Attitude Estimation using low-cost MEMS Sensors

Mihir Kavishwar, 17D070004
Electrical Engineering
Indian Institute of Technology Bombay
Mumbai, India
mihirk@iitb.ac.in

***Abstract*—This report is a literature review of different techniques which use low cost microelectromechanical system (MEMS) sensors for attitude estimation of a rigid body. In addition to analysing these techniques, some MATLAB simulations were carried out to validate the algorithms. All the papers in this review attempt to solve the same attitude estimation problem but for different applications and therefore differ in their approach.**

***Index Terms*—Attitude estimation, MEMS sensors, Kalman Filter**

I. Introduction

Attitude of any rigid body describes how it is oriented in space with respect to some inertial frame of reference. There are different constructs to mathematically specify attitude. The two common constructs are Euler angles and Quaternions. Euler angles, typically denoted by (α, β, γ) or (ψ, θ, ϕ), are defined by the unique sequence of rotations undertaken by the object to reach the final orientation. Any orientation can be achieved by a rotation about a single axis and therefore it may not be intuitive to use Euler angles. Quaternions, denoted by $a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$, use this fact and capture the single axis rotation information in the form of a mathematical object which extends the idea of complex numbers.

Accurate attitude estimation is essential for a variety of applications such as in spacecrafts, aeroplanes, navigation systems, land vehicles, smartphones, fitness trackers, etc. The degree of precision required varies based the task at hand. Almost all these applications make use of a class of sensors called Microelectromechanical Systems (MEMS) to get motion data. MEMS Accelerometers measure the proper acceleration while MEMS gyroscopes measure the rate of change of orientation along primary axes in body frame. MEMS Inertial Measurement Units (IMUs) use a combination of accelerometers, gyroscopes and sometimes magnetometer, to report the same quantities. Over the past few decades, there have been many improvements in the quality of sensors. But precision comes at a cost, and in order to design low cost systems we also need to think about software level solutions which work with cheap sensors too. A recent article [8] summarises the different sensor grades and the corresponding prices. Some data from the same has shown in Table 1 and Figure 2.

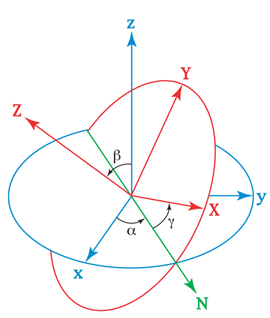

Fig. 1. Classic Euler angles geometrical definition. The xyz (fixed) system is shown in blue, the XYZ (rotated) system is shown in red. Source: Wiki

TABLE I
Bias stability by market grade. Source: Alissa Fitzgerald [8]

Grade	Bias Instability ($^o/s$)
Consumer	10
Automotive	1
Industrial	10
Tactical	1
Short-term	0.1
Navigation	0.01
Strategic	0.001

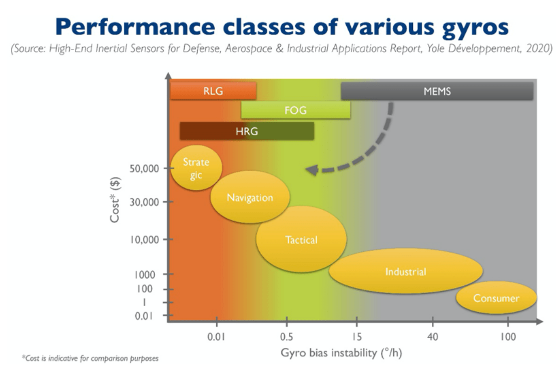

Fig. 2. IMU prices as defined by bias instability. Color indicates technology: RLG = ring-laser gyro; HRG = hemispheric resonator gyro; FOG = fiber-optic gyro. MEMS is gradually improving the grades it can achieve. Source: Yole Développement [8]

As a undergraduate student interested in robotics, the sensors that I typically use for my applications are consumer grade. Therefore, the literature that I am reviewing [2] [3] [4] [5] [6] [7] is focused on using such low cost sensors, not above automotive grade. In Section II I discuss the different kinematic models that were used in these works. In Section III I have given a brief overview on well known filtering techniques. In Section IV I have summarised how people have used this knowledge of modelling and filtering to combine different low-cost sensors and get accurate estimates. Lastly, in Section V some of my simulation results are presented.

