Technical Note 2017-08-22
For CoderBunker

ATLAS

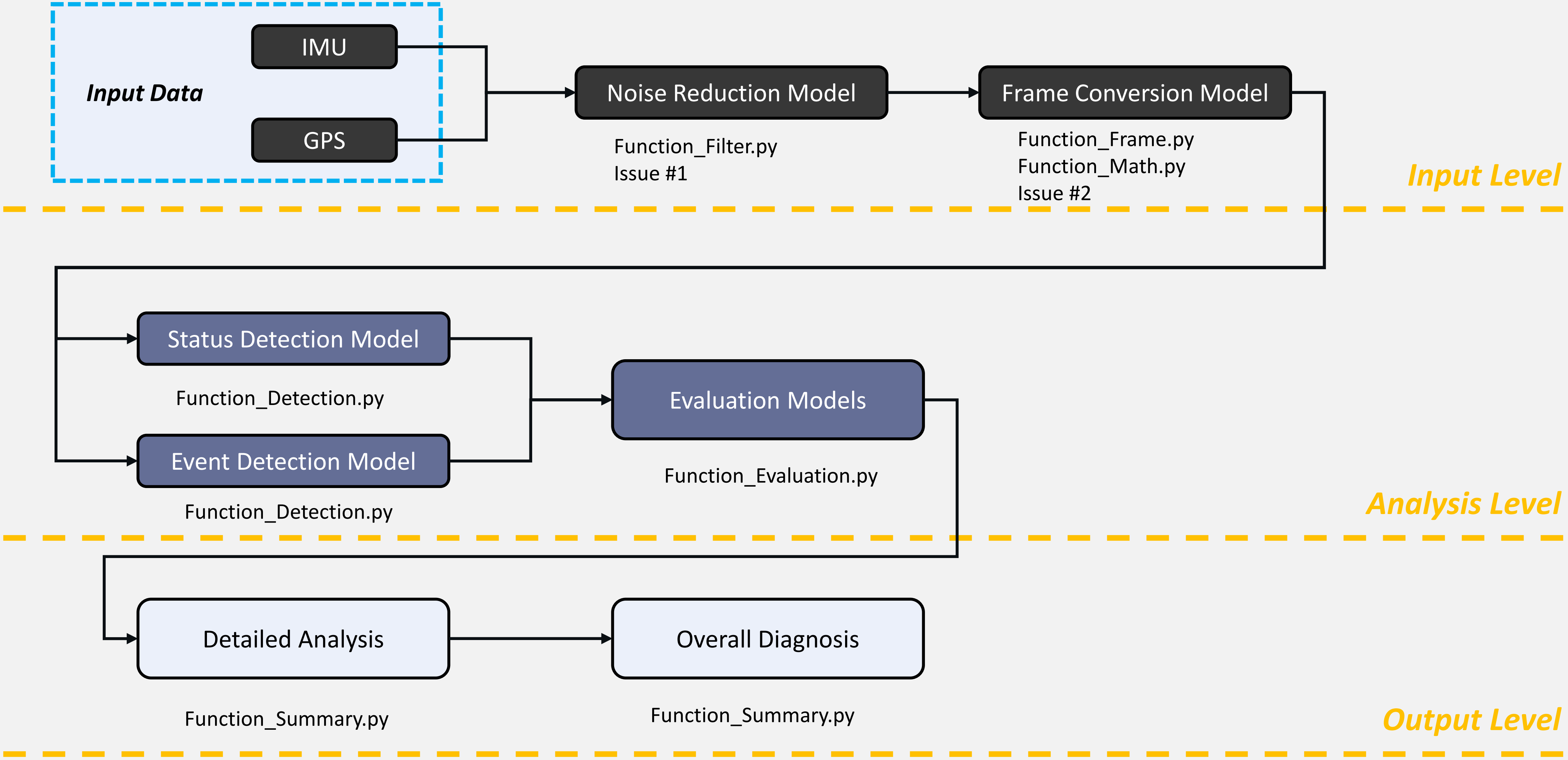
- Sean ZHOU

TABLE OF CONTENT

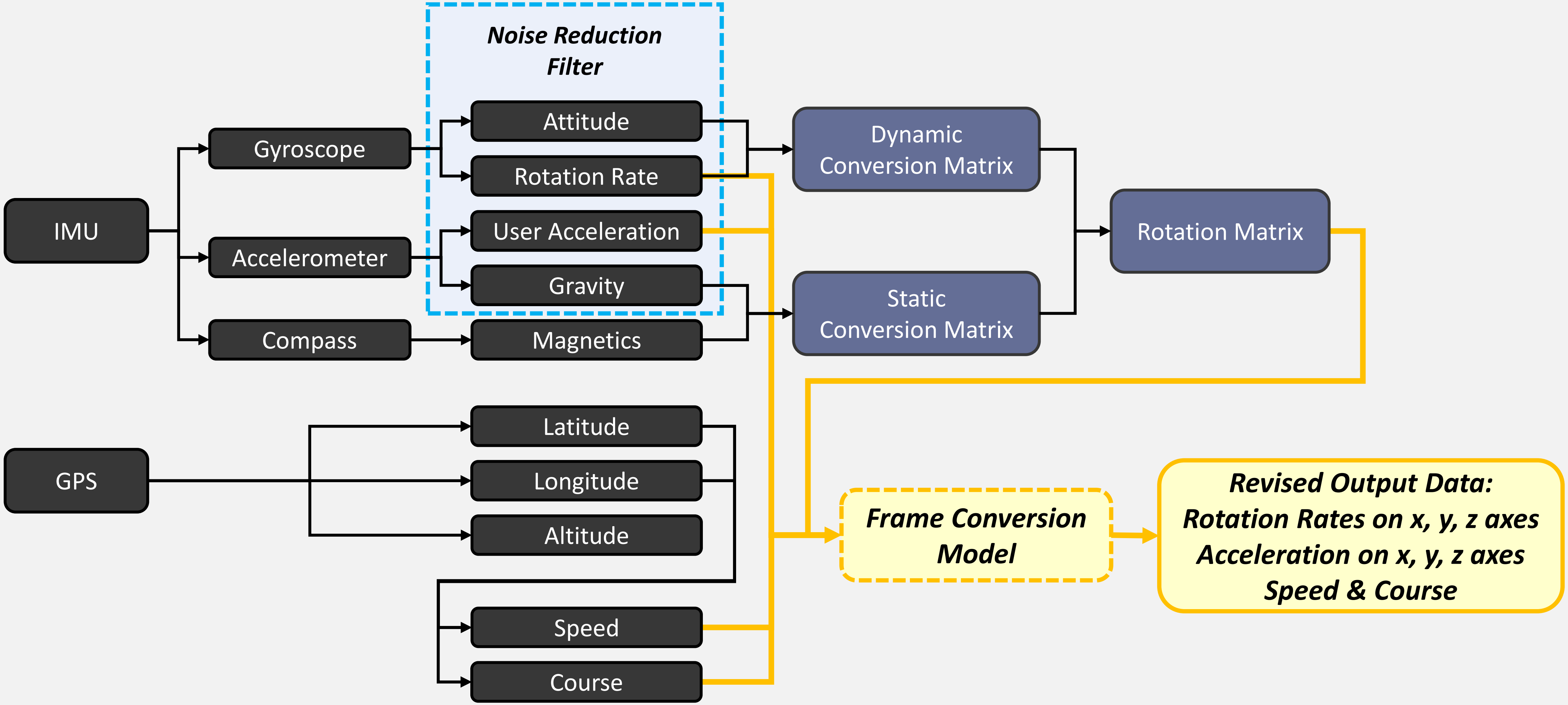
Methodology

- Model Architecture – Overview
- Model Architecture – Input Level
- Model Architecture – Analysis Level (Detection Model)
- Model Architecture – Analysis Level (Evaluation Model)
- Model Architecture – Output Level

