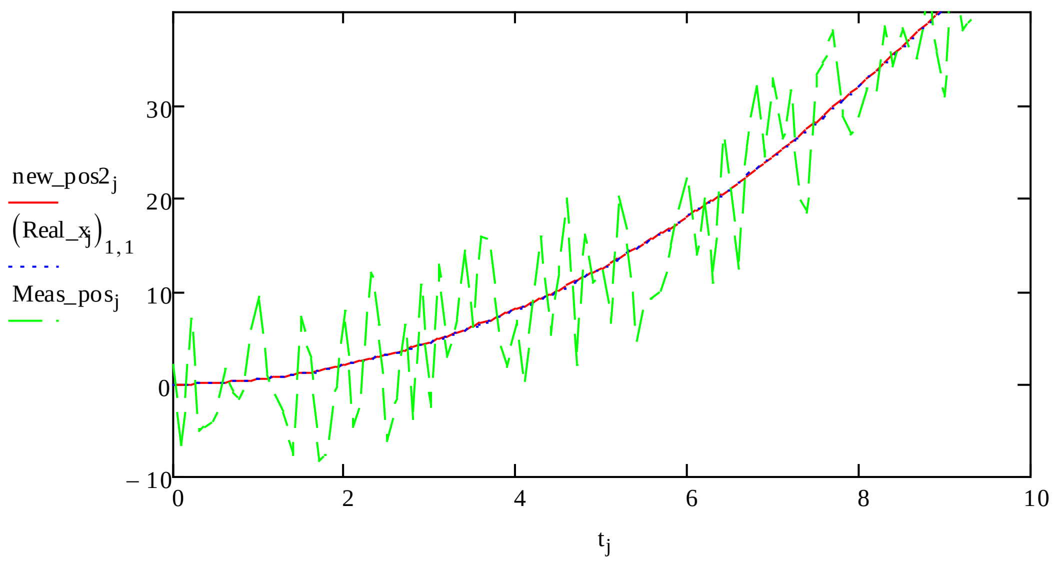
Sensor data fusion with the Kalman Filter

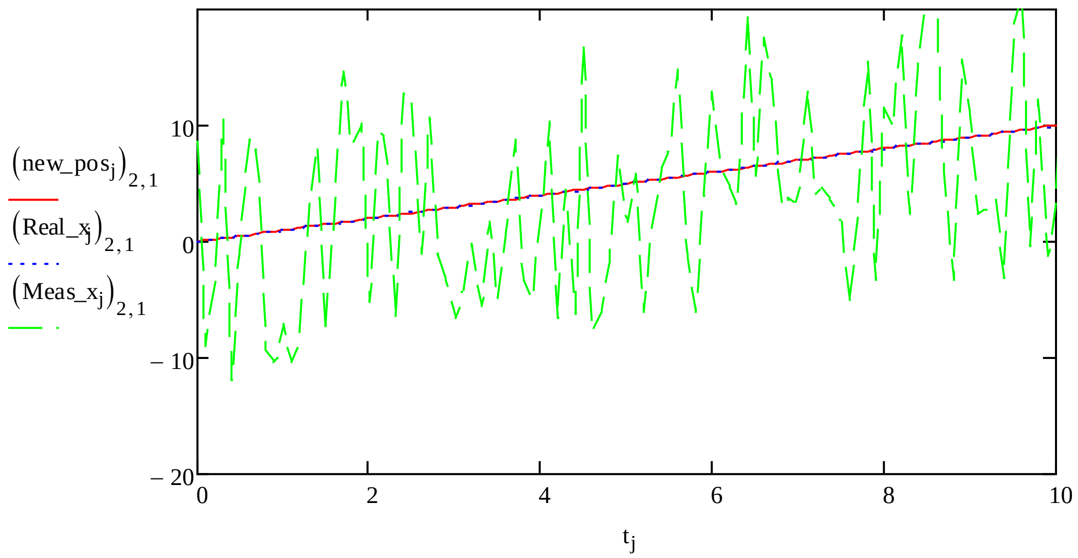
In the last session we saw how the Kalman filter is an extremely good smoothing filter. However, a similar result could also have been achieved with more simple techniques, such as a low pass filter. But the KF can do more than that.

In this session we will see how the Kalman Filter can be used to combine data from two different types of sensor in order to obtain more accurate data. As an example we will use an Inertial Navigation problem.

In the previous example we already saw that the method used (which resulted in smoothed position data from a GPS system) also gave us more data, namely the velocity of the vehicle, which wasn't measured and used in the KF (although we simulated the measurement just for comparisons sake)



Smoothed position data

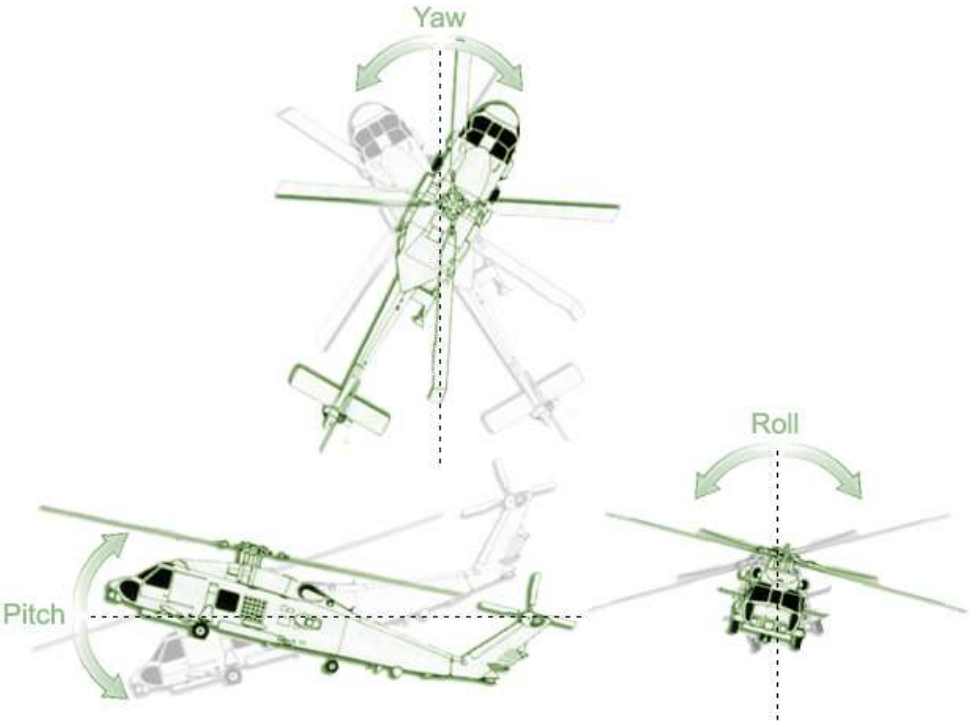


Extra data from the method: smoothed velocity data

So already we see that if we use the system model carefully, we can extract more than just smoothed data.