II. System Modelling

Popular orientation determination techniques with inertial sensors include a propagating procedure with gyro sensor data and an updating procedure with accelerometer data. There are generally three principal methods to propagate the orientation information from the differential form - Euler, Direction Cosine Matrix (DCM) and Quaternion approaches [1]. Publications [2] [3] use a Quaternion based approach while [4] [6] use DCM based approach. [5] doesn't use a Kalman filter and therefore doesn't need to define a state vector. [7] assumes the system to be a Markov process and defines a different kinematic model for the system. I have included DCM equations here since they are used in my simulations.

A. Direction Cosine Matrix based modelling

Apart from slight differences in notations, all works which use DCM follow this modelling. The following set of equations are taken from [6]

Let us denote the pitch, roll and heading (euler angles) of a vehicle by θ, ϕ and ψ respectively. The relationship between the Earth and Body coordinate frames ${}^{E}\mathbf{X}$ and ${}^{B}\mathbf{X}$ can be expressed as

$$ {}^{E}\mathbf{X} = \mathbf{R}.{}^{B}\mathbf{X} $$

where $\mathbf{R}$ is the rotation matrix to rotate any vector in the body coordinate frame (${}^{B}\mathbf{X}$) into the Earth coordinate frame (${}^{E}\mathbf{X}$). This rotation matrix is given by

$$ \mathbf{R} = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} $$

where c and s denote the cos and sin operations. We observe that the last row of the rotation matrix does not contain the yaw angle and we can calculate pitch and roll angles using

$$ \phi = atan\left(\frac{R_{32}}{R_{33}}\right) $$
$$ \theta = atan\left(\frac{-R_{31}}{R_{32}/sin\phi}\right) $$

where R_{ij} denotes the $(i,j)^{th}$ element of R. In order to estimate ψ, a magnetometer sensor is also required. In typical applications we only care about pitch and roll, accelerometer and gyroscope prove to be sufficient. We define our state vector at time t as

$$ \mathbf{x}_t = \begin{bmatrix} R_{31} \\ R_{32} \\ R_{33} \end{bmatrix} $$

Note that in the above equation, the gravitational acceleration vector (which accelerometers can compute) $\mathbf{g}_t$ at any time t as measured in the body coordinate frame is given by

$$ \mathbf{g}_t = g\mathbf{x}_t $$

This gives a measurement model. Now estimation algorithms are used for to estimate this state vector by considering gyroscope measurements in process model. Sometimes gyroscope biases are also included in the state vector as given in [1] [4].

B. Comparison

It has been known that the Euler approach of propagating procedure is conceptually easy to understand but it is the most computationally expensive and the state may reach to singularity. Quaternion approach generally has the least computations with only four variables propagated. Therefore, it is very helpful in some applications which strictly demand fast computation. But, it normally uses the first order approximation for its extended Kalman Filter to deal with its nonlinear relationship so that its accuracy is traded off its computational efficiency. Conversely, the unscented Kalman filter has been used in order to improve its accuracy. In addition, Quaternion parameters have no physical interpretation about the motion. This leads to the difficulty in connecting the practical measurements with quaternion states in orientation estimator. The Direction Cosine Matrix (DCM) method of propagation transferring matrix has been known to show the performance in-between compared with Euler and Quaternion approaches. [1]

III. Filtering Algorithms

Once we have modelled our system in terms of state vector, process equation and measurement equation, we are in the domain of state estimation which is an extensively researched area in itself. The most popular estimation algorithm is the **Kalman Filter** (KF) as it gives optimal estimates in case of linear systems. It has the following 3 basic steps:

1) **Prediction Step** - Using the process model we predict the next state and covariance matrix associated with that prediction
2) **Kalman Gain Computation** - In this step we compute a term called Kalman Gain, which decides how much weight should be given to our prediction as compared to new measurement data
3) **Update Step** - On receiving a new measurement, we update our prediction of state and covariance as per the Kalman Gain computed previously, to incorporate the additional information

Steps (2) and (3) can also be considered as a single Measurement Update step.