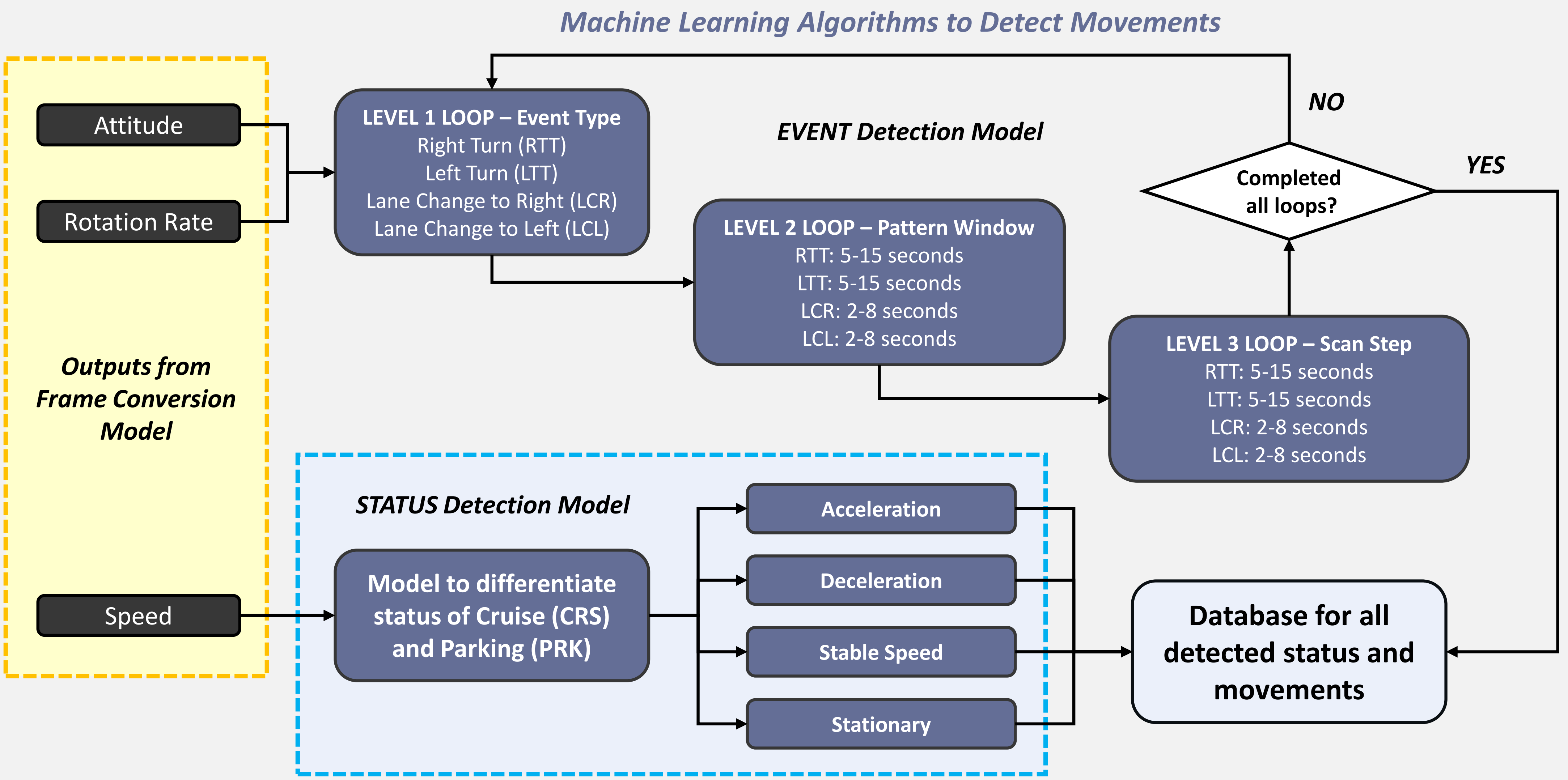
Model Architecture – Overview



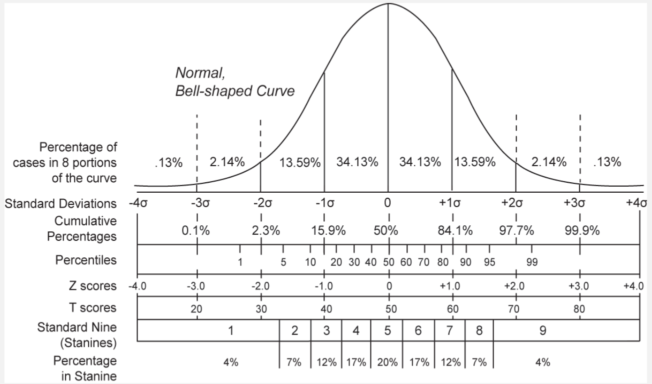
Model Architecture – Input Level



Model Architecture – Analysis Level (Detection Models)

