A self stabilizing platform

Vladimír Popelka
Institute of Automotive Mechatronics
Faculty of Electrical Engineering and Information Technology
Slovak University of Technology in Bratislava
Bratislava, Slovakia
vlado.popelka@gmail.com

***Abstract*— There exist mechanical devices for which it is important to retain a constant position, or a constant direction regardless of their space fitting. Classic applications of such systems include a camera stabilization in moving systems, a platform stabilization for mobile guidance systems, or other special cases. For this and other cases, it is appropriate to use the now widespread and relatively accurate MEMS gyroscopes and accelerometers. Their consumption and size are negligible and can be used right from the smallest models or toys.**

This paper reports on a particular solution build on the sensor MPU6050 and its parameters that can scan 6DOF (3 x 3 accelerometer + gyroscope). In real conditions this task is not trivial, since the signals of these sensors are quite noisy and influenced by other variables such as reaction speed, inertia, and accuracy of data interpretation, sensor drift and others. Self-stabilizing platforms can be of great use especially in the automotive, aviation, marine, robotics, aerospace as well as in everyday life.

Keywords*— *embedded control; measurement; MEMS; G-force; accelerometer; gyroscope; communication

I. Introduction

In nowadays, lots of micro-electronic components, sensors and single chip microcomputers exist with very good performance by minimal power consumption. One area of new technology is MEMS devices and sensors. The abbreviation of MEMS is sign for Micro Electro Mechanical System. This makes labeling of technology or sensor directly. They are in complies with the requirements of low cost, low energy and low dimension and weigh. Motivation for creating the self-stabilizing platform is use this technology for demonstration and for collecting experiences with practical use. This experiment is only one of wide possible use-cases. The self-stabilizing platform can have also some real uses for example in automotive, industry, avionic, robotics etc. This paper describes theory and creation of self-made platform with standard parts available. Most of them are low-cost and easy to buy. The result of this project is to hold in balance some object placed on plane of platform. In the implementation, we met also some problems that have been solved. Sensor for motion information measurement is MPU6050 [6], [7], [8]. Control part is independent, it can be for example AVR 8bit or ARM 32bit. There were tested specifically Arduino pro mini board, AVR (atmega8535) standalone board and STM32F4 discovery board.

II. Theory

The construction of platform can be divided into the following sections.

A. Platform type

The first breakdown is by DOF (Degrees Of Freedom) as follow.

- 1 DOF – one axis stability – tilt x
- 2 DOF – two axis stability – tilt x, y
- 3 DOF – three axis stability – tilt x, y and rotation z
- >3 DOF – full axis stability in space

At higher degrees of freedom is possible completely stabilize the platform in a space. To calculate the position of arms, that hold the platform, is suitable to use inverse kinematic.

B. Construction

The second division is in MEMS sensor position.

1) *Sensor on top of the plattform*
This deal is beneficial measuring real values of G-force and angular acceleration direct on platform and output is independently of base arm position. Other pros is in separation of mechanical properties of base arm and platform. Disadvantage is in sensor noise while plattform is moving.

2) *Sensor on the base arm of platform*
This solution has lower noise from sensors, but not so perfect trace actual plattform position.

After consideration of those characteristics was selected type1.

C. Driving mechanism

Power parts can be linear actuators, servomotors, stepper motors with gearbox, hobby-servos, hydraulic system etc. Each has advantages and disadvantages but the key selection parameter is speed of positioning (reaction time), carrying capacity and precision. For this simple test purpose is used 9g servo.

978-1-4799-3528-4/14/$31.00 ©2014 IEEE

D. Sensors

Sensor was chosen MEMS gyroscope and accelerometer in one package - MPU6050. There is possible to create device with < 2DOF with accelerometer only, or gyroscope only, but better solution is sensor fusion. Advantages are described in part “Filter”.

