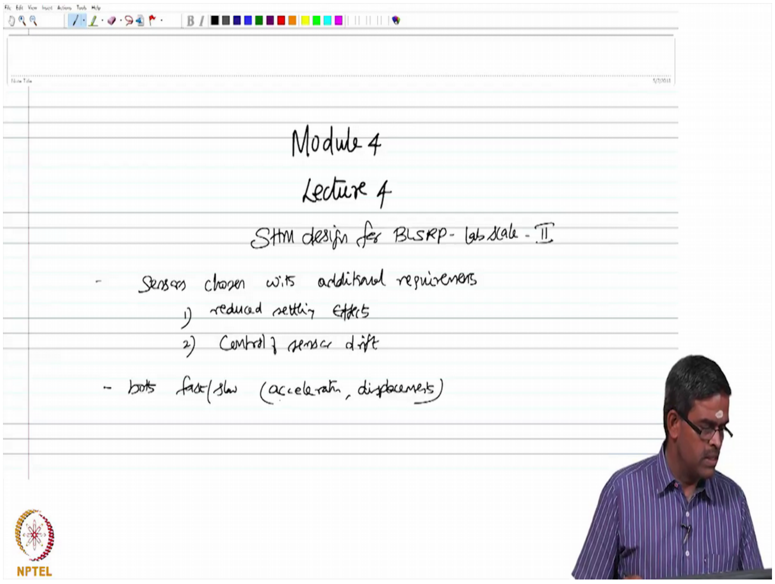
Structural Health Monitoring (SHM)
Prof. Srinivasan Chandrasekaran
Department of Ocean Engineering
Indian Institute of Technology, Madras

Lecture - 71
SHM design- 2 for BSLRP-Part 1

Friends, welcome to the 4th lecture in module 4.

(Refer Slide Time: 00:24)



Module 4

Lecture 4

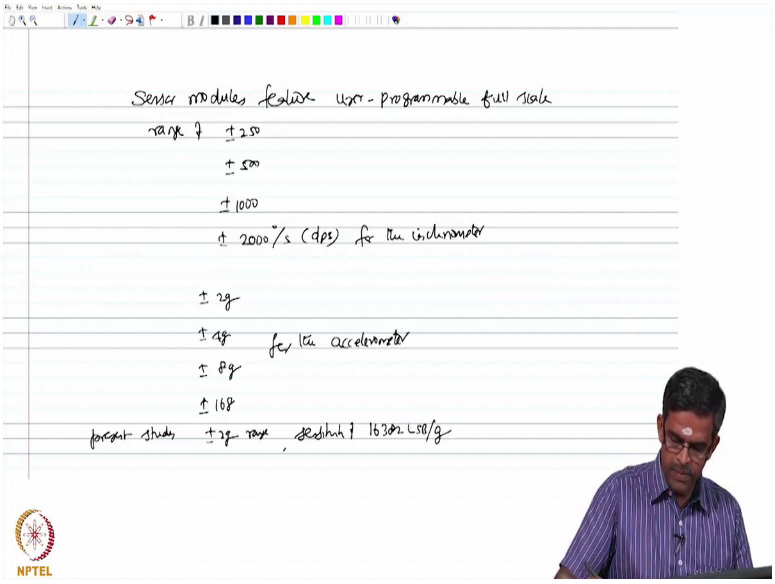
SHM design for BLSRP - lab scale - II

- Sensors chosen with additional requirements
  1) reduced settling effects
  2) Control of sensor drift
- both fast/slow (acceleration, displacements)

Here we are going to continue with the discussions on SHM design for BLSRP in the lab scale which we discussed in the last lecture. This is the second lecture in continuation with the previous one. We look back at the summary in the last lecture, we said that the sensors are now chosen with few additional requirements name the; it should have a reduced settling effect it should also have the control of sensor drift.

So, keeping this in mind, sensors have been chosen, we need to also measure both fast and slow acceleration and displacements, because the platform will exhibit both kinds of characteristics under the wave load.