[6] uses a KF to estimate attitude of a moving land vehicle. The main limitation of Kalman Filter is that our system needs to be linear. To overcome this limitation, **Extended Kalman Filter** (EKF) can be used. The principle of EKF is very similar to KF with they key differences being that prediction step propagates the state through a non-linear model and

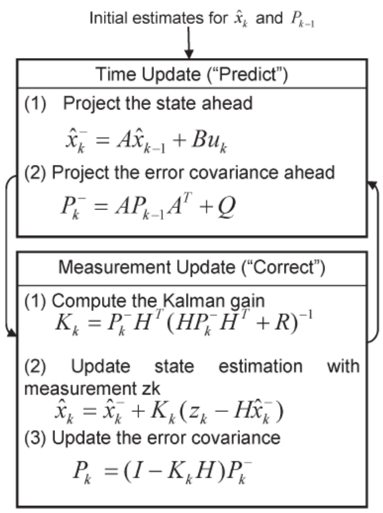

Fig. 3. Flow diagram of the time-discrete Kalman filter. At each time step k, the "Time Update" projects the current state estimation ahead in time. The "Measurement Update" adjusts the projected estimation by an actual measurement. [2]

state transistion matrices have to obtained by linearization to propogate covariances. [2] [4] and [7] use EKF to get good estimates.

Another variant of KF which has recently gained popularity is **Unscented Kalman Filter** (UKF). It is a very computation efficient algorithm and therefore can be used in latency critical applications. In UKF we don't need to calculate covariance matrix through large matrix multiplications. Instead, we generate a set of samples for our state vector, pass them through the process model and compute the variances of generated output samples. The choice of sampling is very critical and hence there are some recommended steps to be taken for generating those sample. [3] uses a UKF for estimating attitude of an unmanned aerial vehicle.

Finally, we can choose not to use any variant of KF at all and use some other idea. In [5], attitude estimation was done using a **Digital Complementary Filter**. The filter exploits the an important fact about accelerometers and gyroscopes - the former suffers from high frequency noise while the later suffers from low frequency drift. Complementary Filter combines the readings of both sensors over the frequency range that they work best individually. Figure 4 communicates this idea.

IV. Attitude Estimation Case Studies

A. *Wearable Motion Capture System*

[2] presents a modular architecture to develop a wearable system for real-time human motion capture. An IMU node called iNEMO, developed by STMicroelectronics was used in their work. iNEMO is equipped with various MEMS sensors to estimate the orientation of human body segments. In particular,

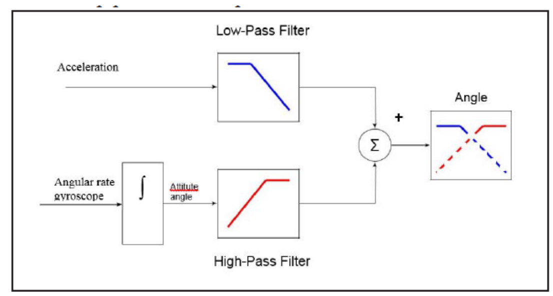

Fig. 4. Block diagram of digital complementary filter system for MEMS gyroscope and accelerometer [5]

they use a three-axis accelerometer, a three-axis magnetometer, and a three-axis gyroscope. A quaternion based EKF algorithm is used for fusing the information from all these sensors as shown in Figure 6. Compared to commercial systems, the cost of the proposed system was reduced by a factor of about eight due to an embedded design based on MEMS sensors.

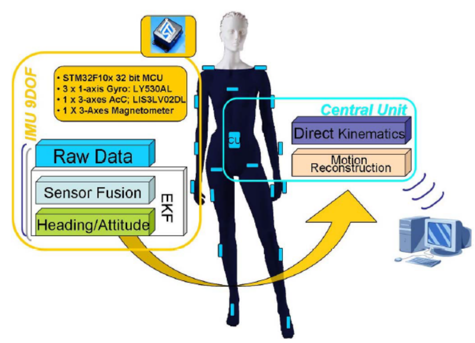

Fig. 5. System data flow: each IMU computes 3-D orientation of the relative segment and sends this information to the CU for motion reconstruction. [2]

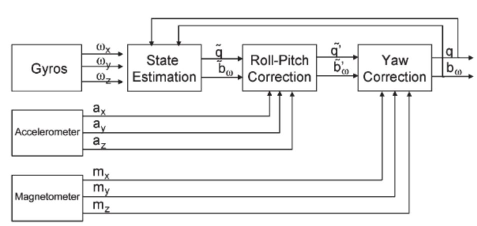

Fig. 6. Flow diagram of the quaternion-based EKF implemented [2]