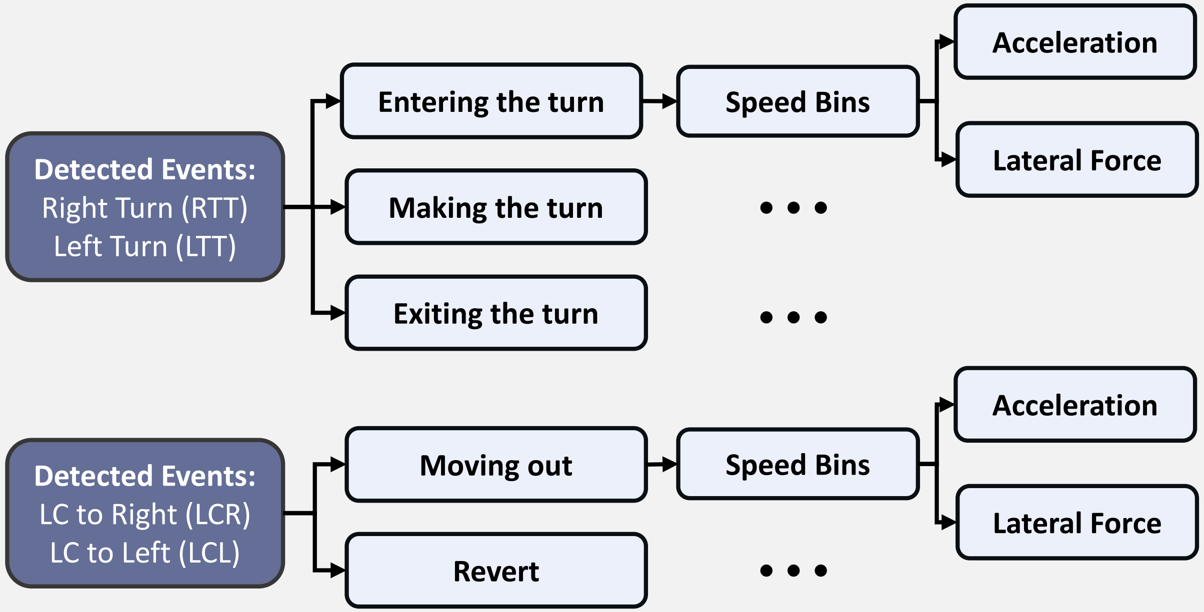


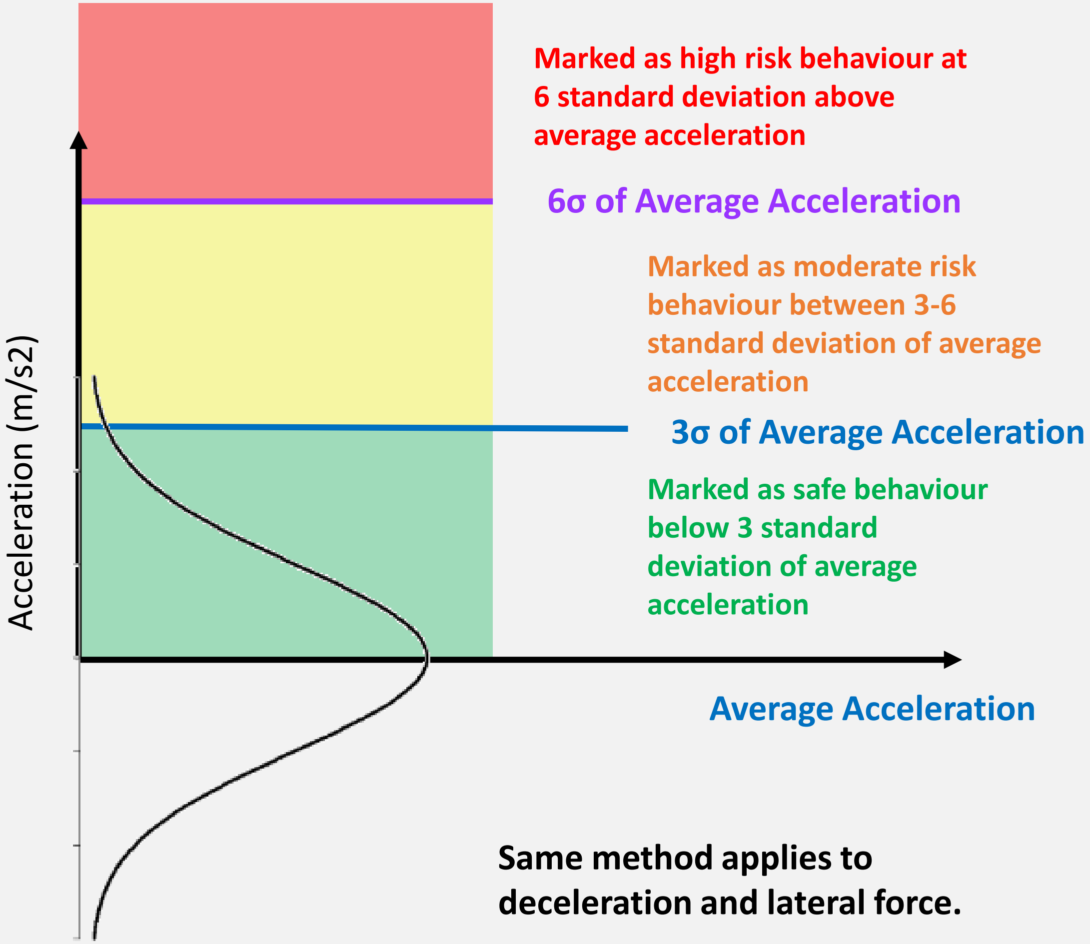
Model Architecture – Analysis Level (Evaluation Models)