(Refer Slide Time: 01:50)



Sensor modules feature user-programmable full scale range of ± 250

± 500

± 1000

± 2000°/s (dps) for the inclinometer

± 2g

± 4g for the accelerometer

± 8g

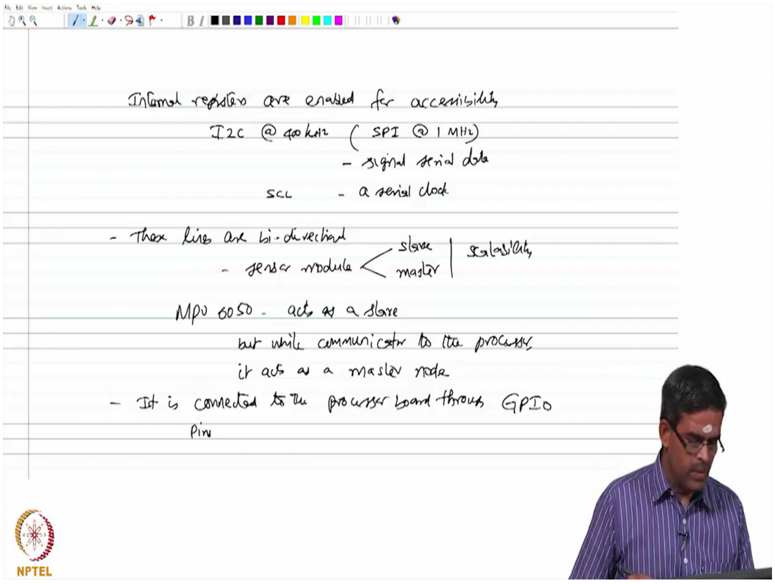
± 16g

present study ± 2g range, sensitivity of 16382 LSB/g

The sensor modules feature also user programmable full scale range of plus minus 250 plus minus 500 plus minus 1000 plus minus 2000 degrees per Celsius, the dps for the inclinometer and plus minus 2 g plus minus 4 g plus minus 8 g and plus minus 16 g for the accelerometer.

So, the present study focus on plus minus 2 g range for accelerometer with sensitivity of 16382 LSB per g.

(Refer Slide Time: 03:16)



Internal registers are enabled for accessibility

I2C @ 400 kHz (SPI @ 1 MHz)

- signal serial data

SCL - a serial clock

- These lines are bi-directional
  - sensor module < slave / master | scalability

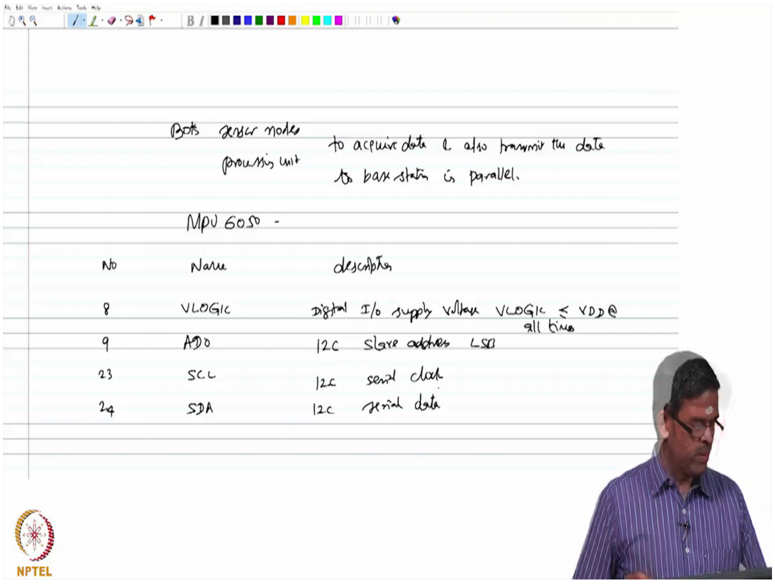
MPU 6050 - acts as a slave
but while communication to the processor
it acts as a master mode

- It is connected to the processor board through GPIO pin

Internal registers for memory or enabled for accessibility, we have used I 2 C at 400 kilo hertz, alternatively one can also use an SP 1 at 1 megahertz which is the signal serial data and a signal clock which is a serial clock, the present study uses SCL; a serial clock. Now these lines are bi directional. Therefore, the sensor module can either act as a slave or as a master. So, that is what we call scalability.

So, MPU 6050 generally acts as a slave, but while communicating to the processor, it acts as a master node. It is connected to the processor board through GPIO pins.

(Refer Slide Time: 05:36)



Both sensor nodes
processing unit
to acquire data & also transmit the data
to base station in parallel.