E. Control system

Control type of platform is linked with platform type – degrees of freedom, speed of setting mechanism and used software filter, which needed processor time. System control focus is set only for using single chips. There are two choices of single chips.

1) *8bit core single chip*
   It can be AVR (frequently used as an Arduino), PIC, or older 51-core family procesors. Cons of these processors is relatively slow data processing. They needed more machine cycles for processing one value bigger than 8 bit. For example with using 10bit output of A/D converter is needed two instruction for read one A/D value. But they are still popular for their simplicity, speed of development and good learning curve.
2) *32bit core sigle chip*
   Probably biggest representative in the field of embedded devices are ARM processors. They allow to use modern programming techniques, control performance, they can speed up calculation and data processing, using DMA, nested vector interrupts. More of them, specifically ARM core M3 and higher, contain FPU (Float Point Unit) and support for SIMD instructions (Single Instruction, Multiple Data). Direct hardvare support for digital signal processing (DSP), image sensor, LCD display support, real time etc. Speed and memory capacity is also much bigger than 8 bit computers and reach more than 100 Mhz and more than 1 MB of memory.

The sketch of platform is on Figure 1.

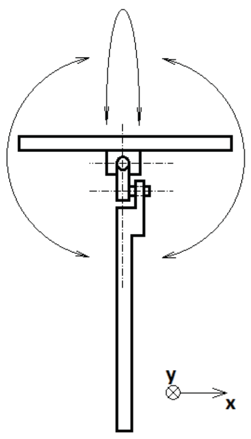


Fig. 1. Sketch of platform with 2 DOF

Detail of the real platform model is on Figure 2. Here we can see two hobby servos on aluminium construction and MEMS sensor (green PCB). Communication cables, from sensor, leads to level converter behind the main arm.

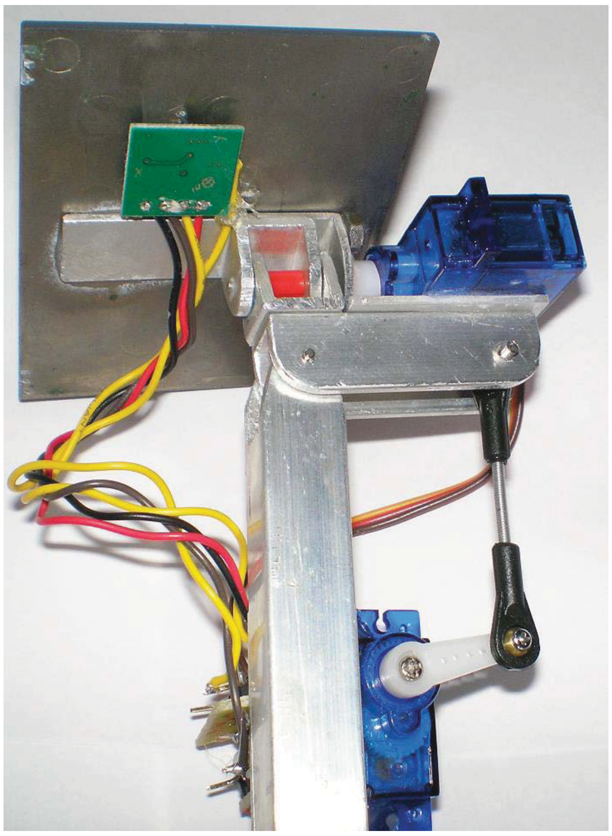


Fig. 2. Real model of self stabilizing platform.

III. Inverse Kinematics

The problem of inverse kinematics is, given the end effector position and orientation, to find the joint angles. In general, inverse kinematics is much harder than forward kinematics. Sometimes no analytical solution is possible, and an iterative search is required. Even with analytical solutions possible, multiple solutions arise from which one must pick up. Another complication is that workspace limits may be violated (the point is outside the reach of the manipulator arm).