B. *Rigid Body on Moving Platform*

Many outdoor activities, such as bicycling, driving, and riding segways can be modeled as rigid bodies on a moving platform. In [7], an attitude estimation scheme is established for the rigid body-moving platform system by using two gyroscopes and relative measurements between the rigid body and the platform. The proposed scheme estimates the drift-free attitudes of the rigid body and the partial drift-free absolute attitudes of the platform without using any global information or reference. The kinematic model plays an important role to obtain the drift-free estimation of the absolute attitude angles. The kinematic modelling and algorithm is quite mathematically intensive and therefore hasn't been included here. Note the important distinction here as compared to other works is that accelerometer is not used.

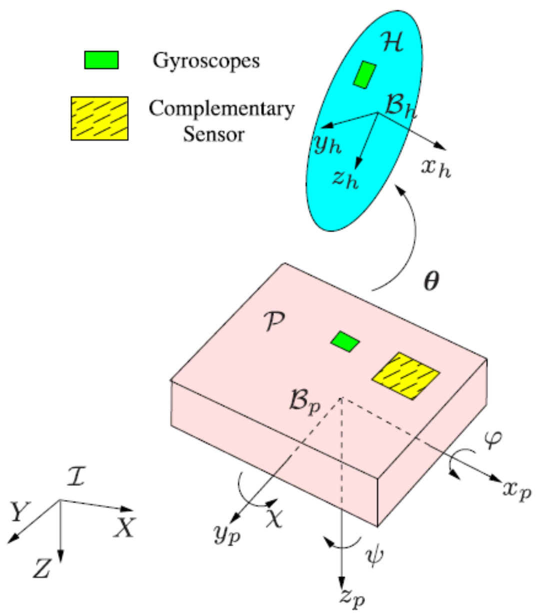

Fig. 7. Schematic diagram of rigid body-moving platform system [7]

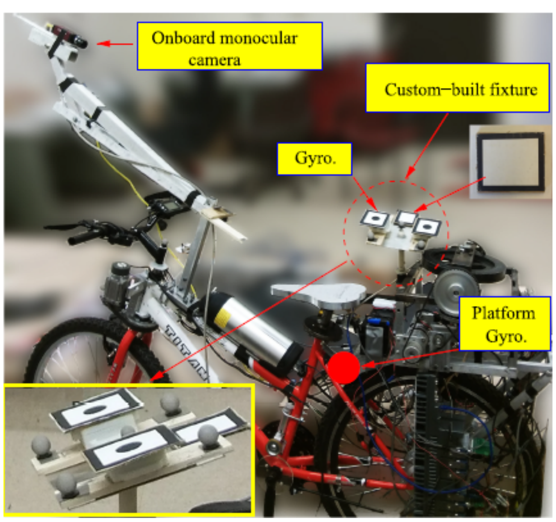

Fig. 8. Experimental setup [7]

C. Moving Land Vehicle

Currently, vehicles deploy expensive gyroscopes for attitude determination. A low-cost MEMS gyro cannot be used because of the drift problem. Typically, an accelerometer is used to correct this drift by measuring the attitude from gravitational acceleration. This is, however, not possible in vehicular applications, because accelerometer measurements are corrupted

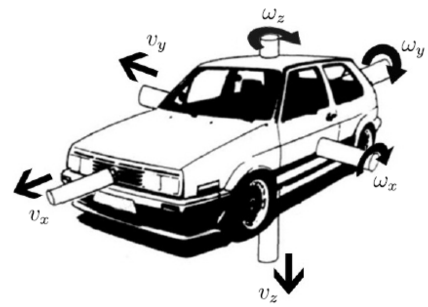

Fig. 9. Directions of linear and angular velocities [6]

by external accelerations produced due to vehicle movements. In [6], authors show that vehicle kinematics allow the removal of external accelerations from the lateral and vertical axis accelerometer measurements, thus giving the correct estimate of lateral and vertical axis gravitational accelerations. An estimate of the longitudinal axis gravitational acceleration can then be obtained by using the vector norm property of gravitational acceleration. Figure 10 shows the complete algorithm.