Anomaly detection (also outlier detection) is the identification of items, events or observations which do not conform to an expected pattern or other items in a dataset.

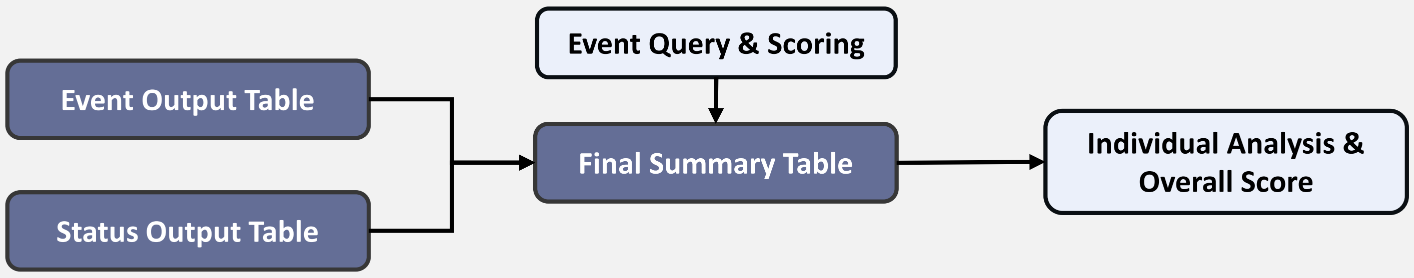
A common practice is to use Gaussian (Normal) Distribution. Any value outside 3 standard deviation can be considered as anomaly, which accounts for 0.13% for each side.



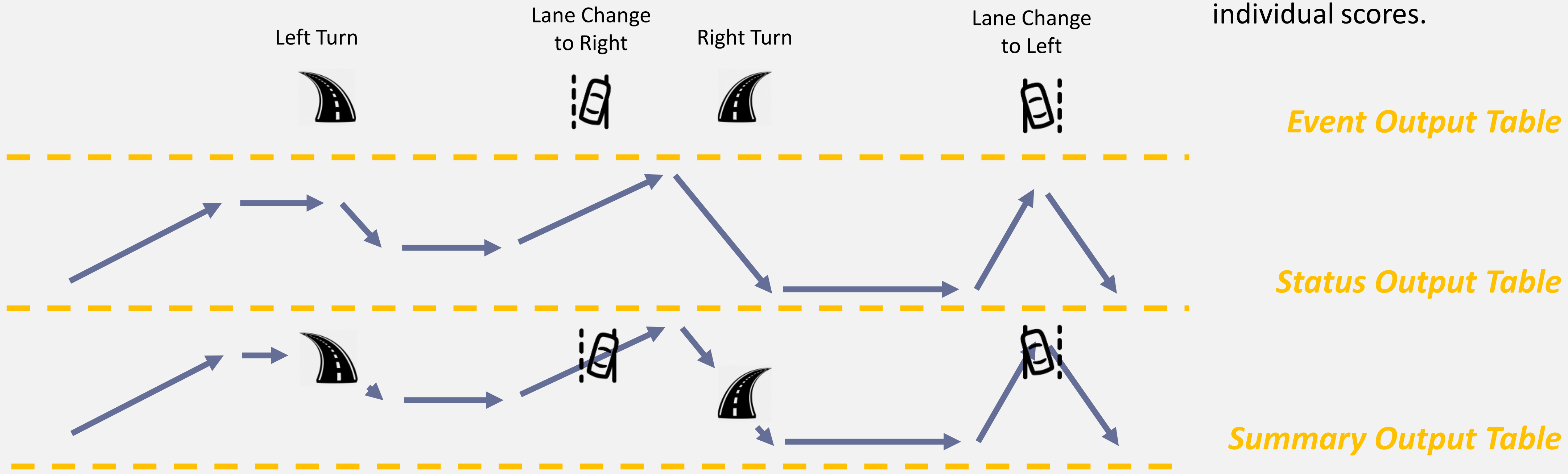


Same method applies to deceleration and lateral force.

