Project Title	**Vehicle Localisation Using GPS and IMU**
By	**K.Prasanna** **3rd Year, B.Tech Mechanical** **National Institute of Technology(NIT), Rourkela, India**
Guide	**Prof Marcelo Ang Jr**
Asst Guide	**Dr Sharon Ong**
Institution	**National University of Singapore(NUS)**
Deptt	**Control and Mechatronics, Mechanical Engg**
Duration	**18 March- 16 July,2009**
Date	**15-07-2009**

Contents

Acknowledgements

I would like to express my sincere thanks to Prof Marcelo Ang Jr for giving me this opportunity to work under him. I would also like to thank Dr Sharon Ong for her valuable advice and guidance at every stage of this project which helped complete this internship. I would also thank Mr James Fu and Dr Koh Niak Wu for their suggestions and assistance.

Introduction

Localisation systems are required in a multitude of fields ranging from vehicle tracking, guidance systems, driver aids in automobiles, etc. It is highly essential to keep track of the position and orientation of dynamic objects in innumerable applications and localisation systems are a must.

Background

Several sensors like accelerometers, gyros, odometers, gps receivers, imu, etc have been used for localisation applications. Several combinations of these sensors have also been used with a suitable integration system. Filters like kalman filter have been popularly choice for sensor fusion. Still there is a great scope for the development of a system that is more accurate than the systems that have been developed so far.

Objective

Development of vehicle localisation system which integrates GPS and INS using kalman filter to give the position, orientation and other navigation parameters of the vehicle.

Localisation

Localisation is the determination of the position and orientation of the object under study. It is highly essential for dynamic objects to know its location and attitude at any point of time.

Sensors Used

1) San Jose Navigation FV M-11 GPS Receiver
2) SerAccel V5 Triple Axis Accelerometer
3) Crossbow DMU RGA300CA- IMU

San Jose Navigation GPS Receiver

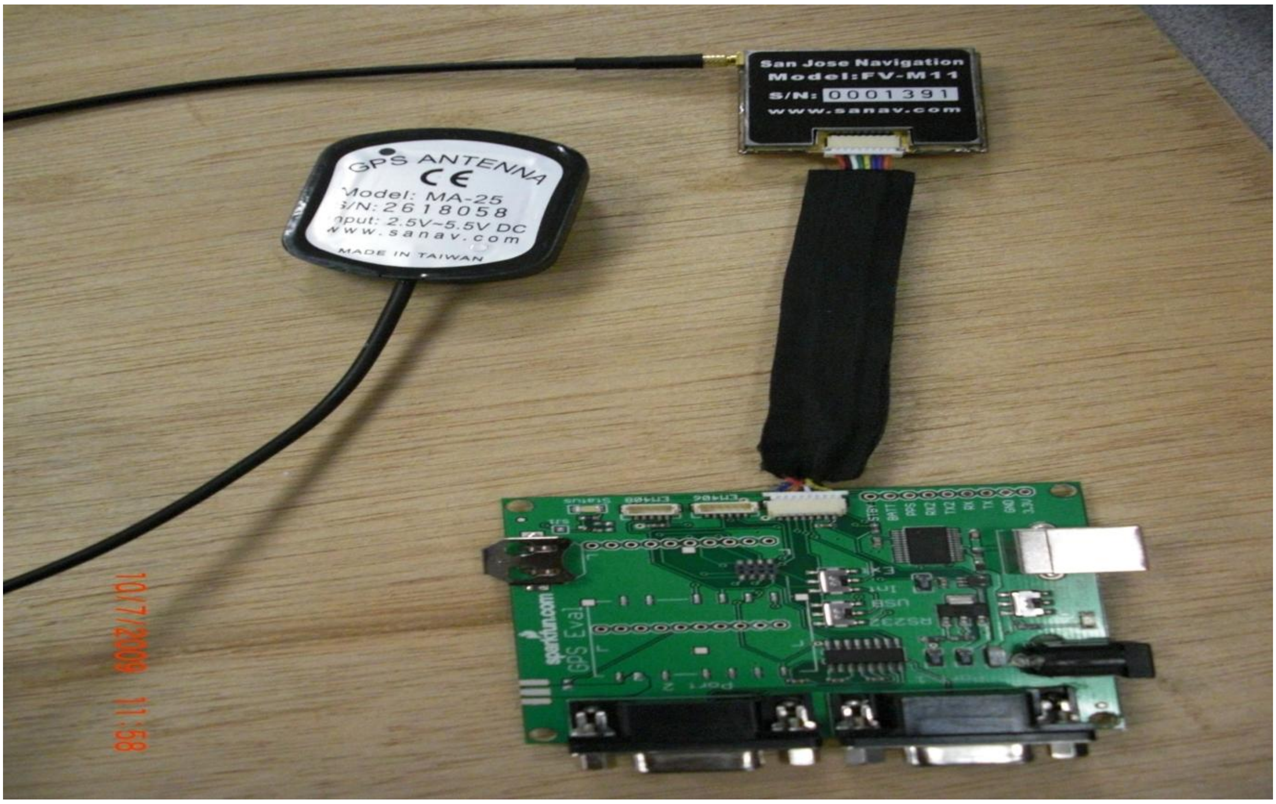
It is a GPS receiver that can be comfortably used for hand-held gps applications.