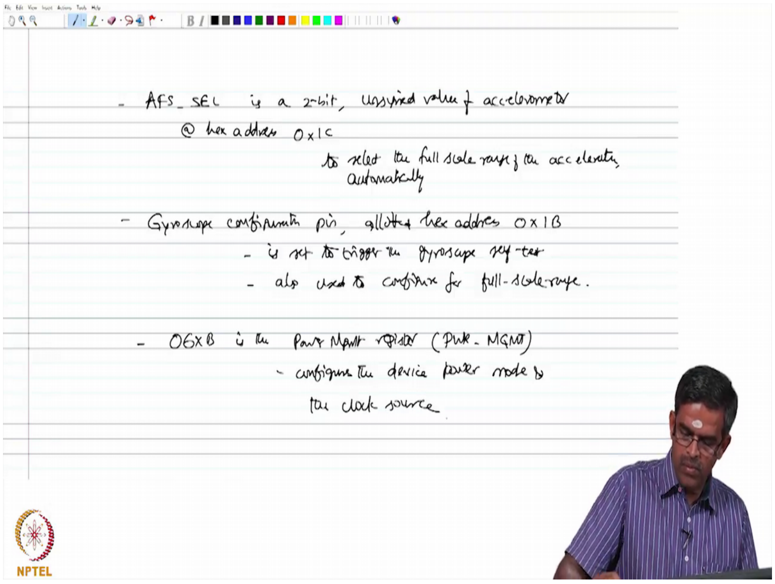
MPU 6050 -

No	Name	description
8	VLOGIC	Digital I/o supply voltage VLOGIC ≤ VDD @ all time
9	AD0	12C slave address LSB
23	SCL	12C serial clock
24	SDA	12C serial data

So, which enables both sensor nodes and the processing unit to acquire the data and also transmit the data to base station in parallel the description of MPU pin which is 6050 is given below. So, there are many pins available, let us see the name number and the description of the pin which is being used in the present study for 8, 9, 23 and 24 the name is VLOGIC, AD0, SCL and SDA. The description could be; this is a digital input output supply voltage.

So, VLOGIC must be less than or equal to VDD at all times ok; that is the condition for operation, this is 12C slave address LSB. This is correspond to LC serial clock.

(Refer Slide Time: 07:34)



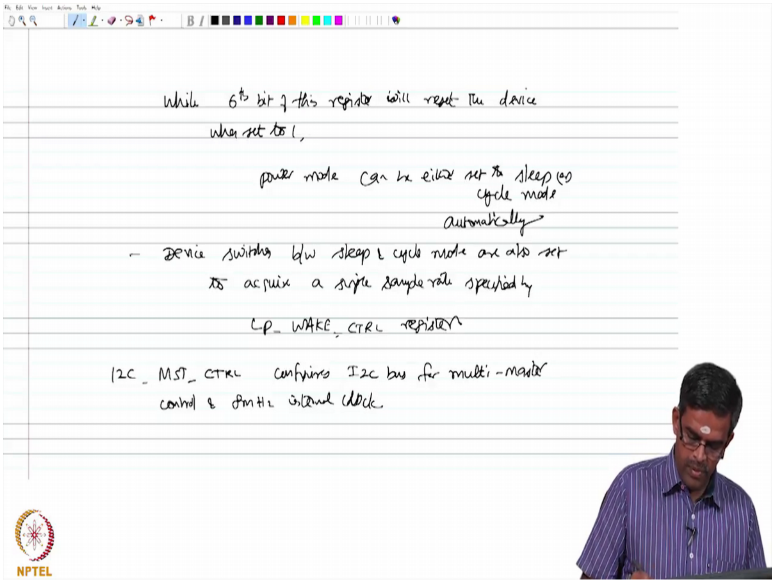
- AFS_SEL is a 2-bit, unsigned value of accelerometer
  @ hex address 0x1C
  to select the full scale range of the acceleration, automatically

- Gyroscope configuration pin, allotted hex address 0x1B
  - is set to trigger the gyroscope self-test
  - also used to configure for full-scale-range.

- 06xB is the Power Mgmt register (PWR_MGMT)
  - configures the device power mode &
    the clock source

As I said in the earlier point, this is 12 C serial data in addition AFS_SEL which is a 2 bit unsigned value of accelerometer had the hex address 0 X 1 C is used to select the full scale range of the acceleration, automatically, the gyroscope configuration pin is allotted the hex address 0 x 1 B which is set to trigger the gyroscope self test and also used to configure for full scale range the 0 6 X B is the power management register which is actually P MGMT which configures the device power more and the clock source while the sixth bit of this register, we will reset the device when set to one.

