Angle Measuring by Mems Accelerometers

Submitted: 26th June 2011; accepted 23rd September 2011

Kamil Zidek, Miroslav Dovica, Ondrej Liska

Abstract:

This article contains the description of MEMS accelerometers implementation to device which is able to measure danger tilt. We can find out actual tilt in two basic axes X and Y, from -90° to +90°. Z Axis can only detect fall of device or in vehicle system very fast downhill grade during movement. For testing of the solution we select small mobile robotic carriage. Hardware and software part of solution are described. Because data from sensor are in raw format from analog MEMS Accelerometer, we use free C# library with Kalman Filter implementation to remove signal error. We can acquire next information from sensor data for example movement's trajectory in X/Y axis (Cartesian system) and actual speed in all three axes. Fast alarm is provided by RGB led diode (red color is dangerous tilt.

Keywords: *mems, Kalman filter, control.*

1. Introduction to MEMS Sensors

Shortcut MEMS means micro electromechanical systems, marks mechanical and electromechanical construction of very small dimensions, and technologies used for their preparation too. MEMS technology is based on many tools and methods, which are used for creating very small structure with dimension of couple micrometers. An important part of technology was takeover from production of Integrated circuit (IC technology). Almost all of these devices are based on a silicon substrate. MEMS structures are realized from thin layer. There are produced by photo lithographic methods. Some other methods also exist, but they aren't derivate straight from technology of IC. There are three basic steps of operation in MEMS technology for layer applying to silicon material to substrate. Process of MEMS is usually a structured sequence of this operation for creating real application. Real device, then, contains central unit for processing of data (microprocessor), and some other mechanical part which compose unit named micro sensor too [4].

2. MEMS accelerometers

One of usual application for MEMS is sensor for measuring of acceleration. This MEMS sensor is usually named Accelerometer. They are divided to one-, two- or three Axes. Measuring of acceleration is possible to use in electronics and robotics for measuring: acceleration, deceleration, tilt, rotation, vibration, collision (crash) or gravitation. Accelerometers are used in many devices, special equipment and personal electrotechnics, for example:

- robots and automated devices with balancing function (segway),
- controls with tilt measuring,
- Auto pilots of aeroplanes,
- car alarm systems,
- car crash detection (used in airbag system),
- monitoring of human movements (virtual reality gloves).

Example of MEMS microstructure sensor magnified by microscope is displayed in Fig. 1 on the left side, right is displayed measuring principle.

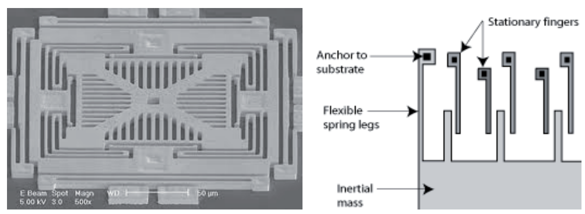

Fig. 1. MEMS sensor and principle of microstructures

Older accelerometers had big dimension and were very expensive. The construction was created from standard metal parts, springs and PCB. That was reason why at that time accelerometers were not used in electronics nor robotics. This situation was changed thanks to progress in MEMS technology. MEMS technologies reduce the price, energetic consumption and dimensions. Main usability is measuring of acceleration in three Axes: X – forward/backward, Y – left/right, Z – up/down. For mobile robotics we can use this sensor for measuring acceleration or deceleration by movement front and back, second Axis for change direction of movement right or left, and third Axis for fall detection of device. Second method of usability is measuring of device tilt based on simple mathematics. Figure 2 shows MEMS Axis configuration and principle of tilt measuring with this sensor along y axis.

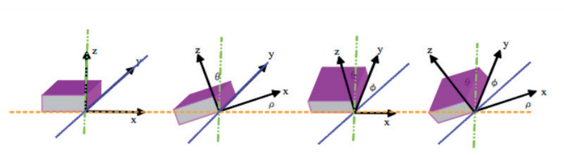

Fig. 2. MEMS sensor: axis configuration, principle of tilt measuring

Output information from accelerometer is voltage which depends on movement or tilt of sensor in space. A static characteristic of sensor is not exactly linear. For common application we can this nonlinearity omit. The acceleration is usually in MEMS application measured in G unit. Expression "1 g = 9,80665 m/s2" means, that for every second, which passed the speed change will be 9,80665 meters for second. That is approximately speed 35.30394 km/h. The three Axis accelerometer can get null G on every Axis, if is in ballistic trajectory known as inertial or free fall. If we turn the accelerometer to 90 ° the output from one Axis will be exactly +1 g. In this situation, accelerometer measuring gravitation Force and can be in static position. Described characteristics for analogue MEMS sensor is depicted in Fig. 3. [1]