Specifications:-

<table>
<tr><td colspan="3">PHYSICAL CONSTRUCTION</td></tr>
<tr><td>Dimension</td><td colspan="2">L35.4mm*W35.4mm*H8.6mm</td></tr>
<tr><td>Weight</td><td colspan="2">11 grams</td></tr>
<tr><td>RF Connector</td><td colspan="2">MMCX Jack</td></tr>
<tr><td>Receiving frequency</td><td colspan="2">1575.42MHZ; C/A code</td></tr>
<tr><td>Connector</td><td colspan="2">8pin connector with 1mm pitch</td></tr>
<tr><td>Construction</td><td colspan="2">Full EMI Shielding</td></tr>
<tr><td colspan="3">ENVIRONMENTAL CONDITIONS</td></tr>
<tr><td rowspan="2">Temperature</td><td colspan="2">Operating: -30 ~ +80 Celsius</td></tr>
<tr><td colspan="2">Storage: -40 ~ +85 Celsius</td></tr>
<tr><td colspan="3">COMMUNICATION</td></tr>
<tr><td>Protocol</td><td colspan="2">NMEA V3.01</td></tr>
<tr><td>Signal level</td><td colspan="2">UART @ 2.8V * 2</td></tr>
<tr><td colspan="3">INTERFACE CAPABILITY</td></tr>
<tr><td rowspan="2">Standard Output Sentences</td><td>Default</td><td>RMC, GGA, GSV*5, VTG, GSA*5</td></tr>
<tr><td>Optional</td><td>GLL, ZDA</td></tr>
<tr><td>PERFORMANCE</td><td colspan="2"></td></tr>
<tr><td>Built-in Antenna</td><td colspan="2">Highly-reliable ceramic patch</td></tr>
<tr><td>Sensitivity</td><td colspan="2">-158dbm</td></tr>
<tr><td>SBAS</td><td colspan="2">1 channel (Support WAAS, GNOS, MSAS)</td></tr>
<tr><td>DGPS</td><td colspan="2">RTCM Protocol</td></tr>
<tr><td>Receiver architecture</td><td colspan="2">32 parallel channels</td></tr>
<tr><td rowspan="3">Start-up time</td><td>Hot start</td><td>1 sec. typical</td></tr>
<tr><td>Warm start</td><td>35 sec. typical</td></tr>
<tr><td>Cold start</td><td>41sec. typical</td></tr>
<tr><td rowspan="2">Position accuracy</td><td>Without aid</td><td>3.3 m CEP</td></tr>
<tr><td>DGPS (RTCM)</td><td>2.6 m</td></tr>
<tr><td>Velocity accuracy</td><td colspan="2">0.1 Knot RMS steady state</td></tr>
<tr><td>Update Rate</td><td colspan="2">1 ~ 5Hz</td></tr>
<tr><td>Power Supply</td><td colspan="2">3.3~5V +- 5%</td></tr>
<tr><td rowspan="4">Power Consumption</td><td>Acquisition</td><td>55mA</td></tr>
<tr><td rowspan="3">Tracking</td><td>42mA (first 5 minutes)</td></tr>
<tr><td>42mA (after 5 minutes)</td></tr>
<tr><td>34mA (after 20 minutes)</td></tr>
<tr><td>External Antenna</td><td colspan="2">GPS antenna with 2.8V power input</td></tr>
<tr><td>Baud Rate</td><td colspan="2">4800 bps (default) & 4800/9600/38400/57600/115200 bps are adjustable</td></tr>
</table>

Supporting Equipment:-

1) San Jose Navigation GPS Antenna
2) Sparkfun GPS Evaluation Board



GPS receiver, Antenna and Evaluation Board

GPS Message Processing

Once the gps receiver is switched on, it starts transmitting message strings which follow the NMEA Protocol.

NMEA Protocol

The NMEA protocol expresses the data in the format of ASCII. This is a standard format for GPS applications.

NMEA Output Messages

NMEA Record	Description
GGA	Global Positioning System Fix Data.
GSA	GNSS DOP and Active Satellites
GSV	GNSS Satellites in View
RMC	Recommended Minimum Navigation Information
VTG	Course Over Ground and Ground Speed
GLL	Geographic Position – Latitude / Longitude
ZDA	Time & Date

Here we use the GPGGA and GPVTG strings only.

GGA – GPS Fix Data

$GPGGA,gga1,gga2,gga3,gga4,gga5,gga6,gga7,gga8,gga9,gga10,gga11,gga12,gga13,gga14*hh<CR><LF>

Parameters	Descriptions	Notes
gga1	UTC time as position is fixed	hhmmss.sss: hh – hour; mm – minute; ss.sss – second
gga2	Latitude	ddmm.mmmmmm: dd – degree; mm.mmmmmm – minute (0 ~ 90 degree)
gga3	Latitude sector	N – North; S - South
gga4	Longitude	dddmm.mmmmmm: dd – degree; mm.mmmmmm – minute (0 ~ 180 degree)
gga5	Longitude sector	E – East; W - West
gga6	GPS quality indicator	0 – No fixed or invalid position 1 – SPS Position available 2 – Differential GPS (SPS)
gga7	Number of SVs used in position estimation	xx: 00 ~ 12
gga8	HDOP	xx.xx: 00.00 ~ 99.99
gga9	Altitude above mean sea level(geoid)	xx.xxx: 00.000 ~ 99.999
gga10	Unit for Altitude	M: meter
gga11	Geoidal separation	
gga12	Unit for geoidal separation	M: meter
gga13	Age of differential corrections	unit : second; null when DGPS is not used
gga14	Reference station ID (DGPS)	xxxx: 0000 ~ 1023
hh	Checksum	hex number (2 – character)
<CR><LF>	End of message	

VTG – Course over Ground and Ground Speed

$GPVTG,vtg1,vtg2,vtg3,vtg4 ,vtg5,vtg6,vtg7,vtg8,vtg9*hh<CR><LF>

Parameters	Descriptions	Notes
vtg1	Course over ground(degrees)	Referenced to true north (xx.xx degrees)
vtg2	Indicator of course reference	T – true north
vtg3	Course over ground(degrees)	Not Support
vtg4	Indicator of course reference	M – magnetic north
vtg5	Speed over ground	x.xxx knots
vtg6	Unit of speed	N – nautical miles per hour
vtg7	Speed over ground	x.xxx km/hr
vtg8	Unit of speed	K – kilometers per hour
vtg9	Navigation mode indicator	A – Autonomous mode (fix); D – Differential mode (fix); E – DR (fix); N – not valid
hh	Checksum	hex number (2 – character)
<CR><LF>	End of message	

Validation of GPS Message

The GPS message is valid only if all the parameters mentioned below are satisfied:-

- The checksum of the message is calculated(see code section for calculation of checksum method). If the checksum matches with the checksum information that comes towards the end of the GPS data strings.
- The 'Number of SVs Used' information must be greater than 4.
- 'GPS quality indicator' information must not be equal to 0.

If these conditions are satisfied, the message is validated.