(Refer Slide Time: 09:48)



While 6th bit of this register will reset the device
when set to 1,

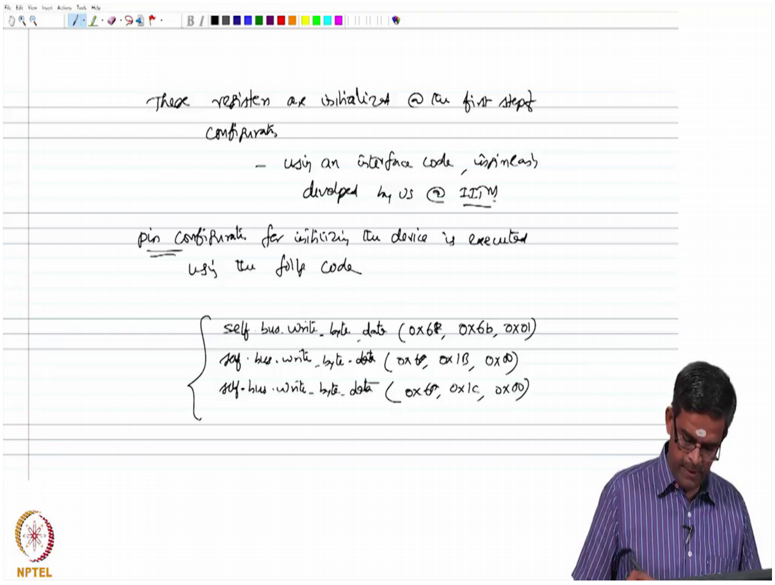
power mode can be either set to sleep (or)
cycle mode
automatically

- Device switches b/w sleep & cycle mode are also set
  to acquire a single sample rate specified by
  LP_WAKE_CTRL register

I2C_MST_CTRL configures I2C bus for multi-master
control & 8MHz internal clock.

So, by this way, power mode can be either set to sleep or cycle mode automatically. So, you can save lot of power the device switches between the sleep and cycle mode or also set to acquire a single sample rate specified by LP WAKE control register, the 12 C MST control configures the I 2 C bus for the multi master control and eight megahertz internal clock.

(Refer Slide Time: 11:56)



These registers are initialized @ the first step of configuration

- using an interface code, indigenously developed by us @ IITM

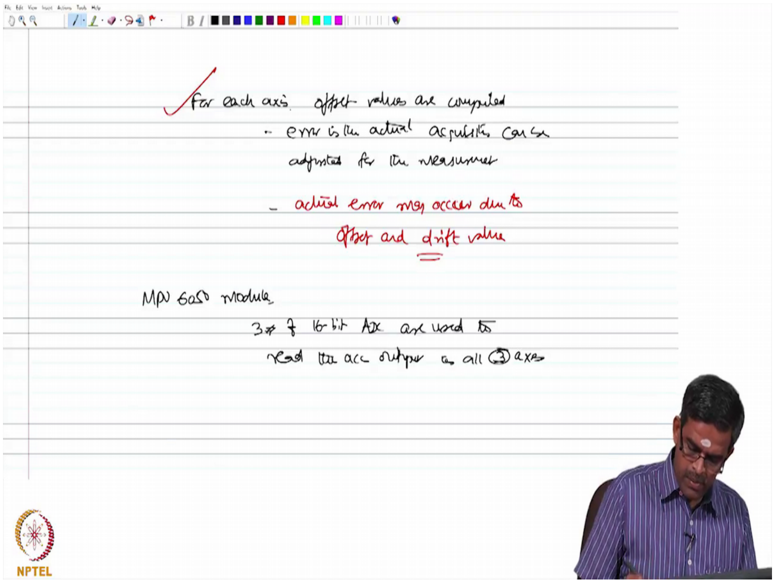
pin configuration for initializing the device is executed using the follg code

```
self.bus.write_byte_data (0x68, 0x6b, 0x01)
self.bus.write_byte_data (0x68, 0x1B, 0x00)
self.bus.write_byte_data (0x68, 0x1c, 0x00)
```

These registers are initialized at the initial step at the first step of configuration, this is done by using an interface code which is indigenously developed by us at IIT, Madras. The pin configuration for initializing the device is executed using the following code which is self bus write byte data 0X68 0X6B 0X01.