Model Architecture – Output Level



- At output level, output tables are merged to a summary output table.
- Individual events are scored and can be queried.
- Final score are aggregated from individual scores.



Technical Issues

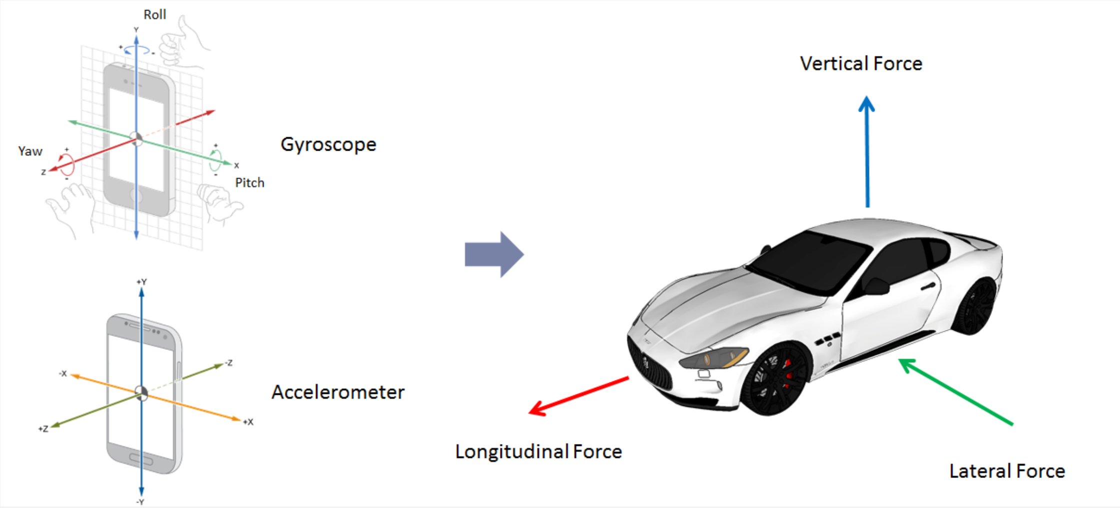
- Frame Conversion
- Filter

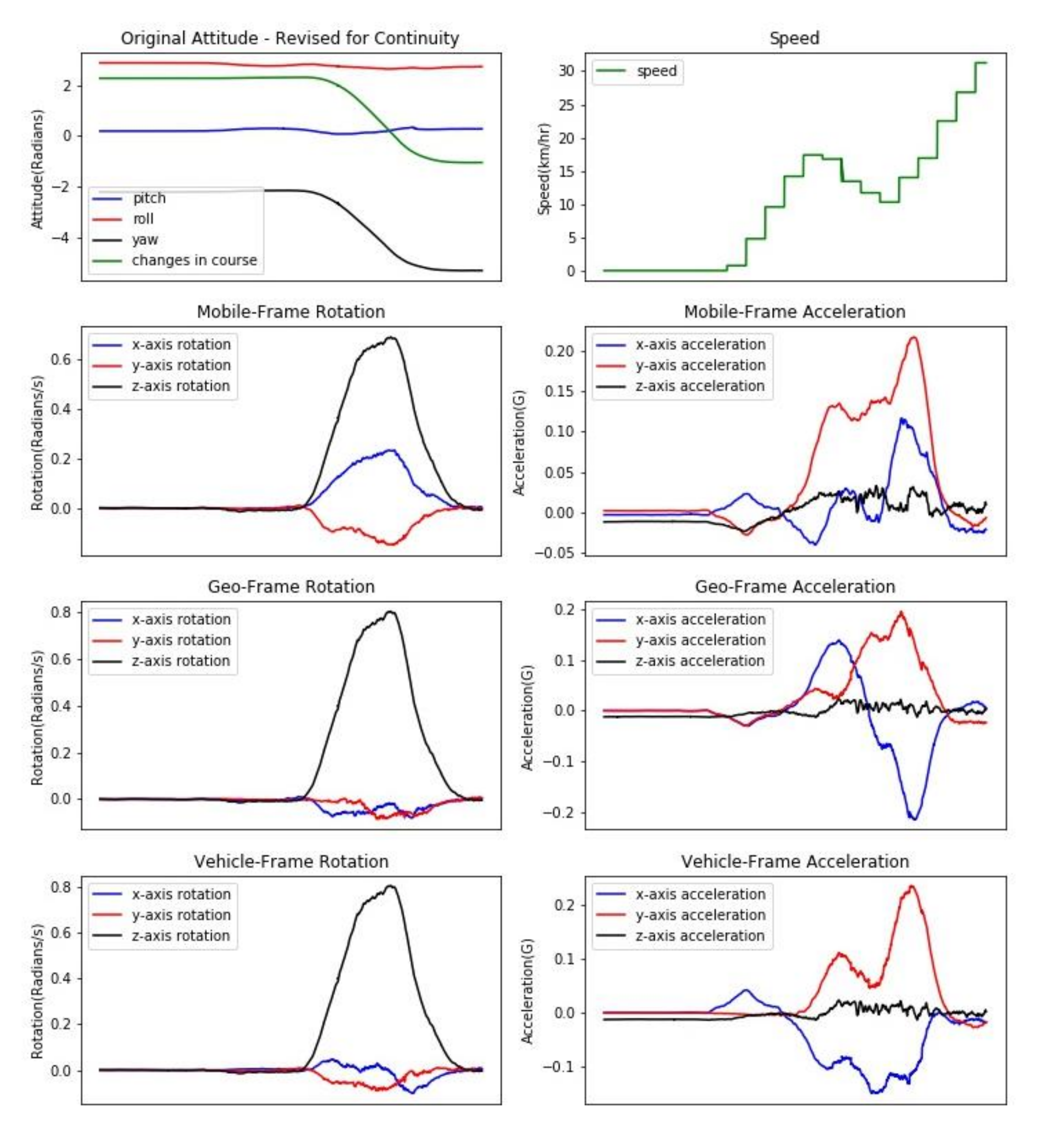
Issue #1: Frame conversion is implemented based on a few papers. The output can be used to identify particular movements without any problems. However, the longitudinal force (i.e. acceleration) and lateral force may not be always consistent with the speed. E.g. when speed goes up, acceleration should be greater than zero, which may not be the case after conversion.