In the next problem, we will use accelerometer and gyro data to calculate horizontal attitude. Both of these sensors return a measurement with respect to the inertial (moving) frame. Any method that measures position and attitude is called an inertial navigation system.

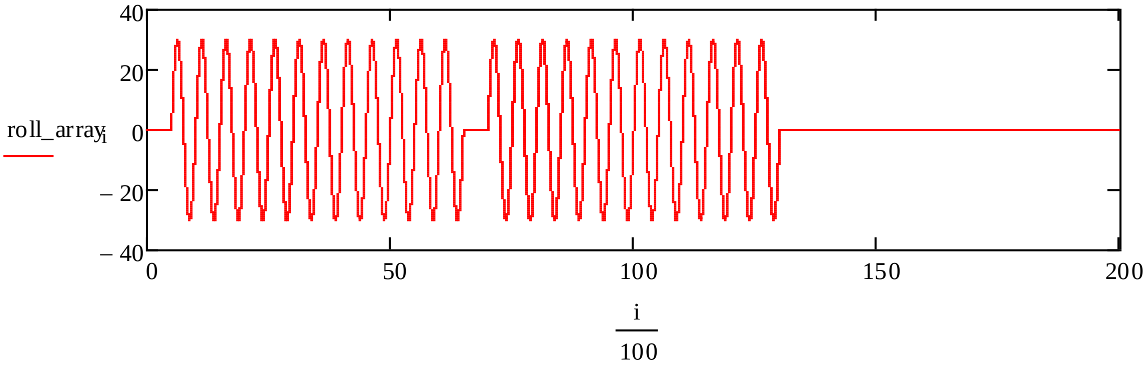
We will use a helicopter as an example, but this is easily modified to other systems.

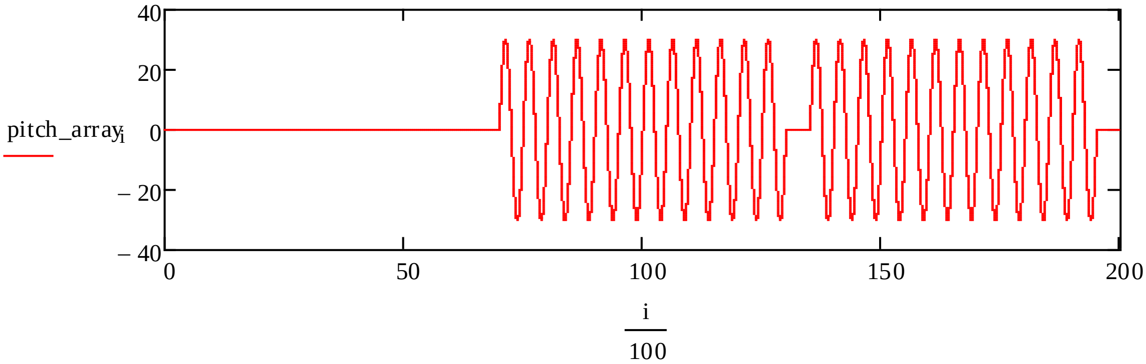


[from : http://www.acme-worldwide.com/dynamic_motion_seat_Rotary.htm}

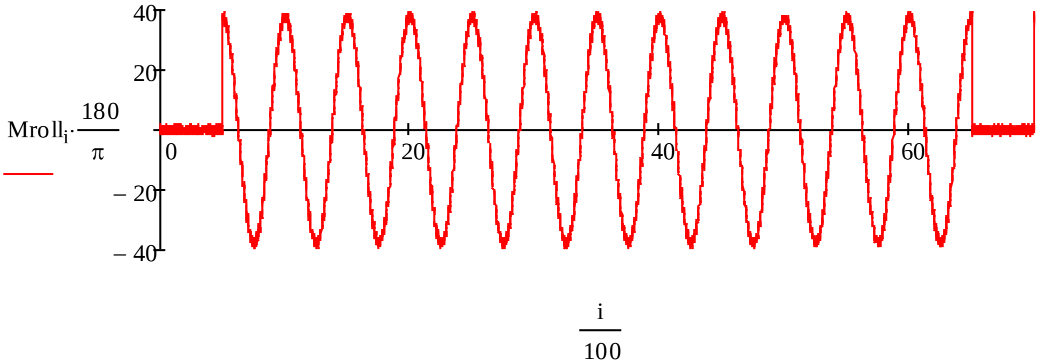
Here we are only interested in the attitude of the helicopter, so only roll and pitch. Yaw does not play a role in the attitude.

We will test the method by measuring gyroscope and acceleration data during a test in which a rolling oscillation, followed by a combined roll and pitch oscillation, then by a pitch oscillation, all with frequency 0.2Hz and amplitude 30° for 60 seconds each movement. See the graph below.