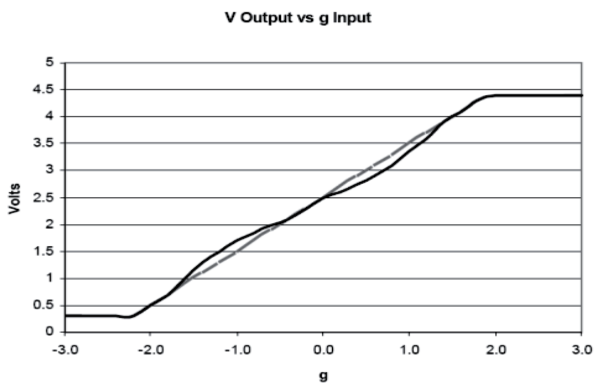

Fig. 3. Characteristics of MEMS accelerometer sensor with nonlinearity [2]

Example of block scheme sensor connection to user application is displayed in the Fig. 4. Additional LCD display connected straight to microcontroller enabling testing of application without Computer necessity.

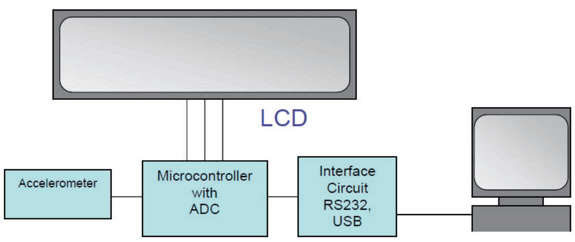

Fig. 4. Block scheme of complete application based on MEMS accelerometer sensor

Sensitivity of measured values depends on sensor G range (most precise we acquire if sensor is set to ±1g). A disadvantage is that we cannot measure the higher values of acceleration. Common sensors are produced to ±5g and it is possible to switch between ranges during application activity. Computations of tilt angles θ are realized thru basic mathematics and goniometric function. V_out is actual value of voltage; V_offset is voltage by 0 g. Sensitivity of sensor is defined by technical documentation. In math is necessary find out positive or negative acceleration according to offset value. Datasheet math count according this:

$$V\_OUT=V\_OFF+ \Delta V/\Delta g*1g*\sin\theta \quad (1)$$
$$\theta=\arcsin(V\_OUT- V\_OFF)/(\Delta V/\Delta g) \quad (2)$$

where:
V_OUT – output of accelerometer (V) from ADC
VOFF – acceleration 0 g offset
ΔV/Δg – sensitivity
1 g – world gravitation
θ – tilt angle

Our values are counted according changed math, because we don't know max and min values for actual accelerometer. This math get extreme values during accelerometer operation.

$$incr = 180 * (H\_max - H\_min) \quad (3)$$
$$\theta = incr * (H\_nam - H\_min) \quad (4)$$

where:
H_max, H_min – initial value of accelerometer extreme
H_nam – actual accelerometer value

3. Tested hardware platform

Introduced solution was tested on mobile computer with open source application in programming language C#. A prototype board contains Accelerometer MMA7341L (analog) and accelerometer MMA7455 (digital) from Freescale. Currently there is active only analog Accelerometer. Microcontroller computes values of voltage for all Sensor Axis with help of three 10 bits ADC converters. Data are coded to frames (9 bytes as string $XXYYZZ1310). Every axis has value coded to two bytes (Low and High 8 bites).

First method of accelerometer communication is only for debug the application. Sensor is connected straight to PC. Data are sent thru serial line to serial port of PC.

For implementation to mobile robot is used USART interface without UART/RS232 Transducer and communicate straight with High Level control system based on AT91 control board with Linux Embedded OS. These serial data are transferred to TCP packet thru ser2net command line application. Data are sent next thru wife interface to C# application. Block diagram of testing debug solution and mobile control system implementation is displayed in Fig. 5. Figure 6 shows is first prototype of sensor without RS232/USB transducer.

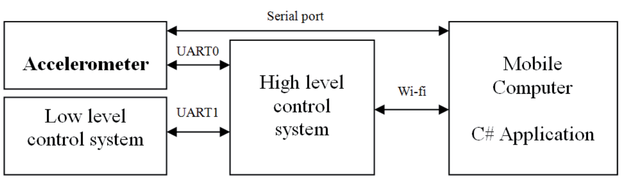

Fig. 5. Block diagram of connection sensor to testing mobile control system

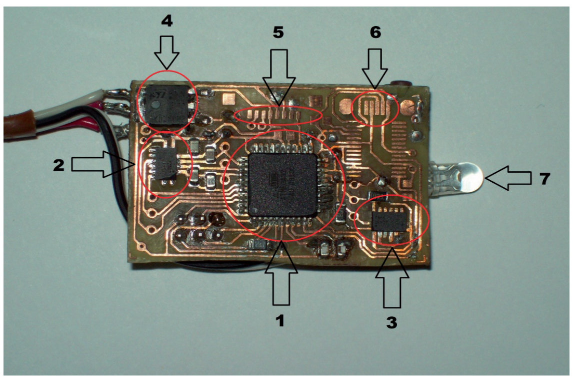

Fig. 6. Hardware of accelerometer. 1 – microcontroller; 2 – accelerometer MMA7341L (analog); 3 – accelerometer MMA7455 (digital); 4 – voltage regulator LF33CDT; 5 – I2C bus for LCD BO1602D; 6 – USB connector; 7 – RGB LED diode.