Data Read From Strings

The required GPS information is extracted from these strings. Namely:-

- UTC Time
- Latitude
- Latitude Sector
- Longitude
- Longitude Sector
- GPS Quality Indicator
- Number of SVs used
- Altitude

- Unit of Altitude
- Course Over Ground
- Indicator of Course Reference
- Speed Over Ground
- Unit of Speed

Processing GPS information

From the latitude and longitude information, the distance travelled between any 2 points whose latitude-longitude coordinates are known can be calculated from the relation shown below, assuming the latitude and longitude coordinates of the first and second location are (lat1,long1) and (lat2,long2) respectively.

$$X\ distance(in\ meters) = (long2 - long1) * \text{Radius of Earth}$$

$$Ydistance(in\ meters) = (lat2 - lat1) * \text{Radius of Earth}$$

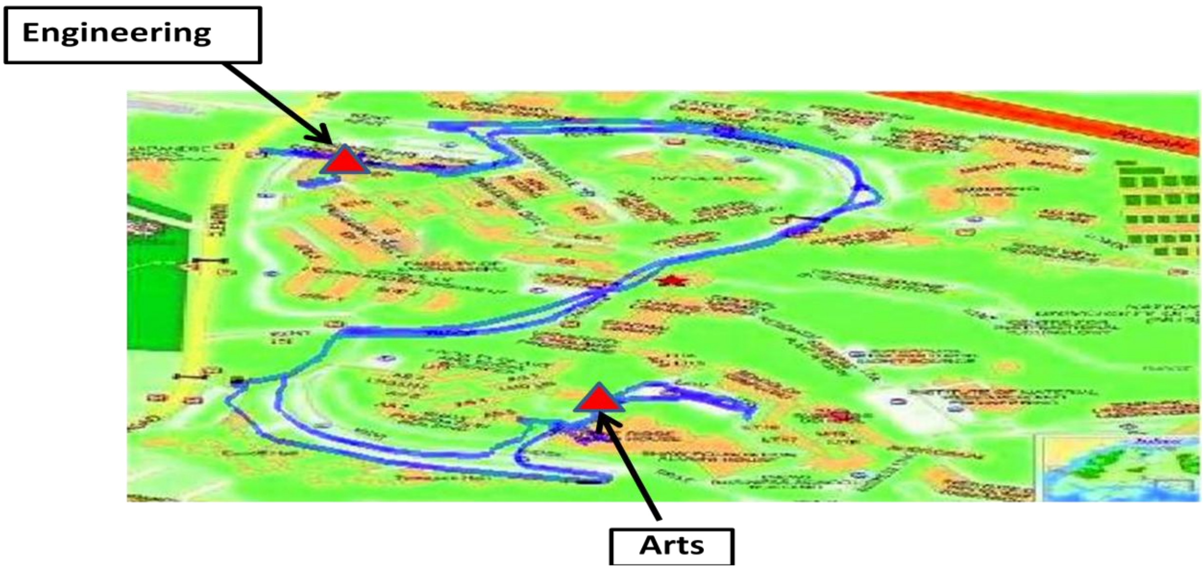
$\text{Radius of Earth} = 6378.14 * 1000\ meters$

Note:-

This relation hold good only when the distance travelled is in the order of a few kilometres.

Sample GPS Plot

The latitude-longitude data of the path from Engineering Auditorium (NUS) to Arts Faculty Canteen (NUS) was plotted over a map of NUS, we get the figure below:-



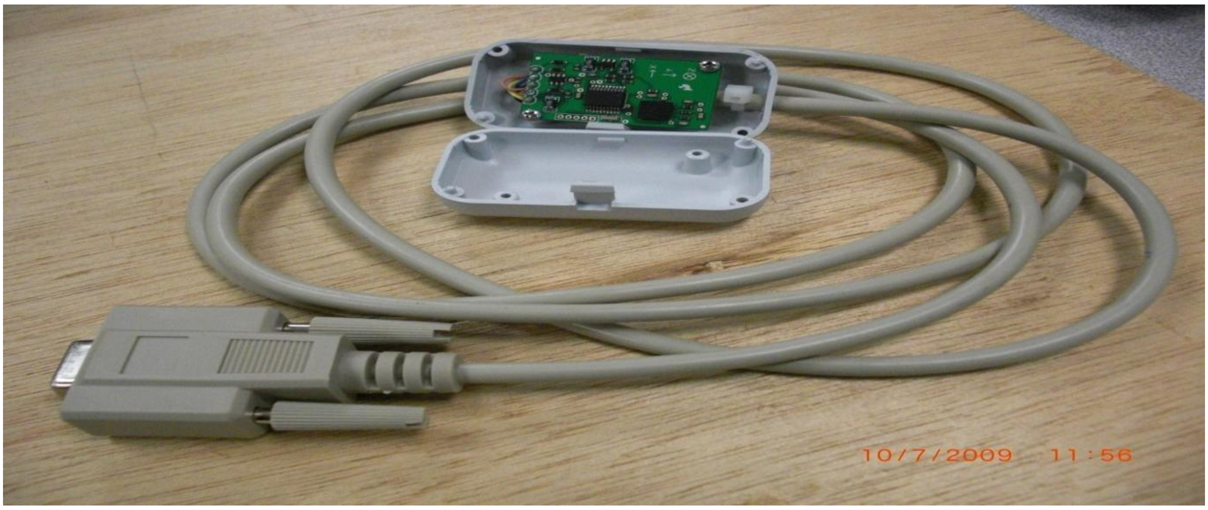
Sample GPS plot of path from Engineering Auditorium (NUS) to Arts Faculty Canteen(NUS)

Sample GPS plot of path from Engineering Auditorium (NUS) to Arts Faculty Canteen(NUS)

Singapore		
	Latitude:	1.3667 deg or 1° 22' 0" N
	Longitude:	103.8 103° 48' 0" E

SerAccel V5 Triple Axis Accelerometer

It is a 3-axis accelerometer. It is a plug and go device as it takes its power from an RS232 or any other USB to RS232 converter through which it is connected. It has a in-built calibration routine in its firmware through which it was calibrated before using it to take measurements.