This is the movement applied to the helicopter. Typical measurements from the gyros, including a typical error, are shown below. (We will come back to the error later)

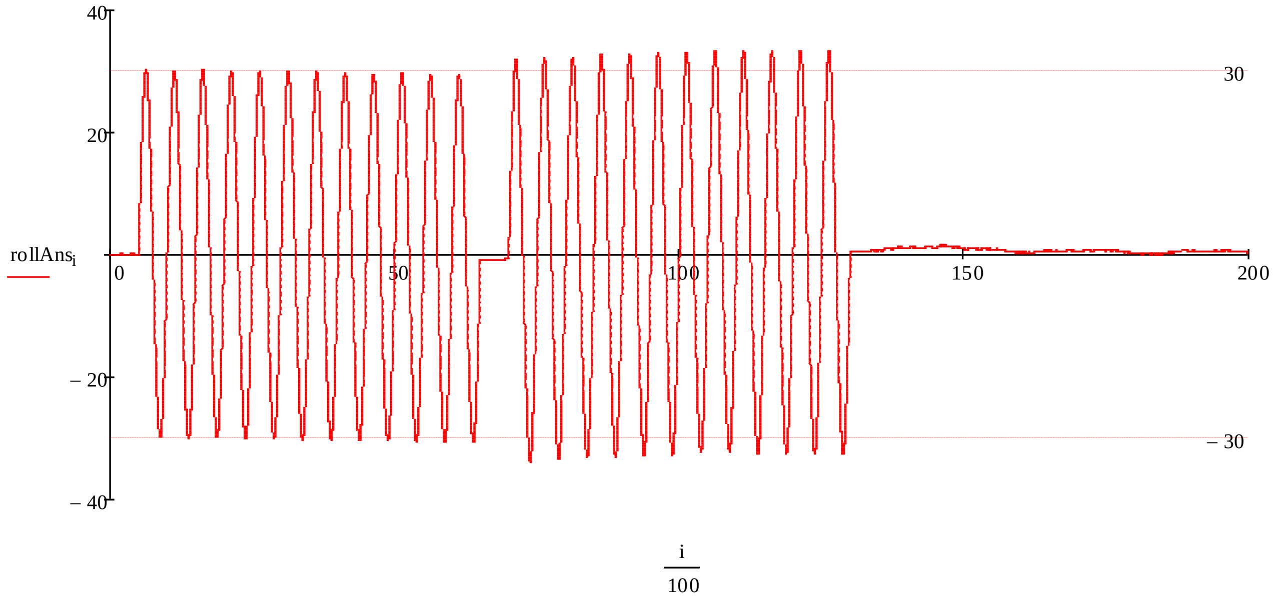


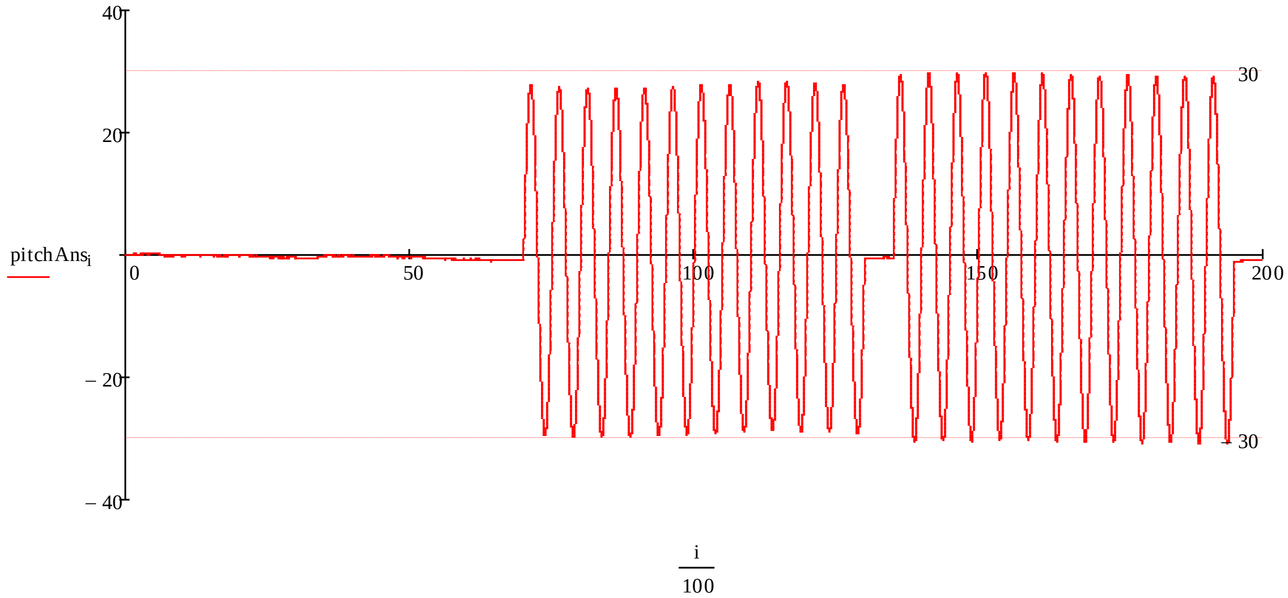
Gyro measurements

We would like to know the Euler angles of the helicopter, that is - with respect to a fixed axis (horizontal, vertical). But the gyros give us information about the rate of change of the angle with respect to the helicopter(moving) axis. The relation between the measured rate of change and the Euler rate of change is given by

