

Analysis of Drift in Knee Joint Angle

Sandeep Kumar

July 14, 2016

Problem

We are using two IMU sensors, one mounted on thigh and the shank to calculate the Knee joint angle (Flexion and Extension).

We are using the techniques mentioned in the paper [1] and [2] to do so.

One of the method mentioned in the papers uses only Gyroscope data to calculate the Joint Knee angle.

Result of using that algorithm on the data gather from the EXL IMU sensors is shown in the Figure 1

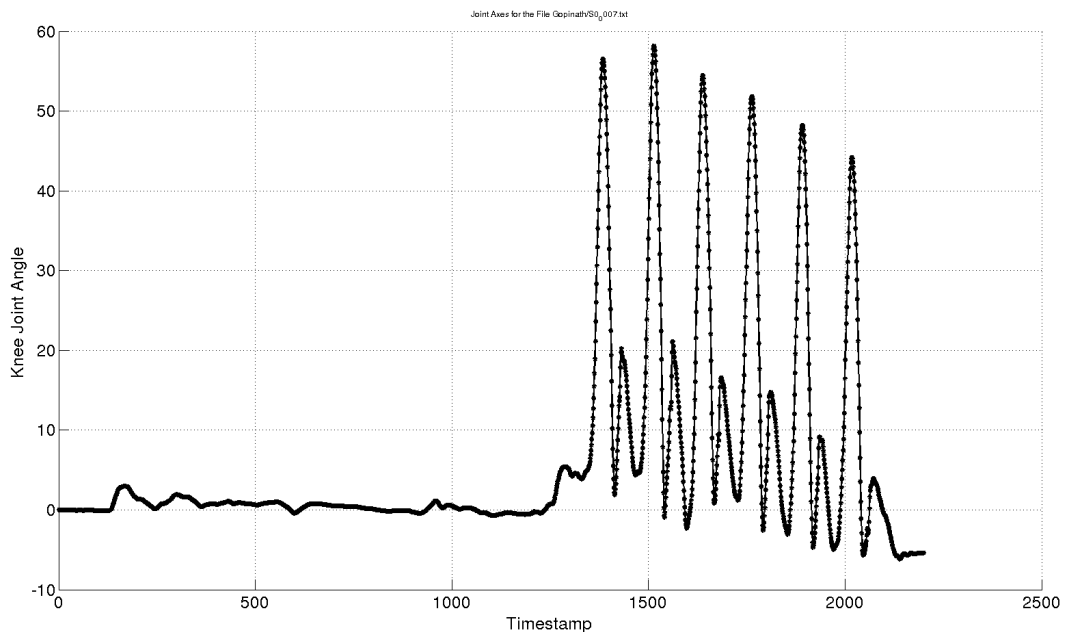


Figure 1: Knee Joint Angle Calculated using only Gyroscope Data

Data generation was started before the subject starts walking, and hence the long flat line in the beginning

The problem we are facing is that the Knee Joint angle, keeps going down with each of the step, as shown in the Figure 1. We need to figure out whether this is because of the method we are using, or there is a problem in the data sent by the Gyroscope to us.

Brief Overview on how it is calculated

According to the methods mentioned in the [1] and [2] we can mount the sensors on the thigh and the shank with any orientation. There is no requirement that on of the

principal axis of the sensor must coincide with that of the joint center axis, as shown in the Figure 2.

The first part in the calculation of the Knee joint angle is to figure out the Knee Joint center Axis and Knee Joint center coordinate with respect to local coordinate system of both of the sensors. There will be j_1 and j_2 Knee joint center axes with respect to sensor 1 and sensor 2 (the blue line, line from sensors to the center of the knee joint) and similarly o_1 and o_2 , the Knee Joint center coordinate with respect to sensor 1 and sensor 2.