Specifications		
Property	**Range**	**Explanation**
Baud rate	9600bps-57600 bps	Configurable
Sensor Range	1.5g,2g,4g,6g	Configurable
Frequency (Maximum)	Calculated Gravity=125 Hz Raw Output=195Hz	Configurable, maximum at 57600bps

Data Processing

Output Modes:-

- Calculated Gravity Values
- Raw Output Mode
- Binary Output Mode

The calculated gravity values were used to obtain the acceleration from the 3 axes.The acceleration output in the calculated gravity mode was of the form:-

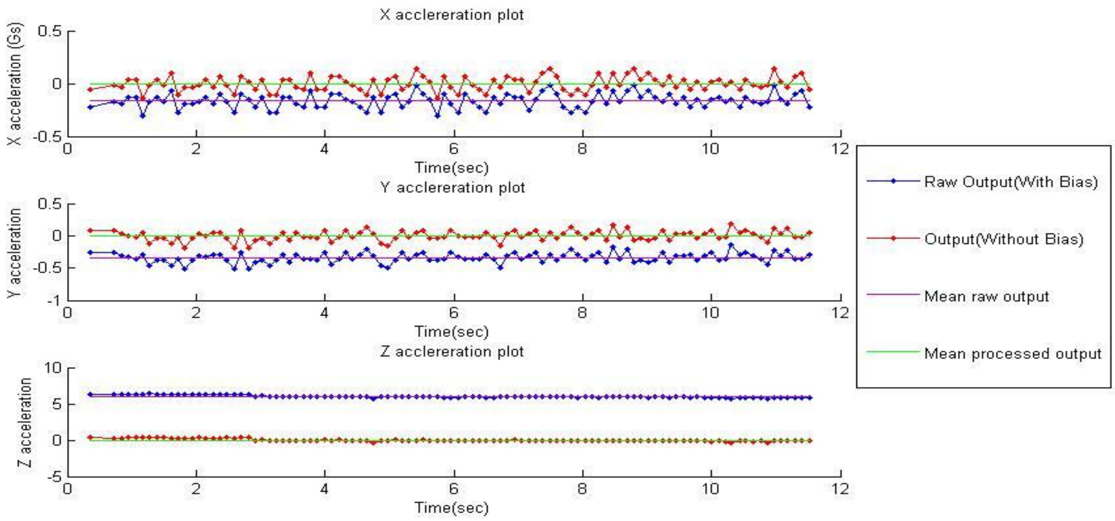
"X=-0.501 Y=-0.007 Z= 0.082"

The string was parsed to get the acceleration outputs. Here X refers to the acceleration in the X direction in gs'(g=9.81 m/s). The procedure is similar for Y and Z accelerations.The accelerometer biases(in gs') were found to be:-

X=-0.1642

Y=-0.3385

Z=+5.9899



The mean of the accelerometer data was calculated when the accelerometer was stationary. The mean was the bias in the respective direction.

Calculating Tilt

Tilt (in degrees) was calculated from the relation,

$$tilt = (180/\pi) * \cos^{-1}(accl_{xyz} / accl_z)$$

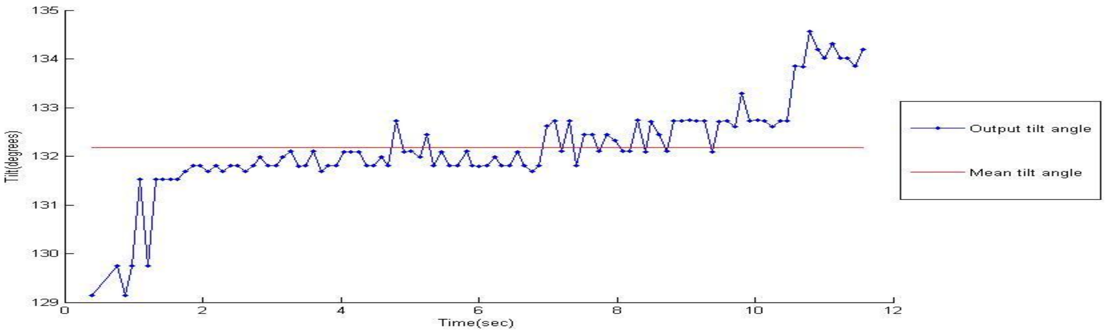
Where,

$$accl_{xy} = \sqrt{accl_x^{\,2} + accl_y^{\,2}}$$

$$accl_{xyz} = \sqrt{accl_{xy}^{\,2} + accl_z^{\,2}}$$

acclx is acceleration in the x direction.
$accl_y$ is the acceleration in the y direction.
$accl_z$ is the acceleration in the z direction.
$accl_{xy}$ is the resultant of the x and y accelerations.
$accl_{xyz}$ is the resultant of the x,y and z accelerations.

A sample plot is given below,



Crossbow DMU RGA300CA

DMU or Dynamic Measurement Unit was the IMU used. It can measure the accelerations about the xyz axis and the roll, pitch and yaw. It comprises 3 accelerometers about the x, y and z direction and a gyro about the z axis. All these sensors inside the DMU are MEMS sensors.

Approximately 130 degrees about x axis.

CrossBow DMU RGA300CA

Interface

The serial interface is standard RS-232, 38400 baud, 8 data bits, 1 start bit, 1 stop bit, no parity, and no flow control.

IMU Commands

Command ASCII	Response	Description
R	H	Ping: Pings DMU to verify communications.
P	None	Change to polled mode. Data packets sent when a G is received by the DMU.
C	None	Change to continuous data transmit mode
G	Data Packet	Get Data: Requests a packet of data from the DMU.
z <0-255>	Z	Calibrate and set zero bias for rate sensors by averaging over time.
b	Change baud rate	Autobaud detection.
S	ASCII String	Query DMU serial number.
v	ASCII String	Query DMU version ID string.

IMU Data-Packet

Data Packet Format

In general, the digital data representing each measurement is sent as a 16-bit number (two bytes). The data is sent MSB first then LSB. The data generally represents a quantity that can be positive or negative. These numbers are sent as a 16-bit signed integer in 2's complement format. The data is sent as two bytes, MSB first then LSB. The timer information and temperature sensor voltage are sent as unsigned integers.

Each data packet will begin with a two-byte header (0xAA 55) and end with a two-byte checksum. The checksum is calculated in the following manner:

1. Sum all packet contents except header and checksum.
2. Divide the sum by 0xFFFF.
3. The remainder should equal the checksum.