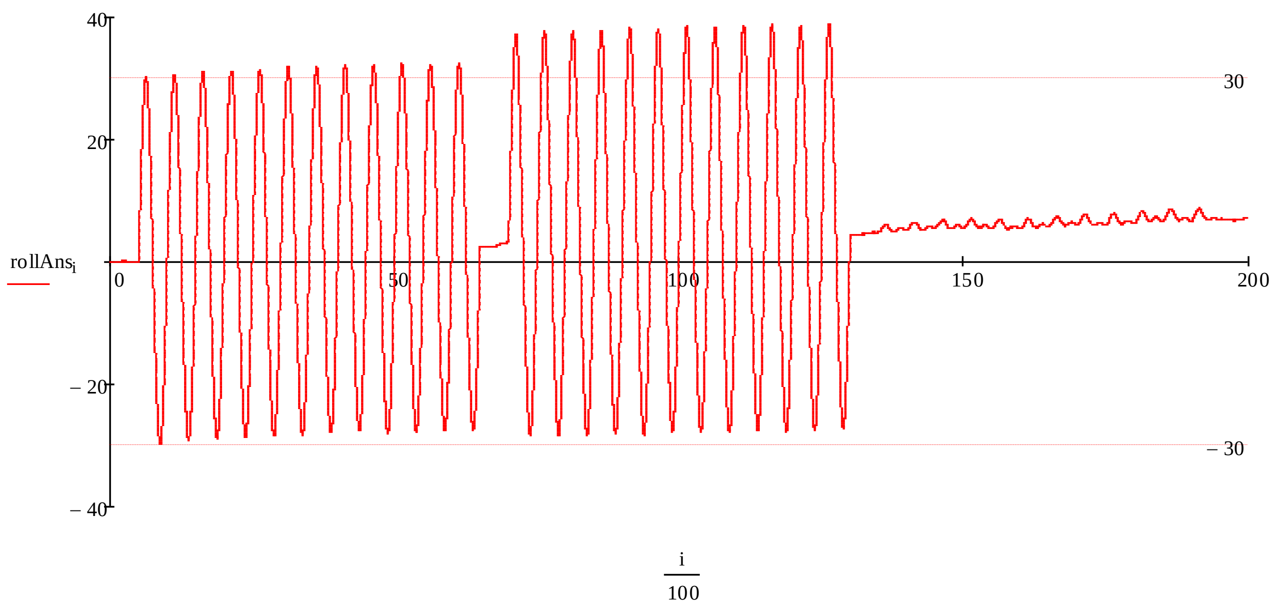
$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 1 & sin\varphi . cos\theta & cos\varphi . tan\theta \\ 0 & cos\varphi & -sin\varphi \\ 0 & sin\varphi / cos\theta & cos\varphi / cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

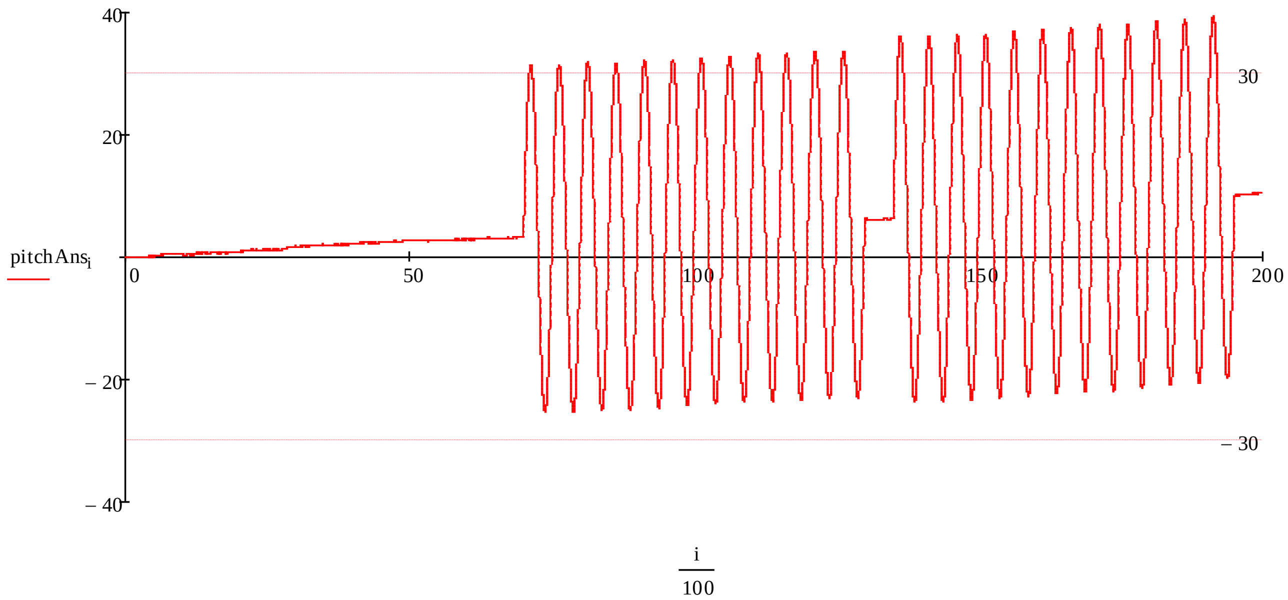
Where p,q,r are the angular velocities from the gyros, and φ, θ, ω are the roll, pitch, yaw Euler angles respectively. We would then have to integrate to get the required angles. If we apply this to the measurement data we will get something similar to the diagrams below. (From MathCad file "KF Gyro 1.xmcd")





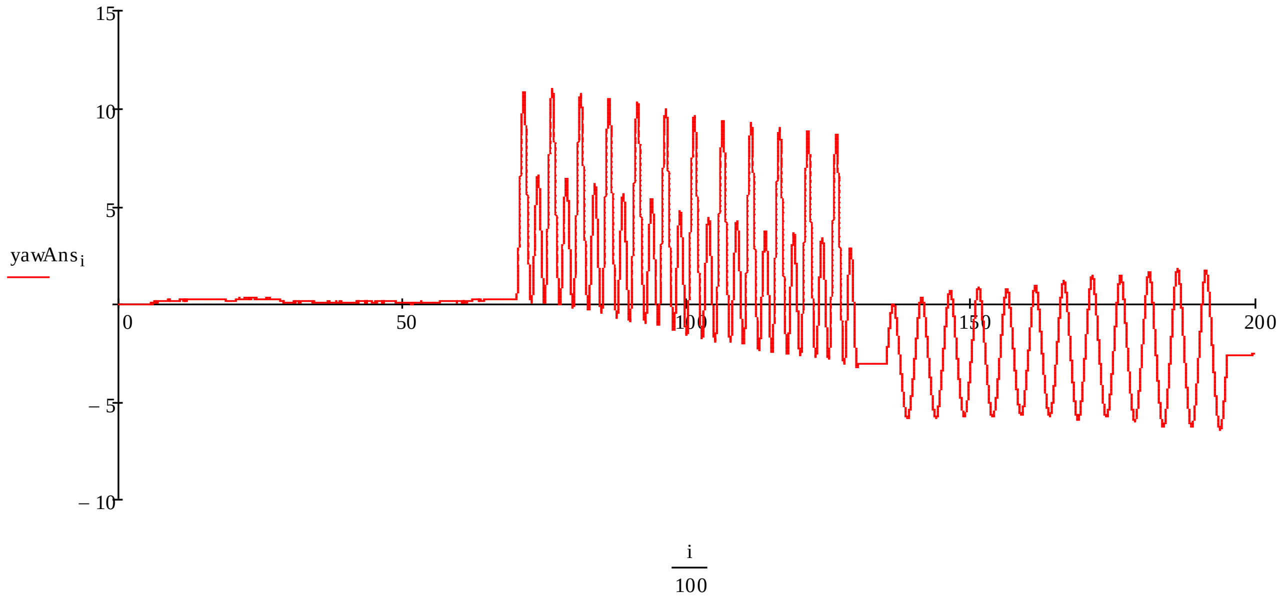
If we examine these results, then we can actually be quite satisfied. The calculated roll and pitch angles seem to be fairly accurate, are contained roughly within the ±30°. However these simulated data measurements are for an ideal sensor with low error (0.5% full scale, full scale being 360°/sec) and zero offset. Because of this zero offset, the 'random' positive and negative errors seem to cancel each other out during the integration step. If, however, we conduct a more realistic simulation with a small offset of 0.05°/sec, then we see the typical drift that is encountered in most gyro measurements.