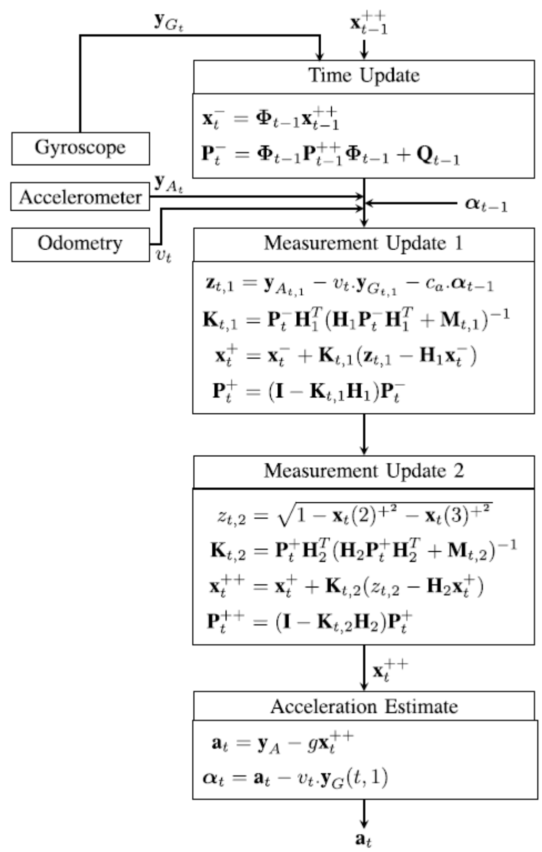

Fig. 10. Complete KF based algorithm [6]

V. Simulations

In order to compare I the performance of some of these algorithms, I implemented 3 filters in MATLAB - KF, EKF and UKF. Raw accelerometer and gyroscope measurements were taken from my smartphone when in motion. I used a DCM based model exactly as given in [4]. The paper implements an EKF but the same model can be used to implement KF and UKF as well.

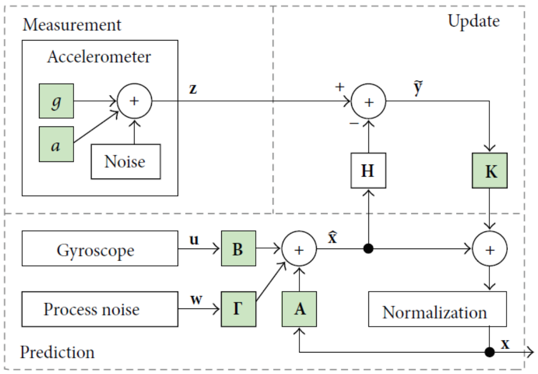

Fig. 11. A simplified block diagram of the DCM IMU filter. The covariance computation has been hidden to simplify the work flow of the EKF filter. The colored blocks are adjusted online. [4]

Noise was simulated in MATLAB to generate true states. Figures 12 and 13 show how each algorithm managed to track the true attitude.

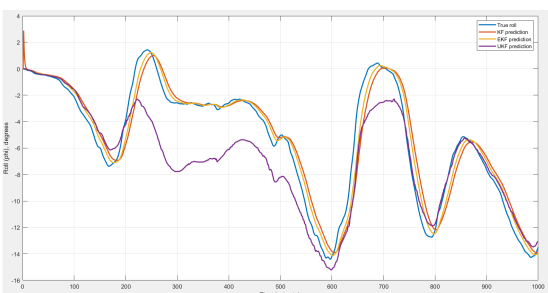

Fig. 12. Estimating Roll (ϕ)