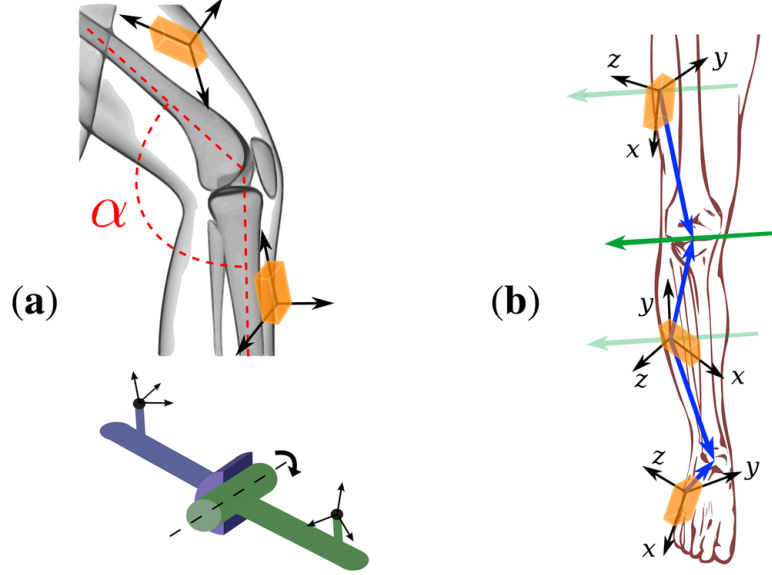


Figure 2: Figure showing the Joint Knee angle, and the Knee Joint Center Axes

Using only Gyroscope Data

We just need j_1 and j_2 to calculate knee joint angle using only Gyroscope data. Joint center coordinates are required during the calculation of the angle using Accelerometer data.

Joint center axis j_1 and j_2 can be calculated using the kinematic constraints in a Hinge Joint. $g_1(t)$ and $g_2(t)$ (The angular velocity as noted by the Gyroscope in the two sensors respectively.) differ only by the joint angle velocity and a time variant rotation matrix. Hence their projections into the joint plane (the plane to which the joint axes is the normal vector) have the same lengths for each instant in time.

$$\|g_1(t) \times j_1\|_2 - \|g_2(t) \times j_2\|_2 = 0 \quad \forall t \quad (1)$$

where $\|\cdot\|_2$ denotes the Euclidean norm.

Basically it says that if we project the angular velocity of the sensors, into the Joint plane (Plane to which Joint center axis is normal), there length should be same.

We have $g_1(t)$ and $g_2(t)$, we need to find j_1 and j_2 . For this we randomly assign some value to j_1 and j_2 and using Gauss-Newton algorithm or any other standard optimization method.

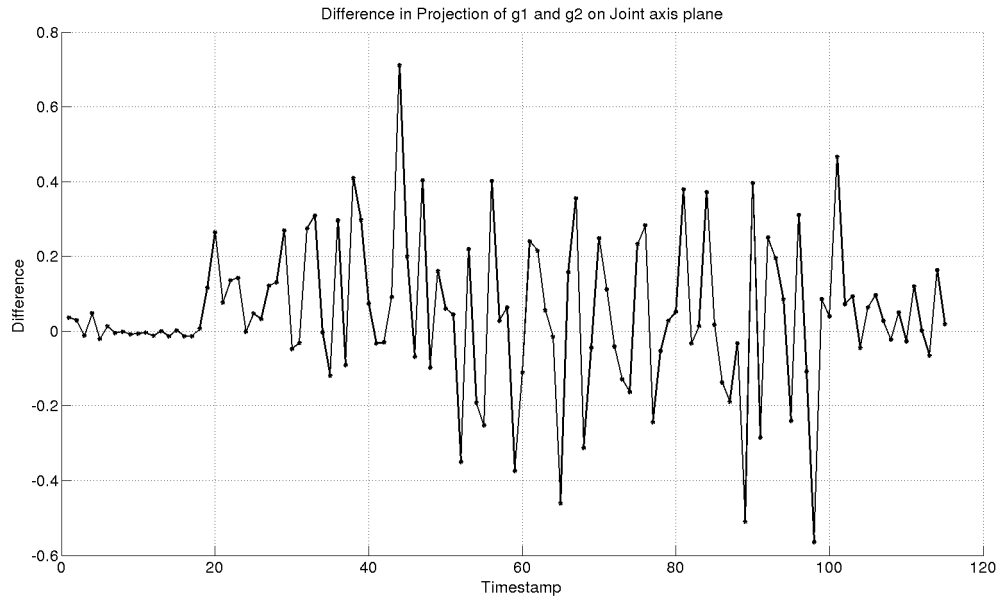


Figure 3: Difference in the projection of g_1 and g_2 on j_1 and j_2 respec.

The calibration data is used for this. After the optimization part 1 should hold (approximately). Figure 7 shows the difference.

We can see in the Figure 7 see that the difference is close to zero (it can be argued that if it is possible to improve it further), we can assume that the calibration part is working correctly, and any drift appearing in the final result does not come from here.

There was a concern that the difference here is show in radians and need to be converted to degrees for the final evaluation. However that is not correct as the graph shows the difference, as shown in the equation 1. So converting the difference to degrees is not correct.