The data packet contains two extra pieces of information: the part number, and the BIT word output.

Bit	Data
0	Header(0xAA)
1	Header (0x55)
2	Roll Angle(MSB)
3	Roll Angle(LSB)
4	Pitch Angle(MSB)
5	Pitch Angle(LSB)
6	Yaw Rate(MSB)
7	Yaw Rate(LSB)
8	Acceleration X(MSB)
9	Acceleration X(LSB)
10	Acceleration Y(MSB)
11	Acceleration Y(LSB)
12	Acceleration Z(MSB)
13	Acceleration Z(MSB)
14	Time(MSB)
15	Time(LSB)
16	Temp Voltage(MSB)
17	Temp Voltage(LSB)
18	Part Number(MSB)
19	Part Number(LSB)
20	Bit(MSB)
21	Bit(LSB)
22	Checksum(MSB)
23	Checksum(LSB)

IMU Data Processing

The angled mode is used as the measurement mode. A polled data acquisition mode is used to acquire data from the device. Acquired data is cross-checked with the calculated checksum of the data before it is validated.

Acceleration $$accel = data*(GR*1.5)/2^{15}$$

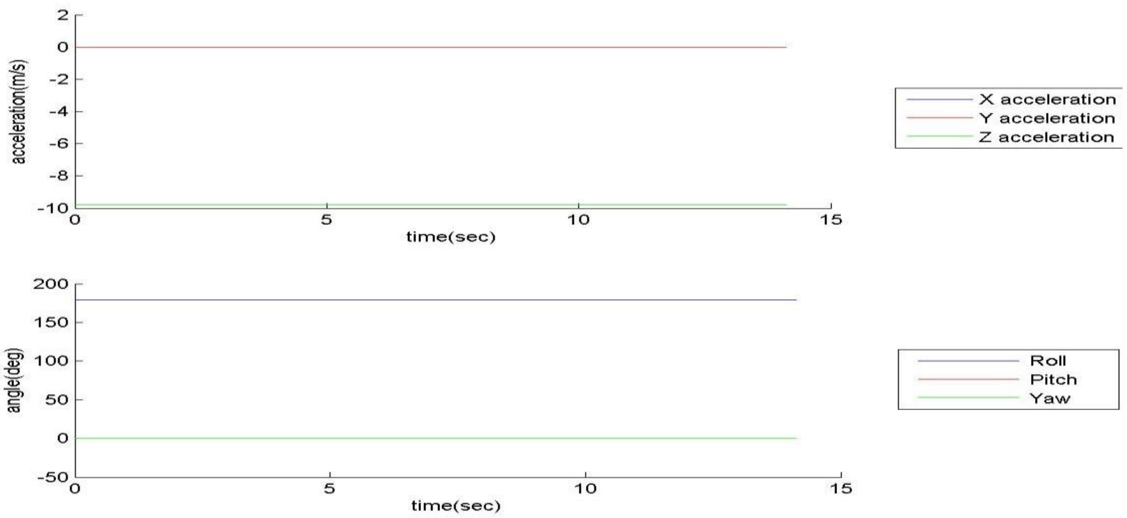
Angular rate $$rate = data*(AR*1.5)/2^{15}$$

Angle $$angle = data*(SCALE)/2^{15}$$

Where,
GR= 2 (G Range)
AR=100 (Angular Rate Range)
SCALE=180 (Angular Range)

Finding Orientation Using IMU

This is a sample plot with the z axis pointing up and the imu is a stationary. So we get roll of 180, pitch and yaw rate of zero. The x and y accelerations are zero and the z acceleration -9.8(approx) which represents the value of acceleration due to gravity.



Z axis pointing up (Upside down)

Kalman Filter

Salient features:-

- Noise Smoothing

  It can be used to reduce the noise and improve the measurements of various sensors.

- State estimation

  It can be used to reduce the error between the actual user input and the corresponding system output (i.e. state feedback applications)

- Recursive

  It computes the state estimates using only the latest or the most recent measurement and hence this filter has to be run recursively between every measurement cycle.

Here kalman filter was applied separately to the GPS and INS.

Kalman Filter Formulation

State Model, $\mathbf{x}(k) = \mathbf{F}(k)\mathbf{x}(k-1) + \mathbf{B}(k)\mathbf{u}(k) + \mathbf{G}(k)\mathbf{v}(k)$

Observation Model, $\mathbf{z}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{w}(k)$

Where,
x is state matrix
z is observation
u is an input (or control) vector
v is some additive noise
w is observation noise
H is observation matrix
B and G are input and noise transition matrices
F is the state transition matrix
k,k-1 are time steps

Kalman Filter Steps

The basic steps in a kalman filter are shown below,

- Project the state ahead $x(k/k-1) = F * x(k-1)$
- Project the covariance ahead $P(k) = F * P(k-1) * F^T + Q$
- Compute the kalman gain $W(k) = P(k) * H^T * (H * P(k) * H^T + R)$
- Update the state estimate $x(k/k) = x(k/k-1) + W(k) * \left(z(k) - H * x(k/k-1)\right)$
- Update the covariance $P(k) = \left(I - W(k) * H\right) * P(k-1)$

Where,
P is covariance matrix
Q and R are co variances of errors in state and observation models.

GPS Kalman Filter

Calculation of Velocity:-

$$vel_x = vel_x' + accl_x * \Delta t$$
$$vel_y = vel_y' + accl_y * \Delta t$$
$$vel_z = vel_z' + accl_z * \Delta t$$

$$P(k) = \left(I - W(k) * H\right) * P(k-1)$$

Where,
velx,vely,velz are the velocities in the x,y and z directions respectively.
velx',vely',velz' are the previous measures of velocity.
acclx,accly,acclz are the accelerations in the x,y and z directions respectively.