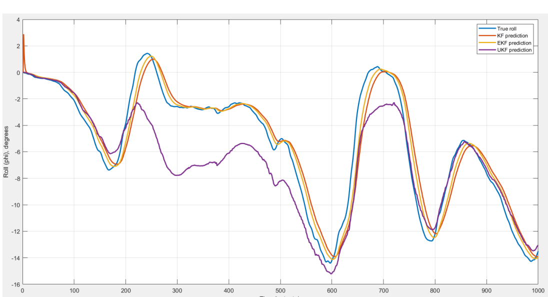

Fig. 13. Estimating Pitch (θ)

```matlab
for k=1:999
    % Linearization
    A_EKF = delf_delx(x_hat_EKF(:,k),u(:,k),freq);

    % Prediction
    x_hat_predict_EKF(:,k) = process(x_hat_EKF(:,k),u(:,k), freq); % Non Linear
    P_predict_EKF(:,:,k) = A_EKF*P_EKF(:,:,k)*A_EKF' + (Gamma^2).*Q;

    % Kalman Gain Computation
    R_EKF = meas_noise_cov(f(k+1),x_hat_predict_EKF(1:3,k),var_a, var_f, g);
    L_EKF(:,:,k) = P_predict_EKF(:,:,k)*H'/(H*P_predict_EKF(:,:,k)*H' + R_EKF);

    % Update
    e_EKF(:,k) = y(:,k+1)-H*x_hat_predict_EKF(:,k);
    x_hat_EKF(:,k+1) = x_hat_EKF(:,k) + L_EKF(:,:,k)*e_EKF(:,k);
    P_EKF(:,:,k+1) = (eye(6)-L_EKF(:,:,k)*H)*P_predict_EKF(:,:,k);

    % Normalization
    norm_mat = cat(1, cat(2, eye(3)./norm(x_hat_EKF(1:3,k+1)),zeros(3,3)),cat(2,zeros(3,3),eye(3)));
    J = norm_jacobian(x_hat_EKF(:,k+1));
    x_hat_EKF(:,k+1) = norm_mat*x_hat_EKF(:,k+1);
    P_EKF(:,:,k+1)=J*P_EKF(:,:,k+1)*J';

    %Estimated Angles
    theta_hat_EKF(k+1)=asin(-x_hat_EKF(1,k+1));
    phi_hat_EKF(k+1)=atan2(x_hat_EKF(2,k+1),x_hat_EKF(3,k+1));
end
```

Fig. 14. For loop of code implementing EKF

From Figures 12 and 13 it is evident that EKF performs better than both KF and UKF. KF performs better than UKF. The same can also be concluded from Table II.

TABLE II
Root Mean Squared Error comparison

	KF	EKF	UKF
RMSE for θ (radians)	0.0241	0.0189	0.0321
RMSE for ϕ (radians)	0.0231	0.0155	0.0428

References

[1] Nguyen Ho Quoc Phuong, Hee-Jun Kang, Young-Soo Suh, Young-Sik Ro, "A DCM Based Orientation Estimation Algorithm with an Inertial Measurement Unit and a Magnetic Compass", *Journal of Universal Computer Science*, vol. 15, no. 4 (2009), 859-876.

[2] Carmen M. N. Brigante, Nunzio Abbate, Adriano Basile, Alessandro Carmelo Faulisi, and Salvatore Sessa, "Towards Miniaturization of a MEMS-Based Wearable Motion Capture System", *IEEE Transactions on Industrial Electronics*, vol. 58, no. 8, August 2011.

[3] Hector Garcıa de Marina, Felipe Espinosa and Carlos Santos, "Adaptive UAV Attitude Estimation Employing Unscented Kalman Filter, FOAM and Low-Cost MEMS Sensors", *Sensors 2012, 12.*

[4] Heikki Hyyti, Arto Visala, "A DCM Based Attitude Estimation Algorithm for Low-Cost MEMS IMUs", *Hindawi Publishing Corporation International Journal of Navigation and Observation*, Volume 2015, Article ID 503814.

[5] Nur Hazliza Ariffin, Badariah Bais, Norhana Arsad, "Low Cost MEMS Gyroscope and Accelerometer Implementation Without Kalman Filter For Angle Estimation", *2016 International Conference on Advances in Electrical, Electronic and System Engineering*, 14-16 Nov 2016, Putrajaya, Malaysia.

[6] Hamad Ahmed and Muhammad Tahir, "Accurate Attitude Estimation of a Moving Land Vehicle Using Low-Cost MEMS IMU Sensors", *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, July 2017.

[7] Yizhai Zhang, Kehao Song, Jingang Yi, Panfeng Huang, Zhansheng Duan and Qijie Zhao, "Absolute Attitude Estimation of Rigid Body on Moving Platform Using Only Two Gyroscopes and Relative Measurements", *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, June 2018.

[8] Bryon Moyer. "Are Today's MEMS Gyros Good Enough?" Semiconductor Engineering, https://semiengineering.com/are-todays-mems-gyros-good-enough/ , October 2020.