This drift quickly causes problems in any navigation system . In this example , after only 3 minutes we are already nearly 10° off actual values, and this is with a relatively small offset.

This also influences the yaw. Even though we are not changing or measuring the yaw here (assuming that it is a constant zero), we still get results for the yaw angle:



Accelerometer measurements

If we have a 3-axis accelerometer attached to the system, then it's possible to gain the attitude by means of the following equation:

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} 0 & w & -v \\ -w & 0 & u \\ v & -u & 0 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} + g. \begin{bmatrix} \sin\theta \\ -\cos\theta \,.\, \sin\varphi \\ -\cos\theta \,.\, \cos\varphi \end{bmatrix}$$

Where f_x, f_y, f_z are the accelerations measured in the three directions relative to the sensor frame; u,v,w are the velocities about each axis in the body frame, p,q,r are the gyro readings, and g is acceleration due to gravity. Roll and pitch are present in the last term. Note : we have not used the yaw here.

In order to determine the roll and pitch angles, we need to know the velocities. For simplicity at this stage, let's assume that the helicopter is hovering. This means that

$$u = v = w = \dot{u} = \dot{v} = \dot{w} = 0$$

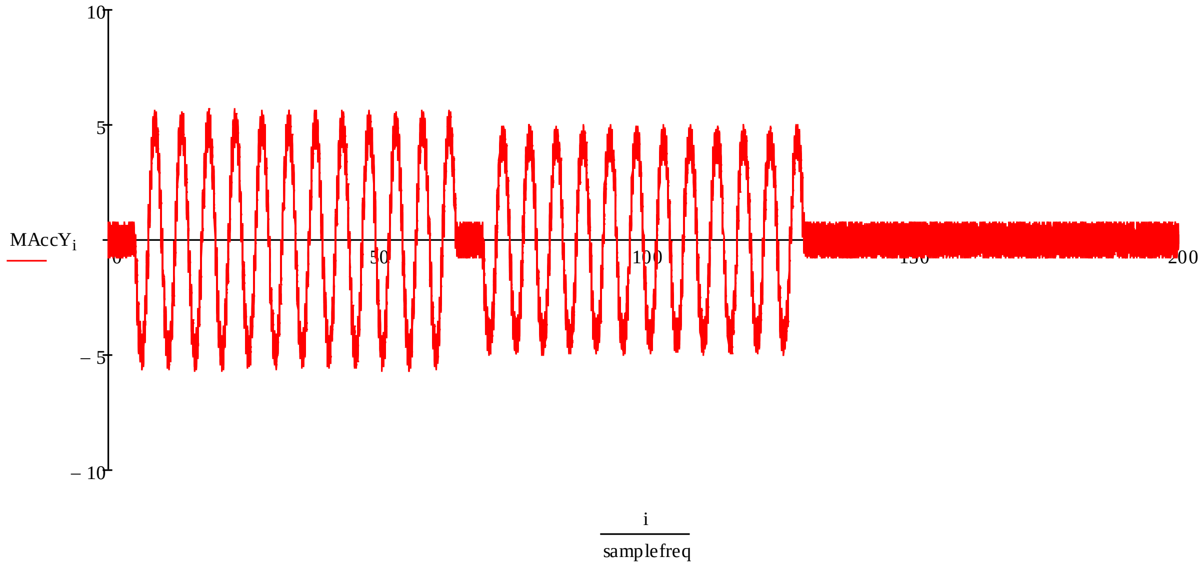
So the first two terms disappear, and the equation becomes

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = g. \begin{bmatrix} \sin\theta \\ -\cos\theta \,.\, \sin\varphi \\ -\cos\theta \,.\, \cos\varphi \end{bmatrix}$$