Parameters

The value of the different matrices and parameters are:-

$$Q = 0.2$$
$$R = 0.5$$

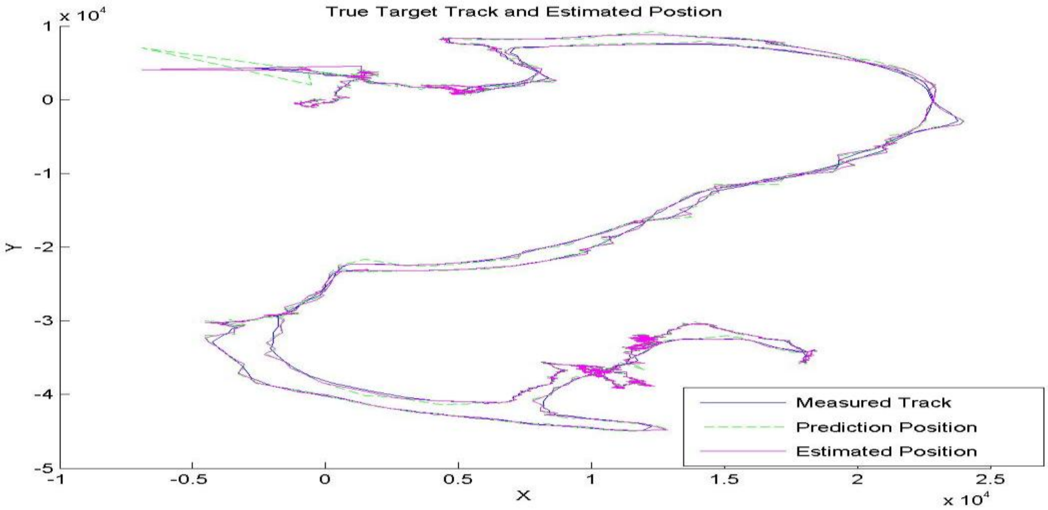
$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad G = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \quad F = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad X = \begin{bmatrix} dist_x \\ vel_x \\ dist_y \\ vel_y \end{bmatrix} \quad Z = \begin{bmatrix} dist_x \\ dist_y \end{bmatrix}$$

Where,
$dist_x$ is the distance travelled in the x direction
$dist_y$ is the distance travelled in the y direction
vel_x is the velocity in the x direction
vel_y is the velocity in the y direction
Δt is the time difference between the previous measurement value and current measurement value.

Sample Kalman Filtered GPS Data

The GPS data of path between Engineering Auditorium to Arts Canteen was kalman filtered to get the plot shown below

.



Kalman Filtered Sample GPS data of path of Engineering Auditorium (NUS) to Arts Canteen

INS Kalman Filter

Compensations and Corrections to INS data

Calculation of Acceleration

$$accl_x = maccl_x + 9.8 * \sin(pitch)$$
$$accl_y = maccl_y + 9.8 * \sin(roll)$$

Calculation of velocity

$$vel_x = vel_x' + accl_x * \Delta t$$
$$vel_y = vel_y' + accl_y * \Delta t$$
$$vel_z = vel_z' + accl_z * \Delta t$$

Calculation of position

$$dist_x = dist_x' + vel_x * \Delta t$$
$$dist_y = dist_y' + vel_y * \Delta t$$
$$dist_z = dist_z' + vel_z * \Delta t$$

Where,
vel_x, vel_y & vel_z are the velocities in the x, y and z directions respectively.
vel_x', vel_y' & vel_z' are the previous measures of velocity.
$accl_x$, $accl_y$ & $accl_z$ are the accelerations in the x, y and z directions respectively.
$dist_x$, $dist_y$ & $dist_z$ are the distances travelled in the x, y and z directions respectively.
$dist_x'$, $dist_y'$ & $dist_z'$ are the distances travelled till the previous reading in the x, y and z direction respectively.
Δt is the time difference between the time of the current and the previous reading.

Parameters

The values of different matrices and parameters are:-

Q=0.7

R=0.2

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \qquad G = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

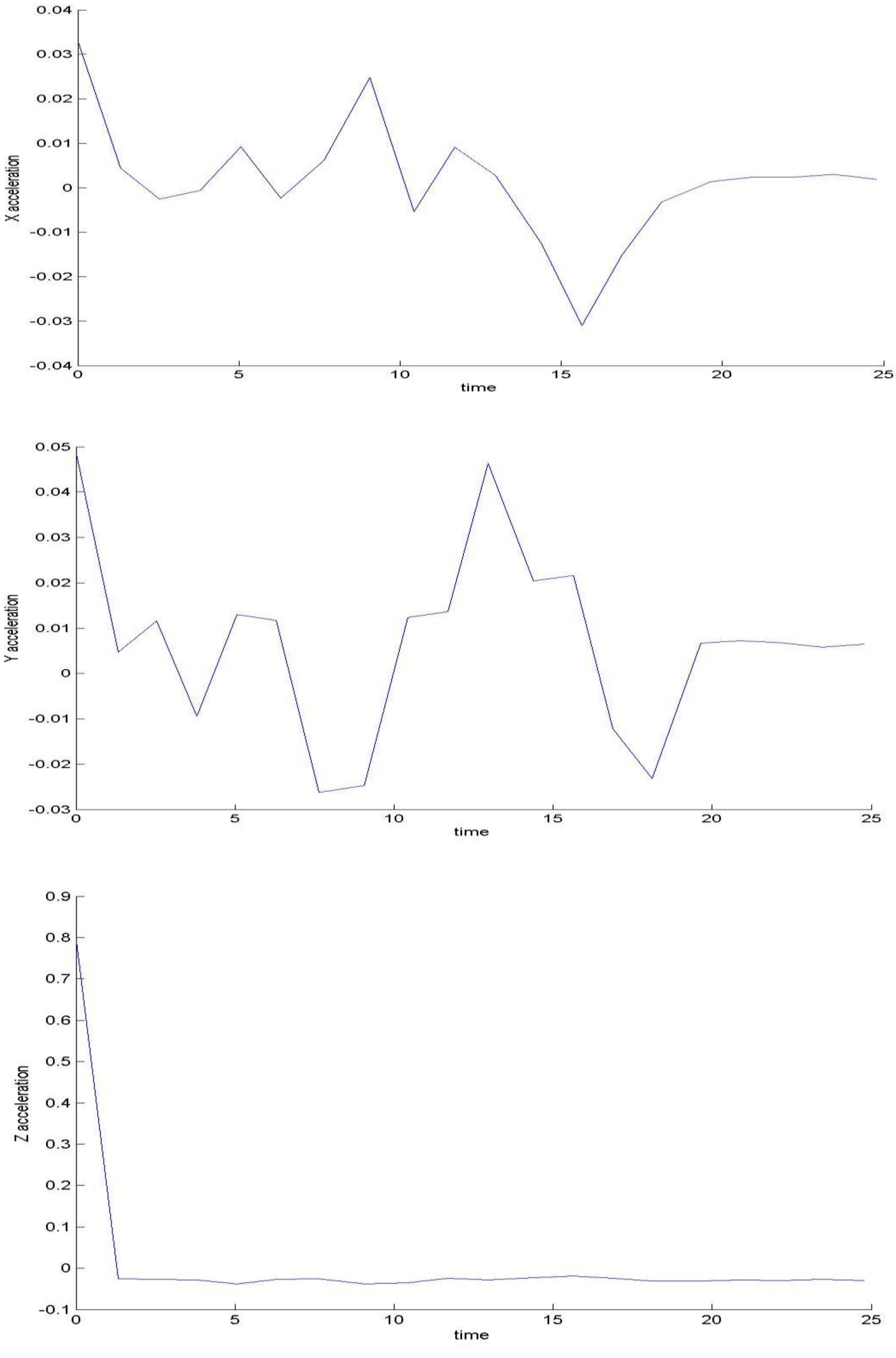
$$F = \begin{bmatrix} 1 & \Delta t & 0.5\Delta t^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & 0.5\Delta t^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t & 0.5\Delta t^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad X = \begin{bmatrix} dist_x \\ vel_x \\ accl_x \\ dist_y \\ vel_y \\ accl_y \\ dist_z \\ vel_z \\ accl_z \end{bmatrix} \qquad Z = \begin{bmatrix} dist_x \\ dist_y \\ dist_z \end{bmatrix}$$