IV. Sensor

Gyroscope features are user-programmable full-scale range of ±250, ±500, ±1000, and ±2000°/sec, 3x integrated 16-bit A/D, bias and sensitivity temperature stability [1], digitally programmable low-pass filter

Accelerometer Features are programmable full scale range of ±2g, ±4g, ±8g and ±16g, 3x integrated 16-bit A/D. Sensor orientation is shown in Figure 3.

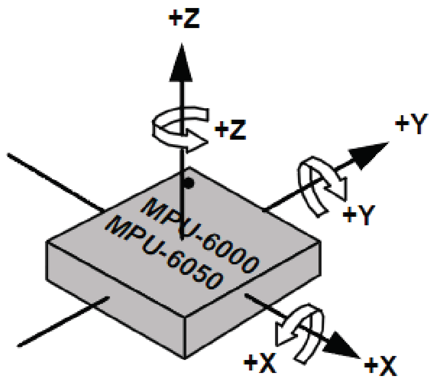


Fig. 3. Sensor orientation.

A. *Sensor calibration*

At the beginning it is necessary to calibrate the sensor. The calibration consists of measuring 1000 values for each axis and averaging. The result values will be subtracted from current values.

B. *Sensor normal operation*

Maximum sensor data output rate is 1 kHz. For test and development purpose (send data out with USART) has frequency set to 250 Hz (data each 4 ms), is sufficient at this moment. The USART – Universal Synchronous/Asynchronous Receiver/Transmitter interface is very old but to nowadays it still unsurpassed, mainly for its simplicity and enlargement. This interface has been selected also for easy implementation and quite fast data transfer.

V. FILTERS

Output from MEMS are relatively noisy, therefore, for better data processing, it is necessarily to use filter. Some sensors contain hardware filters inside, but in this case it is not usable. Other solution is using software filter. Because we don't want any additional delay between data (using internal filter creates delays), it will be better to implement custom filter with desired parameters. The most used types of filters are:

- Kalman filter

The Kalman filter algorithm produces estimates of the true values of sensor measurements and their associated calculated values by predicting a value, estimating the uncertainty of the predicted value, and computing a weighted average of the predicted value and the measured value. The most weight is given to the value with the least uncertainty. The estimates produced by the algorithm tend to be closer to the true values than the original measurements because the weighted average has a better estimated uncertainty than any of the values that went into the weighted average.

- Complementary filter

The complementary filter gives the simplest way to fuse the accelerometer and the gyroscope data to obtain accurate and responsive pitch/roll/yaw attitude outputs. It consists of a common low-pass filter for the accelerometer and a high-pass filter for the gyroscope. Example for driving device using complementary filter is on Figure 4.

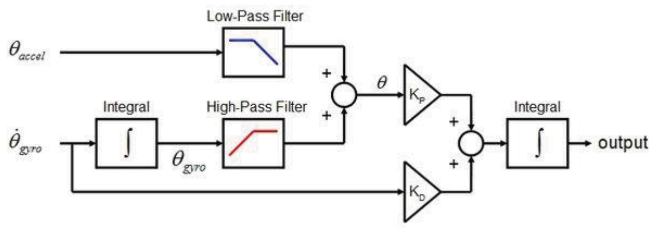


Fig. 4. Complementary filter example

Input variables θ_{accel} and $\dot{\theta}_{gyro,}$ on Figure.4, are output values from MEMS sensor, specifically angle from accelerometer and angular velocity from gyroscope.

VI. CONSTRUCTION

The construction consists of parts shown on Figure 5. Communication between sensor and microprocessor is with I^2C or SPI bus. The I^2C bus (Inter-Integrated Circuit) has good utilization in case for more sensors connected on the same bus, and sufficient speed (on this case 400 kHz). The opposite SPI (Serial Peripheral Interface) is simpler and faster, but little bit complicated in multi-device communication. On multi communication are used more wires for identifying slave device or the Daisy-chain connection. Most of MEMS sensors work with voltage level from ~2 V to ~3.3 V, but for example Arduino board has logic levels 0 to 5 V. Voltage level must be converted. Level shifter schematic is on Figure 6.