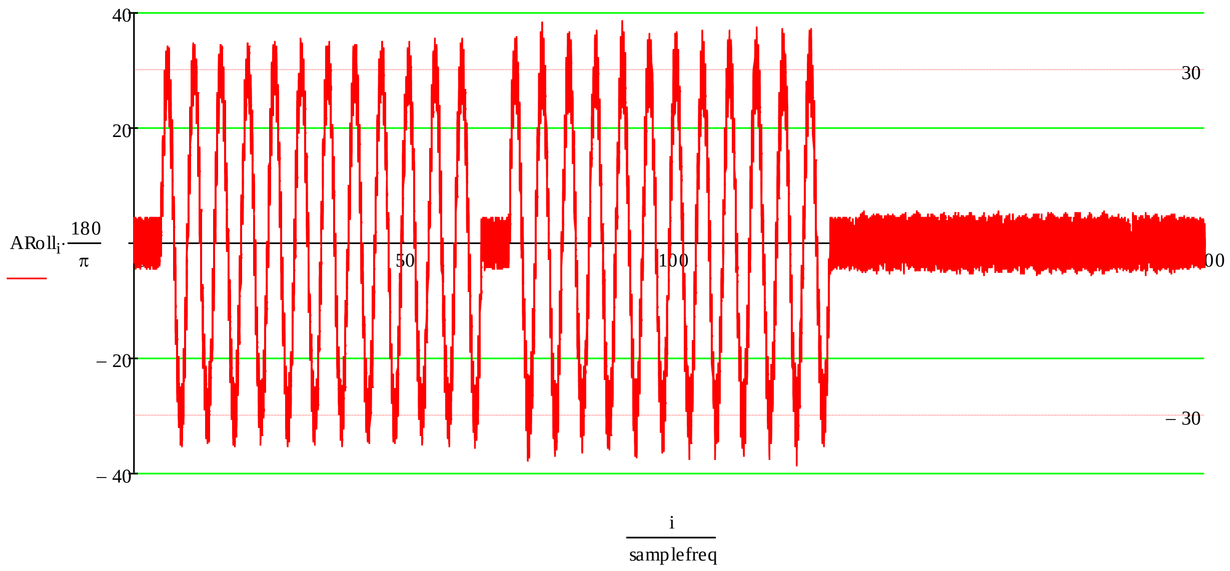
From this we obtain

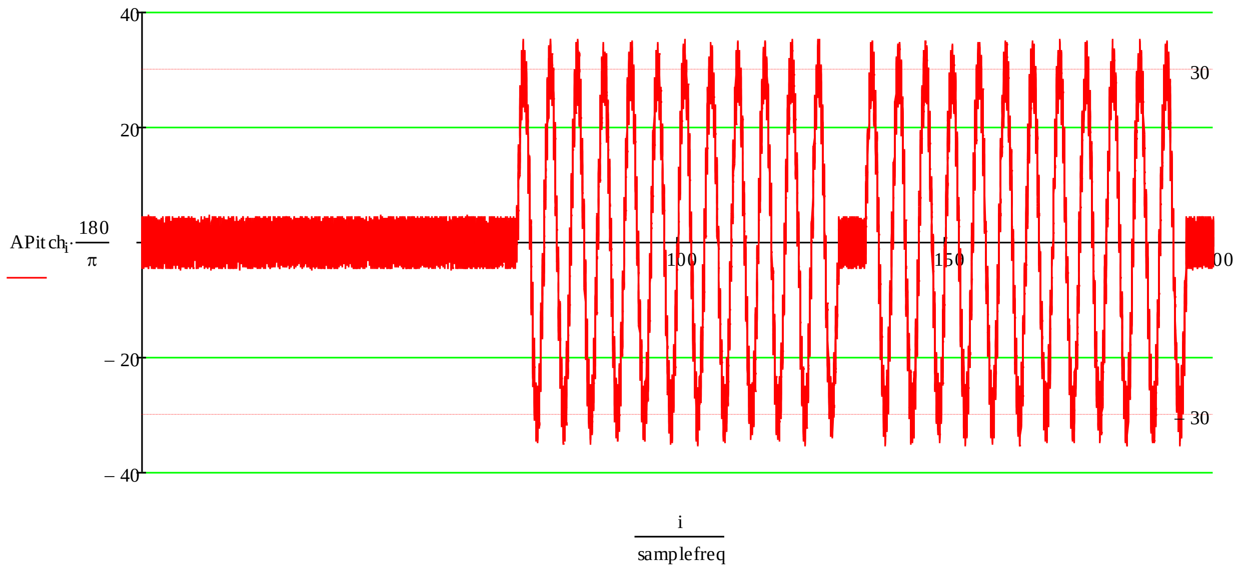
$$\varphi = sin^{-1}\left(\frac{-f_y}{g.\cos(\theta)}\right)$$
$$\theta = sin^{-1}\left(\frac{f_x}{g}\right)$$

We simulate measurements using a fairly inaccurate accelerometer (1% full scale, Full scale being 8 times gravity). No offset is included here. This gives the following graph for the y-direction accelerometer readings.



If the simulated x and y direction accelerometer readings are used in the calculations of the roll and pitch angles, we obtain the following figures:





We see that the oscillations seem to be represented well, and that there is no drift. The values of the roll and pitch angles however are not very accurate. This is simply because we chose to use a low accuracy sensor (for demonstration purposes).

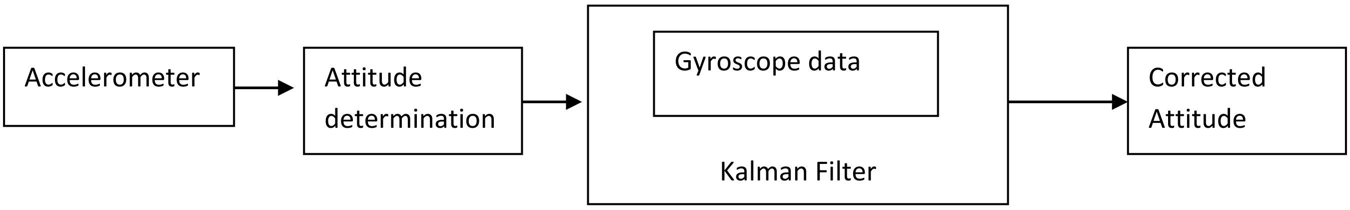
Applying the Kalman Filter to both gyro and accelerometer data.

Above we have seen that both the sensors we use for measuring the roll and pitch angles have shortcomings –

- The gyros seem to be fairly accurate in the angles measured, but will suffer from drift due to the small offset present in any sensor.
- The accelerometer we ‘chose’ was inaccurate (giving large errors in the roll and pitch angles), but did not suffer from drift.

In this paragraph we will demonstrate the power of the Kalman Filter, combining the data from the two sensors to give a more accurate measurement. This is called sensor data fusion.

The principle used here is shown in the following diagram:



But first we have to make the state equation.

