackets, bearing frames, aluminum profiles, weight seats, weighing brackets and other components to form an overall structure that can achieve lever balance weighing.

The linear module is connected to the bearing through the linear module bracket, and the bearing is fixed on the aluminum profile, so that the linear module can use the bracket as the axis to realize the slight rotation of the lever. The slider on the linear module is connected to the weight rack, and two 500g weights are placed on it, and the weight rack and the object move together to achieve the purpose of balancing the lever. The other end of the linear module is connected with the weight support.

In addition to the basic structure, a fixed part is added between the two aluminum profiles to limit the relative position between the two aluminum profile brackets, and at the same time limit the swing angle of the linear module to achieve faster balance.

The Solidworks modeling effect of the assembly and the weight seat is shown in the figure:

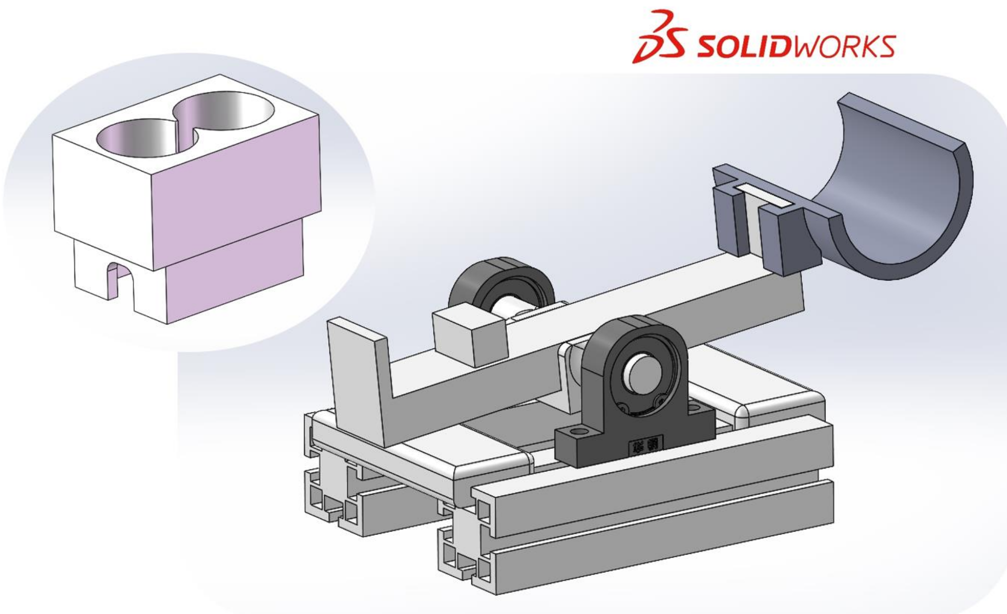

Figure 6 Solidworks Modeling Effect

Software design

The implementation logic flow chart of the quality measurement system is as follows:

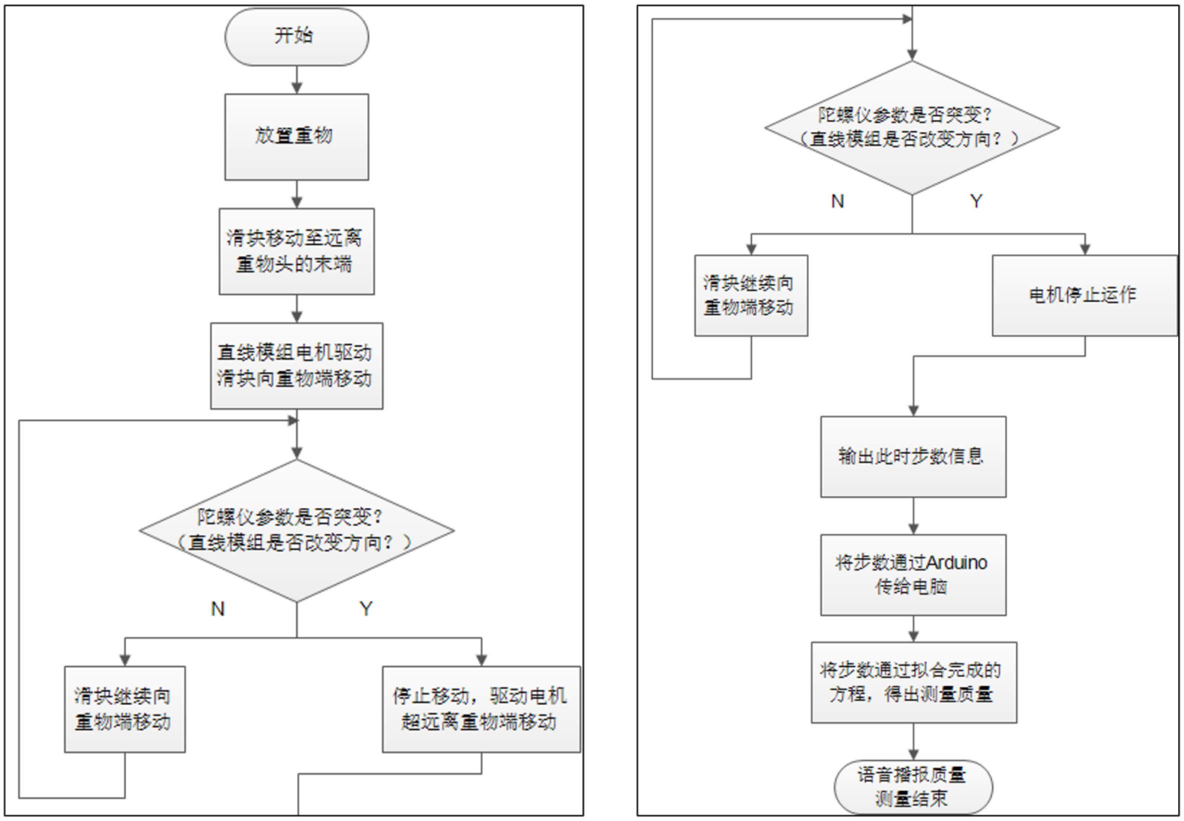

Figure 7 Logic Flow Chart

Data Calibration

Lab data calibration

Place the objects to be tested with different qualities on the measuring bracket and execute the program to get the step number information transmitted by Arduino to the computer. The quality of the item to be tested and the obtained step number information are fitted by a straight line through Excel, and the first-order fitting equation of the step number information relative to the quality of the item to be tested can be obtained. In the subsequent measurement, the step number information corresponding to the unknown quality item is substituted into the first-order fitting equation, and the quality measurement value of the item to be tested can be obtained through calculation.

The experimental results are as follows:

d	m
838.75	0
825	27.8
755	121.6
702.5	198
661.25	253.3
628.75	302
581.25	368.9
545	414
535	432.6
501.25	481.5
491.25	498.8
470	528.9
456.25	548.1
435	578.3
407.5	613.8

Figure 8 Data Sheet

According to the principle of least squares linear regression, make a scatter plot and fit:

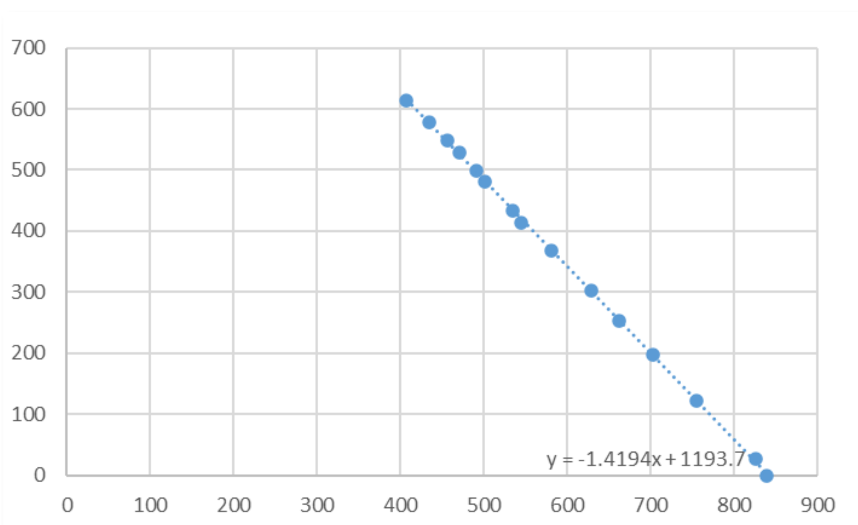

Figure 9 Scatter Plot

The relationship between mass and position obtained by linear fitting is as follows:

$$d = -1.4194m + 1193.7$$

Results and Error Analysis

Results

Eventually, when the quality of the test item is measured and tested, the relative error is stable within $\pm 1\%$.

Error analysis

1. The stepper motor has low subdivision accuracy, voltage imbalance, and errors caused by current gain, which may easily cause the number of steps and distance between the slider and the weight to be inaccurately one-to-one.

2. Errors caused by delay and jitter during gyroscope measurement.

3. The main part of the measurement object is liquid, and the center of mass will shake with the shaking of the device, which may cause errors.

4. The measurement data sample is small, and the linear equation is underfitting/overfitting, resulting in errors.

Problems encountered and solutions

1. Six-axis MPU6050 data jitter: use of Kalman filter.

2. Zero-drift of raw angular acceleration data: apply GY25 attitude data fusion.

3. The measuring range is a little small: adjust the quality of the counterweight and improve the system structure.

4. Bearing friction hinders accurate balance: adjust the mechanical structure and algorithm.

5. Insufficient pin resources of the UNO board: programming for soft serial communication.

6. Board memory usage overflow: combination of soft Uart and hard Uart.

Team Task Allocating

NULL