After obtaining Joint center axis, a gyroscope-based flexion/extension angle can be calculated by integrating the difference of the angular rates around the respective joint axis, i.e.

$$\alpha_{gyr}(t) = \int_0^t (g_1(\tau) \cdot j_1 - g_2(\tau) \cdot j_2) d\tau \quad (2)$$

In Matlab this can be coded as:

```
for i=2:size(gyro_s_knee,2)
    proj1(i-1) = dor(gyro_s_knee(:,i-1), j1_valf) ;
    proj2(i-1) = dot(gyro_s_shank(:,i-1), j2_valf);
    dproj1(i-1) = radtodeg(dot(gyro_s_knee(:,i-1), j1_valf));
    dproj2(i-1) = radtodeg(dot(gyro_s_shank(:,i-1), j2_valf));

    angle_joint_axes (i) = angle_joint_axes (i-1) +...
        radtodeg((proj2(i-1) - proj1(i-1)) * t);
end
```

Problem in Data or Algorithm

The Knee joint angle that we are currently getting using only Gyroscope data, looks like this.

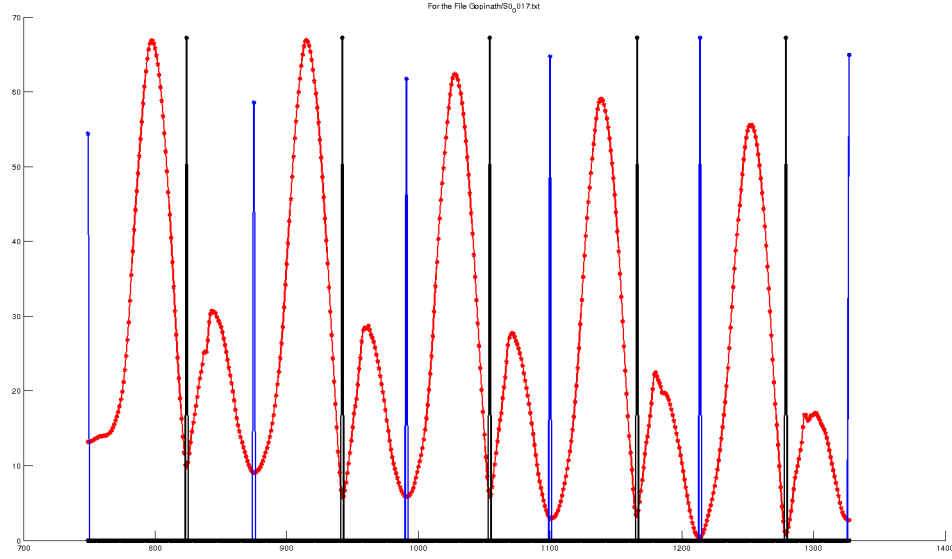


Figure 4: Walk as captured by the EXML IMU Sensors using Gyro Data. Drift is present here.

We can clearly that there is a drift in the angles, (it goes down with each step in the walk). As mentioned in [1], *the gyroscope-based angle is very precise on short time scales, but exhibits some slow drift of about 1.5 degree/sec. The drift depends on the bias of the gyroscope.*

When we manually correct the drift by adding the loss of the angles over the period of time, the walk looks like this:

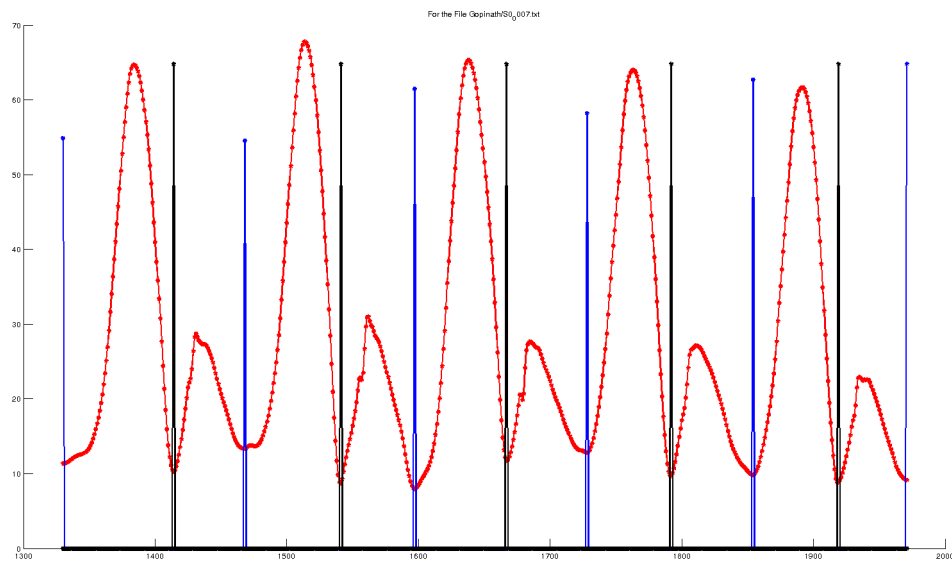


Figure 5: Walk as captured by the EXML IMU Sensors using Gyro Data. Drift free walk.