The property we are interested in is the attitude, so set the state variable to

$$\bar{x} = \begin{pmatrix} \varphi \\ \theta \\ \omega \end{pmatrix}$$

(the roll, pitch and yaw angles in the Euler framework)

The relation between these Euler angles and the rate of change of roll, pitch and yaw from the helicopter framework (i.e. the gyro data) has already been given on page 3:

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 1 & sin\varphi.cos\theta & cos\varphi.tan\theta \\ 0 & cos\varphi & -sin\varphi \\ 0 & sin\varphi/cos\theta & cos\varphi/cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

However, this does not fit our ‘standard’ for the state equation, which should have the form we need in the Kalman Filter

$$x_{k+1} = Ax_k + w_k \Rightarrow \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\omega} \end{bmatrix} = [\blacksquare] \begin{bmatrix} \varphi \\ \theta \\ \omega \end{bmatrix} + w$$

Where $[\blacksquare]$ is a matrix. In order to remedy this, we need to introduce the quaternions. This is, simply said, an alternative method of expressing attitude. So we now use the following state variable

$$\bar{x} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix}$$

A small amount of research on the internet will reveal where these quaternions come from, and how they are related to gyro data:

$$\begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{pmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \cdot \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix}$$

If this is expressed in a discrete form (i.e. making a discretisation of the variables between times k and k+1), then we obtain the following form

$$\begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix}_{k+1} = \left[I + \frac{\Delta t}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \right] \cdot \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix}_{k}$$

This has the correct form for the state equation.

The large matrix is the matrix A from the state equation. Here, however, it is no longer a constant, but changes at every time interval because of the changing gyro data.

Now we need the equations for the measurements. These are derived from the acceleration data. More research reveals the following relation to convert from Euler angles to quaternion

$$\begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} = \begin{pmatrix} cos\frac{\varphi}{2}\ cos\frac{\theta}{2}\ cos\frac{\omega}{2} + sin\frac{\varphi}{2}\ sin\frac{\theta}{2}\ sin\frac{\omega}{2} \\ sin\frac{\varphi}{2}\ cos\frac{\theta}{2}\ cos\frac{\omega}{2} - cos\frac{\varphi}{2}\ sin\frac{\theta}{2}\ sin\frac{\omega}{2} \\ cos\frac{\varphi}{2}\ sin\frac{\theta}{2}\ cos\frac{\omega}{2} + sin\frac{\varphi}{2}\ cos\frac{\theta}{2}\ sin\frac{\omega}{2} \\ cos\frac{\varphi}{2}\ cos\frac{\theta}{2}\ sin\frac{\omega}{2} - sin\frac{\varphi}{2}\ sin\frac{\theta}{2}\ cos\frac{\omega}{2} \end{pmatrix}$$

This relation will be used in the Kalman Filter, as the measurement 'z_k'. The matrix H becomes the unit matrix since all the quaternions are measured via the relation above.

The Kalman Filter we will use looks slightly different from the one in the previous session, but they are equivalent.

1. **Set the initial values $\widehat{x_0}$, P_0**
2. **Predict the state and error covariance**
   - $\widehat{x_k}\sim\ = A.\widehat{x_{k-1}}$
   - $P_k\sim = AP_{k-1}A^T + Q$
3. **Compute the Kalman gain**
   - $K_k = P_k\sim H^T(HP_k\sim H^T + R)^{-1}$
4. **Compute the estimate**
   - **Measurement z_k ➔ $\widehat{x_k} = \widehat{x_k}\sim + K_k(z_k - H\widehat{x_k}\sim)$ ➔ Estimate $\widehat{x_k}$**
5. **Compute the error covariance**
   - $P_k = P_k\sim - K_kHP_k\sim$
6. **Repeat from step 2 for the remaining 'k'**

Step 4 gives us the corrected state variable estimation. Once we have this, then we need to convert back from quaternion to Euler angles. This is done via the following relationship

For Euler angles we get:

$$\begin{bmatrix}\phi\\ \theta\\ \psi\end{bmatrix} = \begin{bmatrix}\arctan\frac{2(q_0q_1+q_2q_3)}{1-2(q_1^2+q_2^2)}\\ \arcsin(2(q_0q_2-q_3q_1))\\ \arctan\frac{2(q_0q_3+q_1q_2)}{1-2(q_2^2+q_3^2)}\end{bmatrix}$$

arctan and arcsin have a result between −π/2 and π/2. We need to replace the arctan by atan2 to generate all the orientations in all 4 quadrants

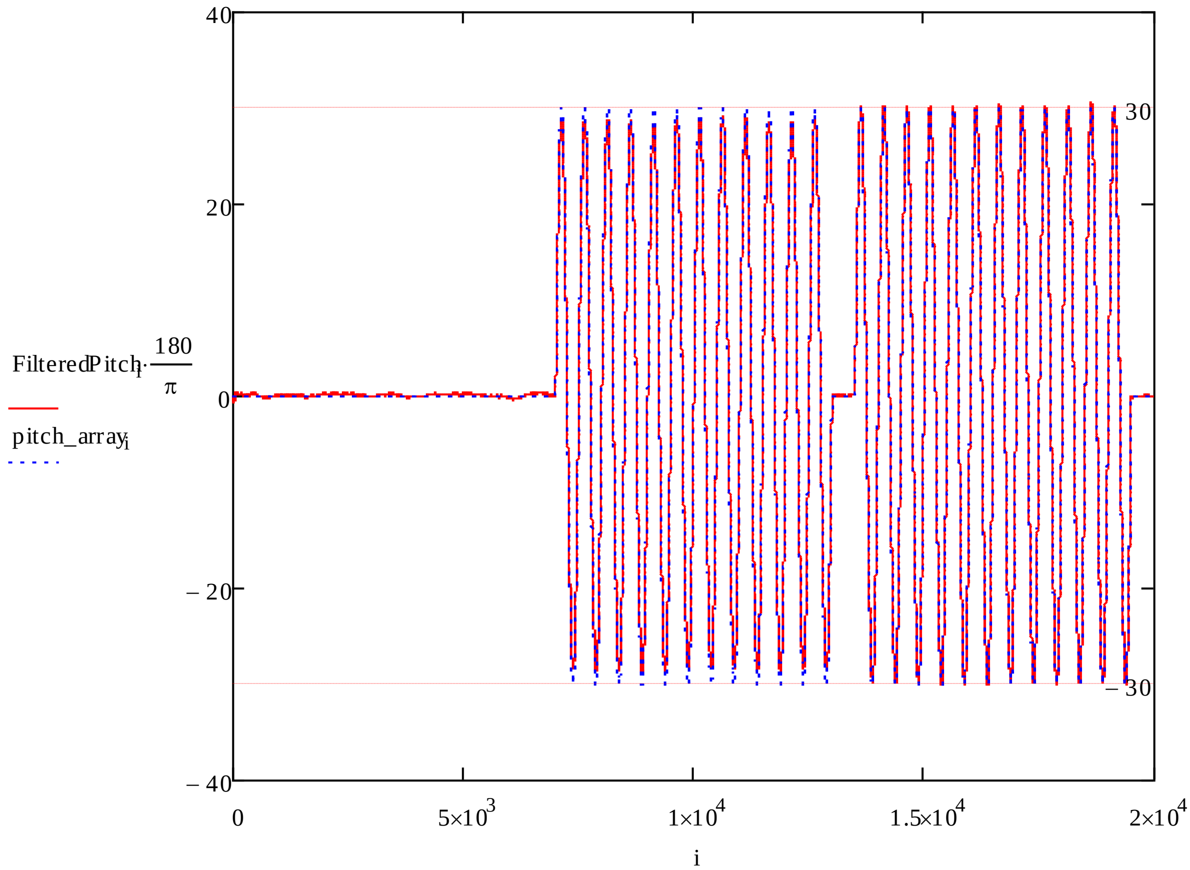
$$\begin{bmatrix}\phi\\ \theta\\ \psi\end{bmatrix} = \begin{bmatrix}\operatorname{atan2}(2(q_0q_1+q_2q_3),1-2(q_1^2+q_2^2))\\ \arcsin(2(q_0q_2-q_3q_1))\\ \operatorname{atan2}(2(q_0q_3+q_1q_2),1-2(q_2^2+q_3^2))\end{bmatrix}$$