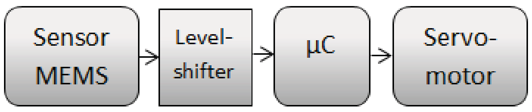


Fig. 5. Block diagram

The abstract block µC symbolizes the microcontroller or development board with optional type of microprocessor. Is assumed to output signal from microcontroller has minimal least 3 V for servo controlling. And there is possible a small servo connected directly to the microcontroller output pin.

VII. SCHEMATIC

Schematic bellow show typical operating circuit.

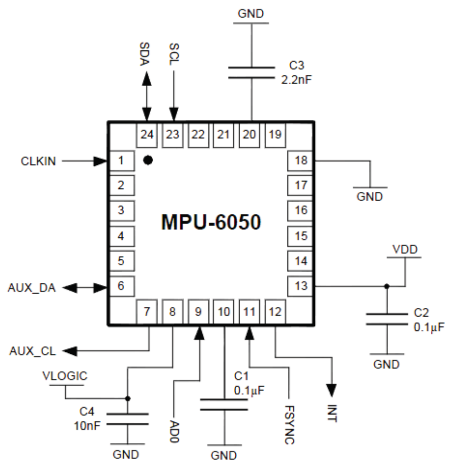


Fig. 6. Typical sensor circuit

As mentioned, it is necessary to adjust voltage levels for each line of I^{2}C – SCL (Serial Clock) and SDA (Serial Data). Easy is using BSS138 FET transistor as shown Figure 5. With this wiring were any problems and it can reaches bus speed to 400 kHz. This is also the maximum sensors communication interface speed.

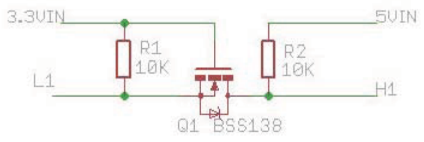
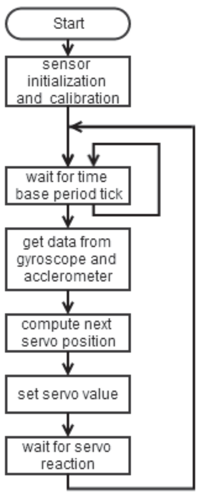


Fig. 7. Bidirectional level converter.

The hobby servo contains inside amplifier and power part, therefore is possible connect it by microprocessor directly. Controlling pulse has period 50 Hz with signal length between 500 µs to 2500 µs. Pulse with width 1500 µs is equal to middle position of servo.

VIII. Control and Algorithm

The basic main algorithm might look as follows on Fig 8.



Start
sensor initialization and calibration
wait for time base period tick
get data from gyroscope and acclerometer
compute next servo position
set servo value
wait for servo reaction

Fig. 8. Control algorithm

The second block, after start block (power on), is initialization and calibration. It is very important part because it should firstly be set the MEMS sensitivity, range of output data, and then calibration the accelerometer and gyroscope. The calibration consists in measuring data in in the zero position of the sensor. Theoretically values may be z_{acc}= 1 g, x_{acc}, y_{acc} = 0 g; and gyroscope x_{gyro}, y_{gyro}, z_{gyro} = 0 deg/s. Real values have random numbers. Therefore is needed calibration, for example, make measurement of 1000 values and make average from them. Thus we get calibration data. These data must be subtracted for obtaining the real data.

A. Simple servo position computing

The simplest servo position computing in one axis is in equation (1).

$$servoPositionX = servoPositionX - (differenceX * sensitivity) \quad (1)$$

The variable "*servoPositionX*" is value of servo pulse width; sensitivity is size of reaction and also speed of positioning. Variable "*differenceX*" is sensor output. For example if full scale range is set to 16g, sensitivity by 16 bit A/D converter in two's complement format is 2048 LSB/g and g-value is *sensor value*/2048.