Where,
vel_x, vel_y & vel_z are the velocities in the x, y and z directions respectively.
vel_x', vel_y' & vel_z' are the previous measures of velocity.
$accl_x$, $accl_y$ & $accl_z$ are the accelerations in the x, y and z directions respectively.
$dist_x$, $dist_y$ & $dist_z$ are the distances travelled in the x, y and z directions respectively.
Δt is the time difference between the time of the current and the previous reading.

Sample Kalman Filtered INS Data

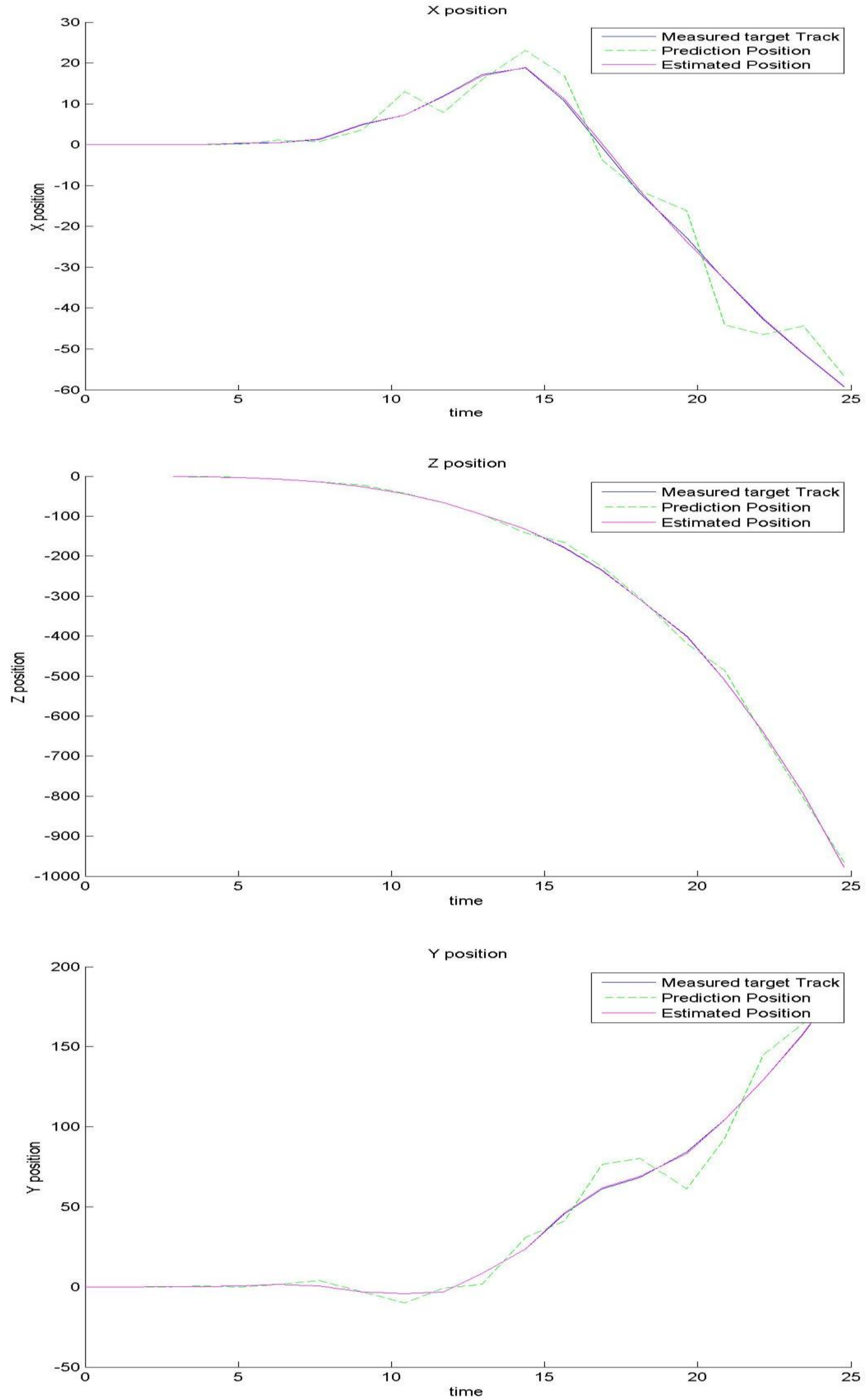
The INS data plot when the INS was moved over a table top (height=2.5 m approx) over the circumference of a rectangle (1.5 m X 2.0 m) taking right-bottom vertex as the starting point till it completes one complete loop over the circumference in the anti-clockwise direction till it reaches the starting point.