4. Software platform implementation

Software solution is based on an open source C# application, which is currently implemented to mobile solution Graphical Interface of solution is displayed in Figure 7. Left is displayed 2D graphics, tilt in x-Axis, left 3D graphics tilt in all three axis X,Y,Z. All values of real time tilt are displayed in graphical interface in text edit boxes.

Basic value of danger tilt is set to value bigger than 40°. This value starts critical routine and block movement of mobile device to actual direction. Danger tilt value can be changed through graphical interface from 0–90°.

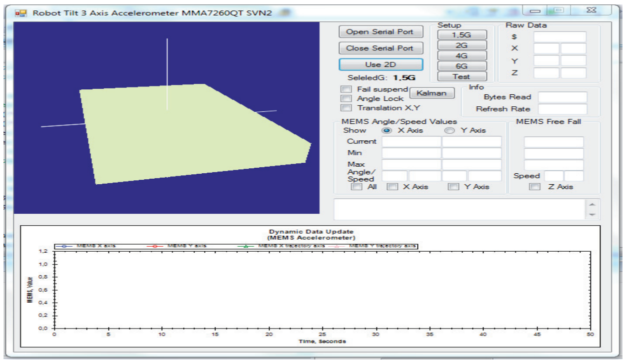

Fig. 7. C# application, 3D tilt X,Y Axis and configuration panel

5. Kalman filter implementation to smooth raw data

The Kalman filter is an efficient recursive filter that estimates the internal state of a linear dynamic system from a series of noisy measurements. The Kalman filter is used in sensor fusion and data fusion. Typically real time systems produce multiple sequential measurements rather than making a single measurement to obtain the state of the system. These multiple measurements are then combined mathematically to generate the system's state at that time instant. Acquired data from MEMS sensor are in raw form with many disturbances, white noise etc. For testing solution we implement free C# Math. NET Neodym (Signal Processing) [8] with Kalman filter function to desktop application. Graphical interface provides settings of three basic values of Kalman filtering r, T, q which is necessary for customizing filter for real application.

r – Measurement covariance

T – Time interval between measurements

q – Plant noise constant

Discrete Kalman Filter consists of two parts: prediction and update.

prediction:

$x(k|k-1) = F(k-1) * x(k-1|k-1)$ (5)

$P(k|k-1) = F(k-1)*P(k-1|k-1)*F(k-1) + G(k-1)*Q(k-1)*G'(k-1)$ (6)

update:

$S(k) = H(k)*P(k|k-1)*H'(k) + R(k)$ (7)

$K(k) = P(k|k-1)*H'(k)*S^{(-1)}(k)$ (8)

$P(k|k) = (I-K(k)*H(k))*P(k|k-1)$ (9)

$x(k|k)=x(k|k-1)+K(k)*(z(k)-H(k)*x(k|k-1))$ (10)

where:

S – Measurement covariance,

K – Kalman gain,

P – Covariance update,

x – State update,

F – State transition matrix,

G – Noise coupling matrix,

Q – Plant noise covariance matrix,

H – Measurement model,

R – Covariance of measurements,

I – Matrix identify,

z – Measurements of the system.

Figure 8 shows graph of actual values when MEMS sensor is stand statically on the ground (blue plotline). Black plotline shown filtered value cleared from errors and noise from ADC transduction. There is used for testing application only 1D Kalman filter for filtering only actual acceleration value. Next extension will be implementation of 2D or 3D filter for all three Axes.

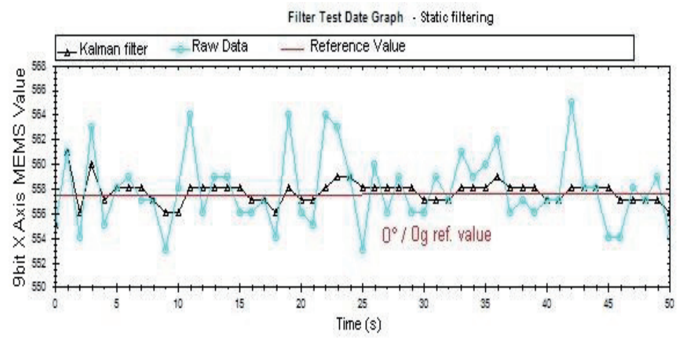

Fig. 8. Kalman filtering for raw accelerometer data in static position

In the Fig. 9 is displayed data from accelerometer during tilt to 90° to one side, next to static position and then tilt to opposite side. Reference signal is red plotline. Black line is Kalman filtered value.