Similarly, self bus write byte data 0X68, 0X1B, 0X00, similarly self bus write byte data 0X68, 0X1C and 0X00 or used for configuring the pin initially for each axis offset values are calculated.

(Refer Slide Time: 14:19)



For each axis offset values are computed

- error is the actual acquisition can be adjusted for the measurement

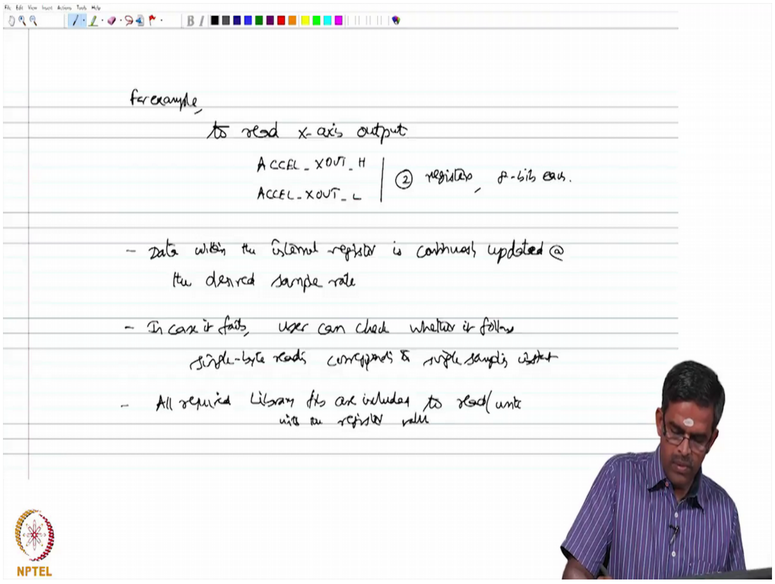
- actual error may occur due to offset and drift value

MPU 6050 module

3# of 16 bit ADC are used to read the acc output in all 3 axes

So that the error in the actual acquisition can be adjusted for the measurements; the actual error may occur due to the offset and drift value. So, they are adjusted in the beginning itself in the pin MPU 6050 module, 3 numbers of 16 bit analog digital converters are used to read the accelerometer output in all 3 axis.

(Refer Slide Time: 15:56)



For example

to read x-axis output

ACCEL_XOUT_H
ACCEL_XOUT_L | 2 registers, 8-bits each.

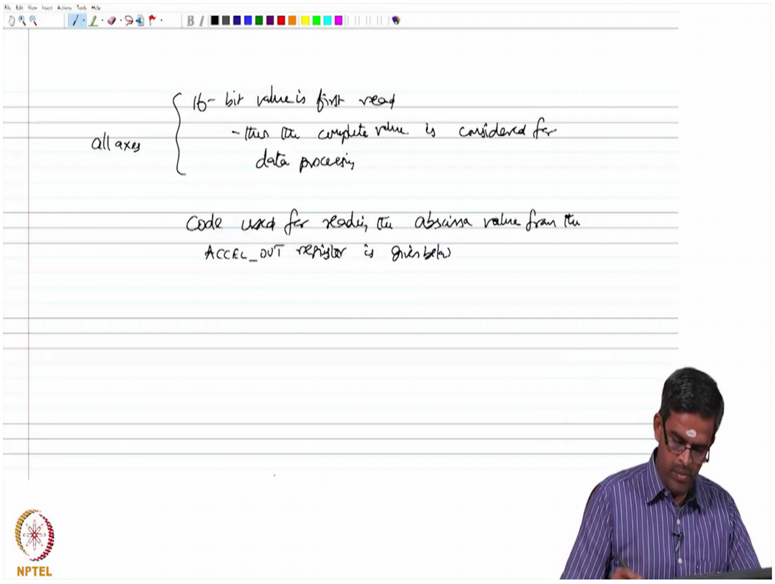
- Data within the internal register is continuously updated @ the desired sample rate

- In case it fails, user can check whether it follows single-byte reads corresponds to single sample is used

- All required library files are included to read/write with the register value

For example, to read X axis output A C C E L ACCEL, X OUT H and A C C E L ACCEL X OUT L are the two registers which will have 8 bits each. Now data within the registry is continuously updated at the desired sample rate, in case, it fails, user can check whether it follows a single byte reading corresponding to single sampling instant, all the required library files are included to read and write with the register values 16 bit value is first rate.