Acceleration Plots



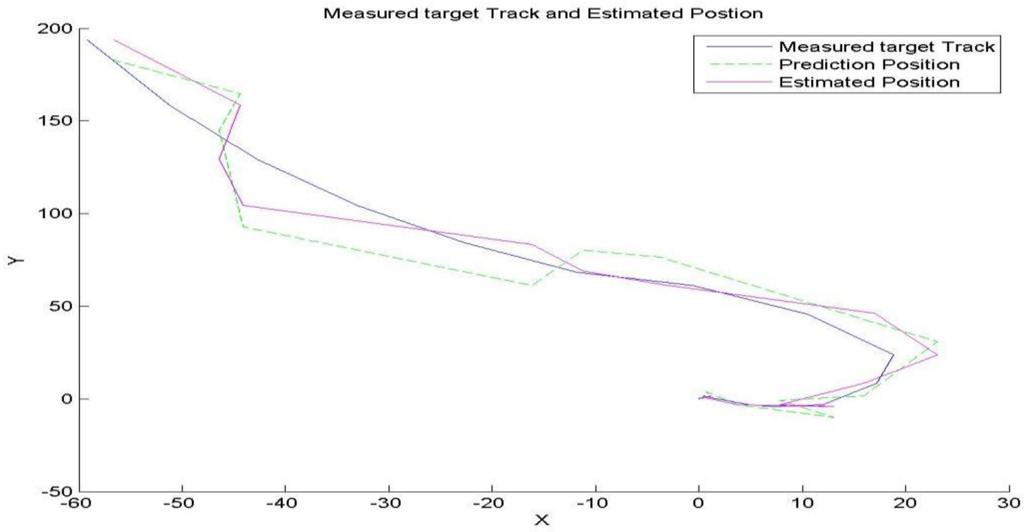
Plots of X, Y & Z against time

Displacement Plots



X Y & Z displacements against time

Path Comparison between Measured, Predicted and Estimated Tracks



Y displacement plotted against X displacement to get the path traversed. Measured, Predicted and Estimated paths are shown.

The plots have to be improved upon as they do not portray the expected rectangular or close to rectangular trajectory. With further corrections and calculations better results can be obtained.

Conclusion

GPS, INS and accelerometer capabilities and accuracies were tested. Several types of paths, orientation and environments were chosen to check the devices' performance.

The San Jose Navigation GPS was used to track the location of a dynamic body. The results obtained were highly satisfactory. Kalman filter was applied on the output data to improve its accuracy. The trajectories generated when the GPS output data was plotted closely corresponded with the expected trajectories which were derived from the observer's perception. When plotted over a map, it showed close correspondence to the actual path in map.

The SerAccel V5 accelerometer was used to find the accelerations in three mutually perpendicular x, y and z axes when it was fixed to a dynamic body. It was found to give noisy measurements with an accuracy of 0.2g. It was satisfactorily used for tilt sensing applications. It was found to give inaccurate results when used to measure displacement of a dynamic body.

The CrossBow DMU was used to to get the accelerations in the x, y and axes and also the roll, pitch and the yaw rate of a dynamic body when the DMU was fixed to the body. The orientation outputs of the body calculated from the IMU data was found to be satisfactory. The displacements of the dynamic body when calculated from the IMU date were found to have a very high drift.

Further Study

The results of the accelerometer could be improved upon by implementing kalman filter or some other filtering technique to enhance its accuracy and remove the errors. The DMU could be further worked upon by implementing better filters than the currently used kalman filters to estimate a more accurate result.

Some Interesting Projects:-

- Implementation of a Extended Kalman Filter on the IMU
- Integration of GPS, INS, accelerometers, camera and other sensors like LiDAR can be integrated to produce a localisation system.

Codes

1. Accelerometer as tilt sensor, matlab code to find the tilt angle of a body using accelerometer – Accelerometer_tilt.
2. GPS Kalman Filter, matlab code to kalman filter GPS data –GPS_KalmanFilter.
3. Superimposing GPS plot over a map, matlab code to superimpose a gps path plot over a map –GPS_Map_Superimpose.
4. INS orientation, matlab code to find the orientation using INS data –INS_Orientation.
5. INS Kalman Filter, matlab code to kalman filter INS data –INS_KalmanFilter.
6. GPS INS Accelerometer Kalman Filter Integration, Visual C++ Project which integrates GPS, INS and Accelerometer and applies sensor fusion using Kalman Filter – GPS_INS_KalmanFilter.
    a. GPS Header File(gps.h)
    b. GPS Code(gps.cpp)
    c. INS Header File(ins.h)
    d. INS Code(ins.cpp)
    e. Accelerometer Header File(accl.h)
    f. Accelerometer Code(accl.cpp)
    g. Kalman Filter Header File(kalman.h)
    h. Kalman Filter Code(kalman.cpp)
    i. Position Header File(position.h)
    j. Integrating Code[Main function](gps_ins.cpp)

References

1. Kadir Akyol, Xiaoping Yun and Eric Bachmann, *Hardware Integration of a Small Autonomous Underwater Vehicle Navigation(SANS) using a PC/104 Computer*, Naval Post Graduate School, Monterey, California, 1999.
2. *Crossbow 6-axis Dynamic Measurement System DMU RGA30CA user manual*, Crossbow INc.
3. *MTK-3301 GPS Receiver Series user manual*, Sanav Inc.
4. *SerAccel V5 triple axis accelerometer user manual*, SparkFun Electronics.
5. Robertson Bayer, *Windows Serial Port programming,* 2008.
6. George Kantor, *Kalman's Beautiful Filter,* power point presentation, Carnegi Mellon University, 2000.
7. Lee-Ling Sharon Ong, *An Introduction to Bayesian Estimation,* Power Point Presentation, National university of Singapore.
8. Martin Bauer, Katharina Pentenrieder, *Sensor Fusion Using Kalman Filter,* internet article, http://campar.in.tum.de/Chair/KalmanFilter , 2005.
9. Distance Calculation, Internet Article, http://www.meridianworlddata.com/Distance-Calculation.asp.