The Sensor sends data after each time base tick. There must be defined boundary of signal, where servo can make correction or nothing (dead zone). If it was not defined, servo will correct each small signal and error noises, and stability would be disrupted.

B. Servo position computing using complementary filter

Computing position vector using two accelerometer values is on equation (2).

$$\theta_a = \arctan\left(\frac{a_y}{a_x}\right) \quad (2)$$

If a_y and a_x are normalized acceleration values in range [–1g; +1g], θ_a is the tilt angle in degrees.

$$\theta_g = \theta_g + (\omega_x - \omega_{x0}) \cdot S \Delta T \quad (3)$$

Angular displacement θ_a (in degrees) from gyroscope for one axis data are counting according formula (3). Gyroscope angular velocity is ω_x, ω_{x0} is zero - rate level finding in calibration, S is sensitivity and ΔT is sampling time interval. The final tilt angle from the complementary filter will look like the following

$$\theta = \theta'_g + \theta'_a = \beta \cdot \theta_g + (1 - \beta)\theta_g \quad (4)$$

Constant β is between 0 and 1. Good results are by the value 0.95.

$$\theta = 0.95 \cdot \theta_g + 0.05 \cdot \theta_a \quad (5)$$

If the accelerometer and gyroscope data are sampled at 100Hz, then the time interval ΔT is 0.01 second. So the time constant of the complementary filter is

$$\tau = \frac{\beta \cdot \Delta T}{1-\beta} = \frac{0.95 \cdot 0.01}{1-0.95} = 0.19\, sec \qquad (6)$$

IX. CONCLUSION

At this moment the experiment fulfills requirements of the communication testing, carrying out connection, control and timing, measuring and filtering MEMS sensor data. The resulting positioning inaccuracies are due to the use of a hobby servo, whose motion is not smooth enough, since their control characteristic contain dead band. It means, a small signal change does not make change of the servo arm position. But, still it manages to stabilize the platform and to hold an object on the platform. For servo position computation two types of control were used – a simple computing test and a complementary filter. Improvements can be made by using an alternative method to detect the platform position, for example an optical-camera system, to apply stronger servos and a direct motor control for eliminating the dead band. For the program creation were used programming tools as Arduino IDE v.1.0.5, AVR Studio v.4.18 and IAR Embedded Workbench for ARM v.6.40.

ACKNOWLEDGMENT

This work is a part of study on the project APVV–0000–12 Computer aided design and development of robust nonlinear controllers.

REFERENCES

[1] R. Faragher "Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation" IEEE Signal processing magazine, Sep. 2012, pp. 128-132.

[2] S. Kucuk and Z. Bingul. Robot Kinematics: Forward and Inverse Kinematics, Industrial Robotics: Theory, Modelling and Control, Sam Cubero (Ed.), ISBN: 3-86611-285-8, 2006, pp. 117-147.

[3] S. C. Narkar, S. R.Bhalekar, Tushar K. Nawge, Keshav H. Parab, P. V.Patil "Self Balancing Platform" International Journal of Computer Technology and Electronics Engineering (IJCTEE), vol 3, March-April 2013.

[4] S. Colby Fraser II, B. Jacobs, K. Santiago Jr. "Self Balancing Transportation Platform" unpublished.

[5] I2C-bus specification and user manual.
Rev. 5 — 9 October 2012
http://www.nxp.com/documents/user_manual/UM10204.pdf (17.1.2014)

[6] MPU6050 overview
http://www.invensense.com/mems/gyro/mpu6050.html (17.1.2014)

[7] MPU6050 specification
http://invensense.com/mems/gyro/documents/PS-MPU-6000A-00v3.4.pdf (17.1.2014)

[8] MPU6050 register map
http://invensense.com/mems/gyro/documents/RM-MPU-6000A.pdf (17.1.2014)