(Refer Slide Time: 18:10)



all axes

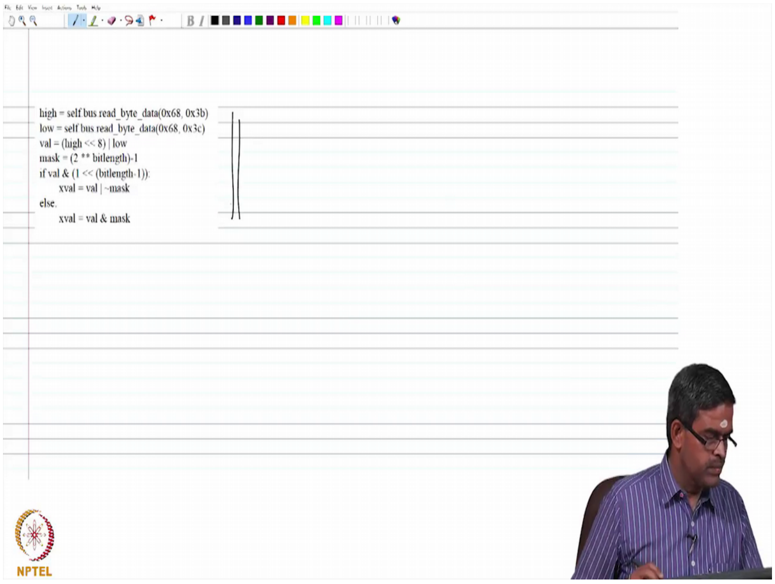
16- bit value is first read

- then the complete value is considered for data processing

Code used for reading the abscissa value from the ACCEL_OUT register is given below

There is a sequence of reading, and then the complete value is considered for data processing this is being done for all access ok. Now, interestingly the code used for reading the abscissa value from the acceleration output register is given below.

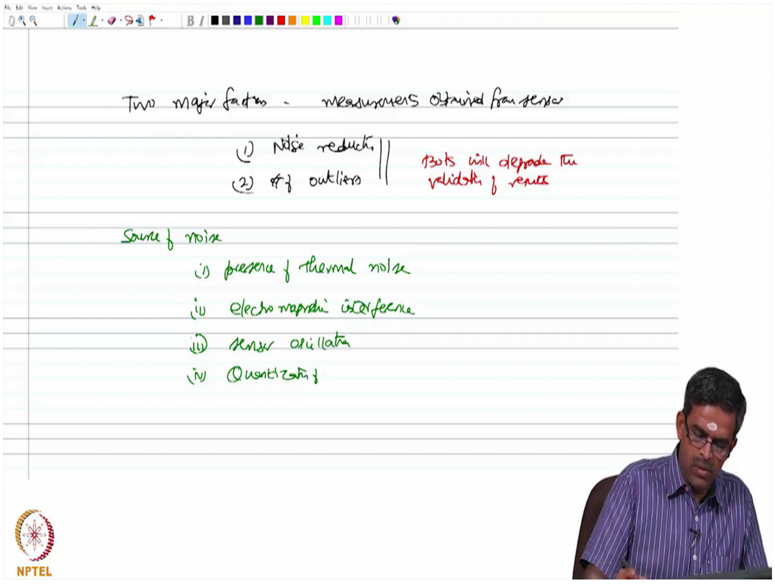
(Refer Slide Time: 19:25)



```
high = self.bus.read_byte_data(0x68, 0x3b)
low = self.bus.read_byte_data(0x68, 0x3c)
val = (high << 8) | low
mask = (2 ** bitlength)-1
if val & (1 << (bitlength-1)):
        xval = val | ~mask
else:
        xval = val & mask
```

One can see, this is the code used for reading the abscissa value for the acceleration output.

(Refer Slide Time: 19:39)



Two major factors - measurements obtained from sensor

(1) Noise reduction || Both will degrade the validation of results
(2) # of outliers ||

Source of noise

(i) presence of thermal noise
(ii) electromagnetic interference
(iii) sensor oscillation
(iv) Quantization

Interestingly, two major factors are considered while dealing with the sensor measurements. One is the noise reduction and second is the number of outliers because both will degrade the validation of results. So, they should be checked. Now interestingly, the source of noise may arise from various factors. One presence of thermal noise coming from the temperature variation, it can be electromagnetic interference, it can also be due to sensor oscillation, if it is not firmly fixed it can be due to quantization of the noise.