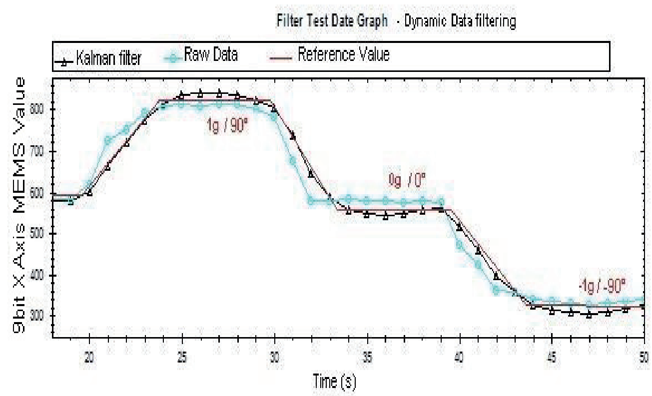

Fig. 9. Dynamic data from MEMS Accelerometer sensor with Kalman filtering

We experimentally find out constants for kalman filter with compromise of minimal displace during dynamic and static operation: r = 30.0, T = 2.0, q = 0.1.

There is one problem in setup filter, when there is very fast acceleration and deceleration. This situation can occur when the real device fall or crash to the obstacle. We can avoid this situation by setting adequate value to danger alarm tilt and implementation of obstacles sensor detection (infra or ultrasonic) to mobile solution.

6. Conclusion

Introduced measuring solution is implemented to mobile device. Actual possibilities are measuring of tilt device – 90° to 90°. You can select bound angle for start indication of danger device tilt with next visual or sound alarm. We can improve precision data from MEMS sensor by using 12 bit ADC but then is necessary change the microcontroller. Next idea can be change of Accelerometer with digital I2C output, which removes error generated by ADC conversion. We are computing next values from acquired data for example: trajectory, deceleration, average and actual speed.

Next work on this solution will be an implementation of Kalman filter to program of MCU firmware and display actual angle value and alarm on LCD display. This remove testing mobile computer from actual solution and application will be small and compact device. This researched accelerometer device will be used in rehabilitation system as safety circuit to monitor extreme acceleration and deceleration for fast action to stop device.

Acknowledgements

The research work is supported by the Project of the Structural Funds of the EU, Operational Program Research and Development, Measure 2.2 Transfer of knowledge and technology from research and development into practice: Title of the project: Research and development of the intelligent non-conventional actuators based on artificial muscles ITMS code: 26220220103

AUTHORS

Kamil Zidek*, Ondrej Liska, Miroslav Dovica – Technical University of Kosice, Faculty of mechanical engineering, Department of Biomedical engineering automation a measuring, Košice, 042 00, Slovakia, kamil.zidek@tuke.sk, ondrej.liska@tuke.sk, miroslav.dovica@tuke.sk

*Corressponding author

References

[1] Tuck K., "Tilt Sensing Using Linear Accelerometers", Application Note, AN3461, *Rev 2*, 06/2007.http://www.freescale.com/files/sensors/doc/app_note/AN3461.pdf

[2] Clifford M., Gomez L., "Measuring Tilt with Low-g Accelerometers", Application Note, AN3107, Rev 0, 05/2005,http://www.freescale.com/files/sensors/doc/app_note/AN3107.pdf

[3] "What is MEMS Technology?" https://www.memsnet.org/mems/what-is.html, Online, cit. 8.2.2010

[4] Johnson C. D., "Accelerometer Principles", *Process Control Instrumentation Technology*, Apr 14, 2009, 0-13-441305-9.

[5] Zidek K: "Open robotics control system", Technical University Kosice, Online, www.orcs.sebsoft.com

[6] Saloky T., Piteľ J., Vojtko I., "Control systems design with reliability defined in advance". In: *Proceedings of the 1st IFAC Workshop on New Trends, Design of Control Systems*, Smolenice, Slovakia, 7th–10th September 1994, pp. 404–407.

[7] Zidek K., *MEMS Accelerometer SVN*, Google code, 2010. http://code.google.com/p/orcs/source/browse/#svn/MEMS_Accelerometer_SVN2

[8] Christoph Rüegg, Math.NET Neodym 2008 February Release, v2008.2.2.364, "http://www.mathdotnet.com/downloads/Neodym-2008-2-2-364.ashx?From=NeodymCurrentRelease"