The noise covariance matrices Q and R are related to the system signal and have to be derived by experiment. Initial values of $\widehat{x_0}$, P_0 are set as

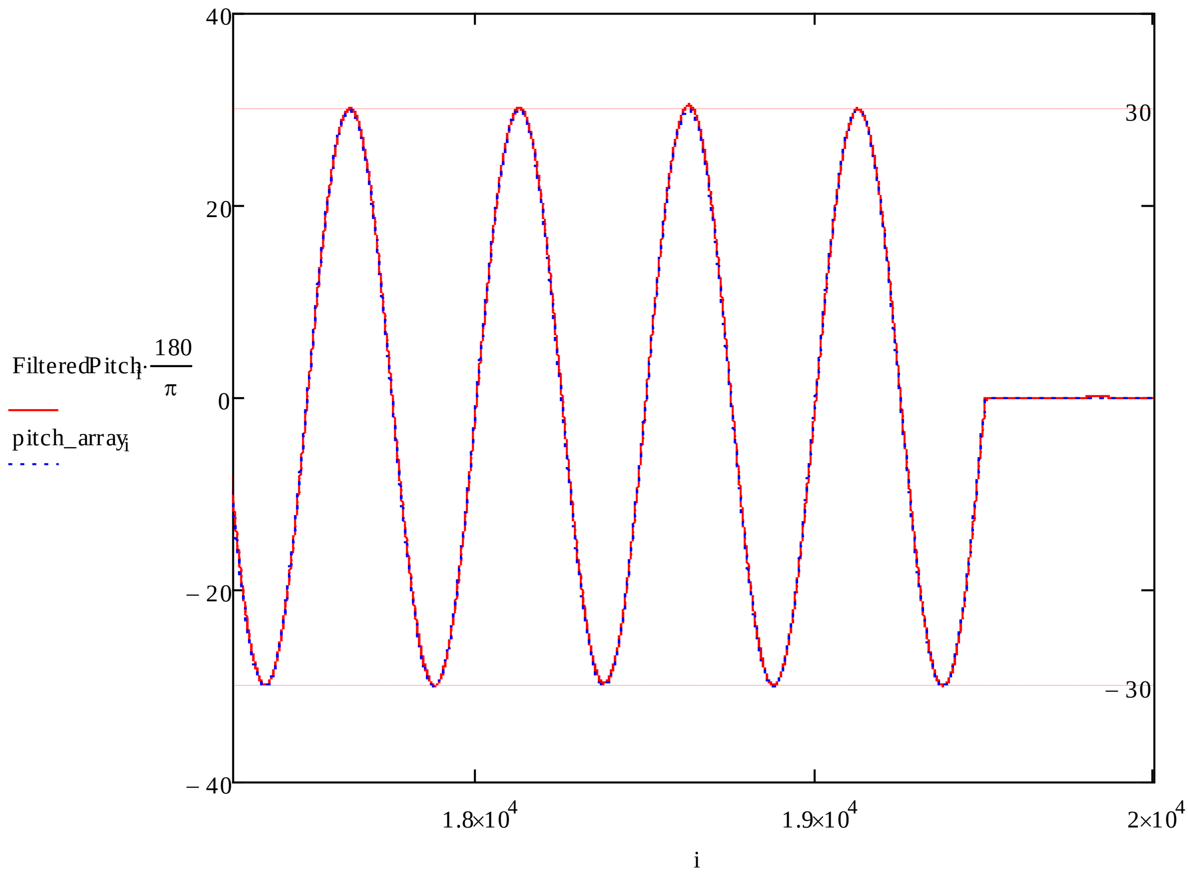
$\widehat{x_0}\sim = \begin{pmatrix}q_1\\ q_2\\ q_3\\ q_4\end{pmatrix} = \begin{pmatrix}1\\ 0\\ 0\\ 0\end{pmatrix}$, $P_0\sim = \begin{bmatrix}1&0&0&0\\ 0&1&0&0\\ 0&0&1&0\\ 0&0&0&1\end{bmatrix}$. Physically this translates to all the Euler angles initially being zero.

The MathCad document shows this whole process.

Results for the pitch



Zooming in on the least accurate part, we see very little difference between the actual pitch and the estimated pitch.



Assignment 2

a) Given the gyro and accelerometer sensor data readings to be found in the accompanying excel sheet, use the Kalman Filter algorithm above to combine the data and find a new estimate for the attitude.

b) Alternatively ; do the same as in (a), but use real data that you gain yourself from a 3 dof gyro and 3 dof accelerometer/