As we can clearly see that the walk is free from the drift. From this we can infer that the drift present is due to Gyroscope and not because of the algorithms used. We present some possible solutions to fix this in the next section.

Possible Solutions

Even though the paper says that the Knee angle calculated using only the Gyroscope values will be close to the actual value, they accept that sometimes the results can be affected by drift.

The hinge joint angle can be calculated by integrating the difference of both angular rates around the corresponding coordinate axis. Since even the most precise calibration will yield a non-zero bias, this calculated angle will be subject to drift.

The solution is to combine the Gyroscope angle, with a noisy, but drift less, joint angle estimate that is calculated from the measured accelerations.

Here is the angle calculated by only using the Accelerometer data.

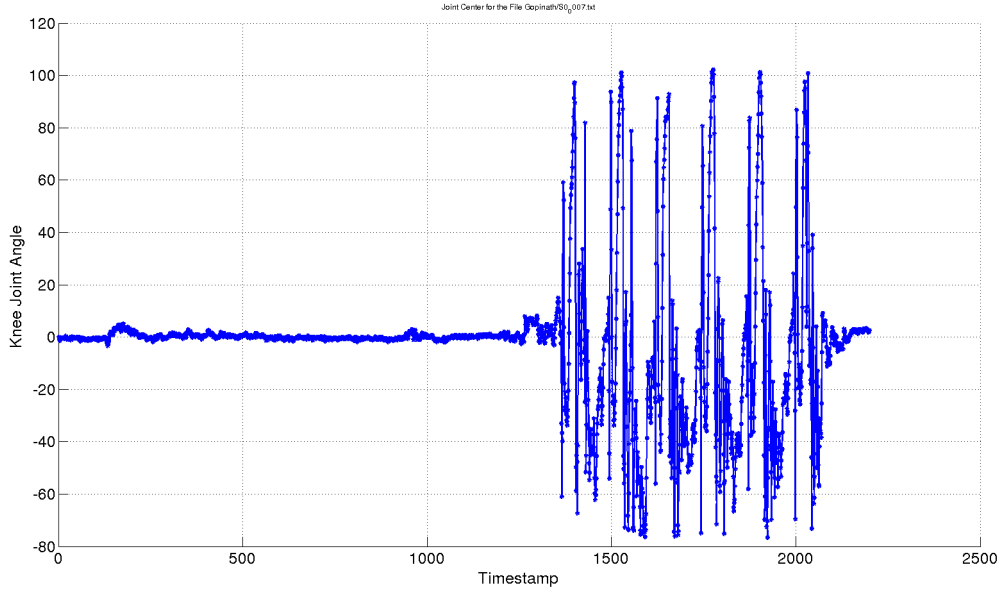


Figure 6: Noisy but drift less Knee Joint Angle using Accelerometer. Range of the angles is also higher than that the angle obtained by using the Gyroscope data.

We can see that, though the data is noisy there is no drift in it. We can combine the two angles using a complementary filter or a Kalman filter. Angles can be combined using equation 3

$$\alpha_{acc+gyr}(t) = \lambda\alpha_{acc}(t) + (1 - \lambda)(\alpha_{acc+gyr}(t - \Delta t) + \alpha_{gyr}(t) - \alpha_{gyr}(t - \Delta t)), \lambda \in [0, 1] \quad (3)$$

In Matlab this can be implemented like:

```

lambda = 0.01;
deltaT = 2; % 0.02 sec
angle_FE3(1,1) = angle_joint_axes(1,1);
angle_FE3(1,2) = angle_joint_axes(1,2);

for t = 3:data_size
    angle_FE3(1,t) = lambda*angle_joint_centre(1,t) +(1-lambda)*...
        (angle_FE3(1,t-deltaT)+angle_joint_axes(1,t)-angle_joint_axes(1,t-deltaT));
end

```

Here is a plot of the combined angle looks like:

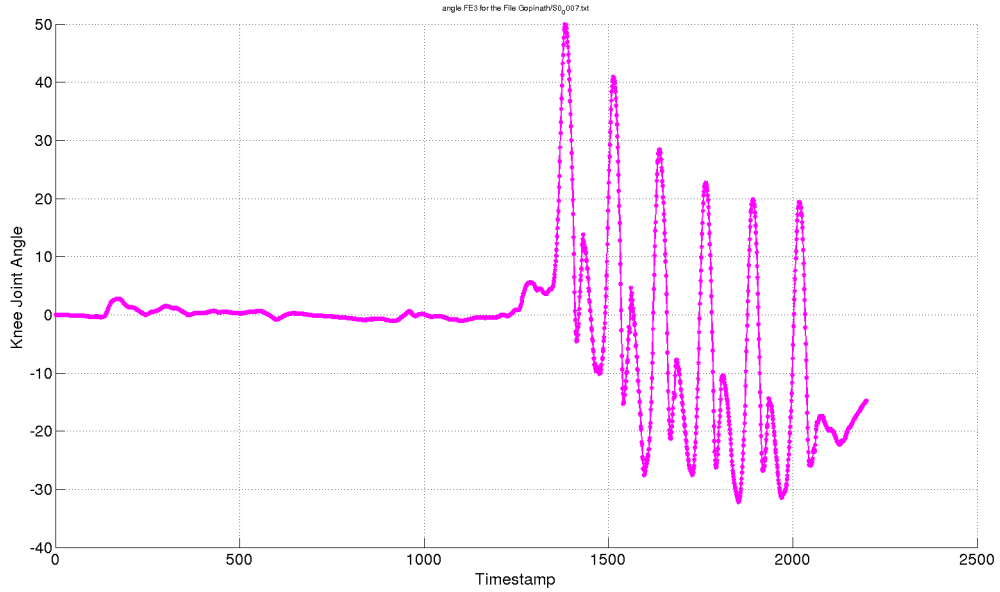


Figure 7: Combined Knee Joint Angle using $\lambda = 0.01$ and $\Delta t = 0.02$ sec

The value of the hyper-parameters λ and Δt controls the final Flexion Extension angle. We need to find the proper values of these hyper-parameters to get the desired result.

Please note that the the angle obtained by the using accelerometer in the paper [1] is less noisy, as can be seen in the Figure 8

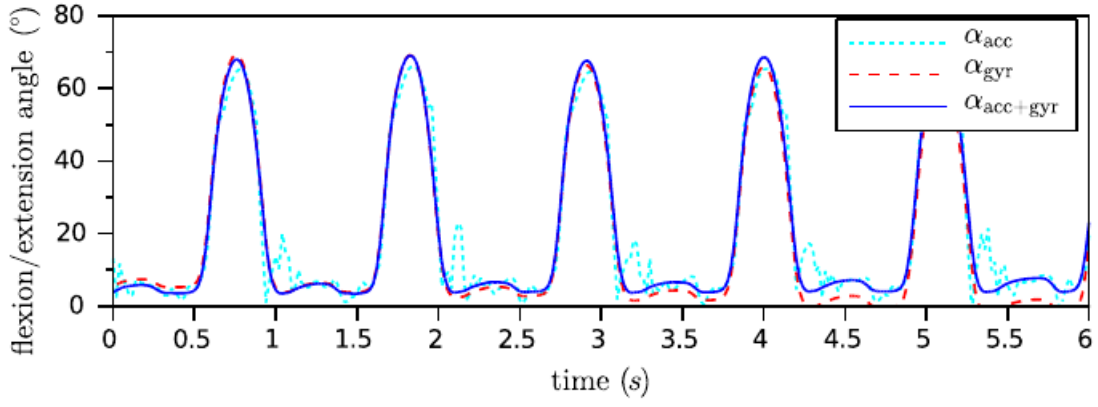


Figure 8: Knee Joint angle as obtained using, Gyrometer, Accelerometer, and a combination of both of them using the equation 3 and values of the hyper-parameters as $\lambda = 0.01$ and $\Delta t = 0.02$ sec

The noise which they are getting is very low when compared to the noise that we are getting in angles obtained from using accelerometer. We need to see that whether 3 is the correct way to proceed further or we need some other method for this purpose.

References

- [1] Thomas Seel, Jrg Raisch, and Thomas Schauer. Imu-based joint angle measurement for gait analysis. *Sensors*, 14(4):6891, 2014.
- [2] Thomas Seel, Thomas Schauer, and Jrg Raisch. Joint axis and position estimation from inertial measurement data by exploiting kinematic constraints. In *CCA*, pages 45–49. IEEE, 2012.