Issue #2: For prototype purpose, moving average is used to reduce the noise. The proper method should apply some sort of filter (e.g. Kalman Filter). However, the proper values of covariance matrix remains an issue. In addition, based on the papers, filter should be imbedded within frame conversion implementation. It is now separated into two process.

The following slides set out the basic description of frame conversion and the last one shows inconsistency as mentioned in Issue #1.

Methodology – Data Processing & Frame Conversion





Frame Conversion:

The sensors' data was collected when the phone was placed in a random orientation. To better understand the longitudinal and lateral forces of the vehicle, it is essential to convert the data from device-frame to vehicle-frame.

Frame Conversion and Noise Reduction

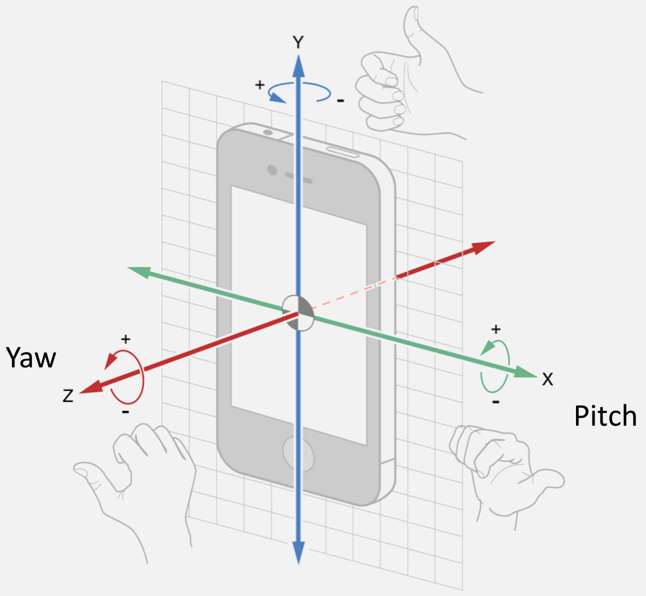
References:

1) 2016 Orientation and Displacement Detection for Smartphone Device Based IMUs
2) 2014 Use It Free: Instantly Knowing Your Phone Attitude
3) 2015 Heading Estimation for Indoor Pedestrian Navigation Using a Smartphone in the Pocket

Methodology:

- The three aforementioned thesis try to solve the same problem – how to identify and estimate the phone orientation and detect the direction that the phone moves. These thesis share similar methodologies in the following steps:
  - Static conversion – assuming the phone is static, use gravity and magnetic data to convert device frame to geodetic frame through rotation matrix.
  - Dynamic adjustment – since the phone is on the move, the direction of gravity is not exactly downwards. Gyroscope data is used to make the estimation through ways:
    1) Thesis 1 – calculate rotation matrix from Euler Angles (suffer from Gimbal Lock);
    2) Thesis 2 – convert to quaternion and calculate integral of rotation rate to get the changes in rotation through Runge-Kutta Method;
    3) Thesis 3 – convert to quaternion and use PCA to identify walking direction.
  - All methods use some form of filter to reduce noise in the data. The common filters include Kalman Filter, Extended Kalman Filter, Complementary Filter and etc.
- The technical detail is laid out as below.

Rigid Body Rotation - Basics



For a smartphone:

- Y-axis represents pitch – θ (Theta)
- Y-axis represents roll – φ (Phi)
- Z-axis represents yaw – ψ (Psi)

Rotation matrix for x-axis:

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c\theta & s\theta \\ 0 & -s\theta & c\theta \end{pmatrix}$$

Rotation matrix for y-axis:

$$R_y(\varphi) = \begin{pmatrix} \cos(\varphi) & 0 & -\sin(\varphi) \\ 0 & 1 & 0 \\ \sin(\varphi) & 0 & \cos(\varphi) \end{pmatrix} = \begin{pmatrix} c\varphi & 0 & -s\varphi \\ 0 & 1 & 0 \\ s\varphi & 0 & c\varphi \end{pmatrix}$$

Rotation matrix for x-axis:

$$R_z(\psi) = \begin{pmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} c\psi & s\psi & 0 \\ -s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Note: The above rotation matrices are expressed as coordinate rotation.

Rigid Body Rotation – Rotation Matrix

Sequence 312: $R_z(\psi)R_x(\theta)R_y(\varphi)$ Yaw – Pitch – Roll

Conversion Matrix: $R_I^B(\psi,\theta,\varphi) = R_{v2}^B(\psi)R_{v1}^{v2}(\theta)R_I^{v1}(\varphi)$

Actual rotation sequence: Roll -> Pitch -> Yaw

Converting from inertial frame to body frame: $z_B = R_I^B z_I$

Reverse Conversion: $R_B^I(\psi,\theta,\varphi) = R_I^{v1}(-\varphi)R_{v1}^{v2}(-\theta)R_{v2}^B(-\psi) = R_I^B(\psi,\theta,\varphi)^T$

Converting from body frame to inertial frame: $z_I = {R_I^B}^T z_B$

$$R_I^B(\psi,\theta,\varphi) = R_{v2}^B(\psi)R_{v1}^{v2}(\theta)R_I^{v1}(\varphi)$$

$$= \begin{pmatrix} c\psi & s\psi & 0 \\ -s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & c\theta & s\theta \\ 0 & -s\theta & c\theta \end{pmatrix} \begin{pmatrix} c\varphi & 0 & -s\varphi \\ 0 & 1 & 0 \\ s\varphi & 0 & c\varphi \end{pmatrix}$$

$$= \begin{pmatrix} c\psi c\varphi + s\psi s\theta s\varphi & s\psi c\theta & -c\psi s\varphi + s\psi s\theta c\varphi \\ -s\psi c\varphi + c\psi s\theta s\varphi & c\psi c\theta & s\psi s\varphi + c\psi c\theta c\varphi \\ c\theta s\varphi & -s\theta & c\theta c\varphi \end{pmatrix}$$

Rigid Body Rotation – Quaternion

Euler Angles to Quaternion:

$$q_{312}(\psi, \theta, \varphi) = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} c\left(\frac{\psi}{2}\right) c\left(\frac{\theta}{2}\right) c\left(\frac{\varphi}{2}\right) + s\left(\frac{\psi}{2}\right) s\left(\frac{\theta}{2}\right) s\left(\frac{\varphi}{2}\right) \\ c\left(\frac{\psi}{2}\right) s\left(\frac{\theta}{2}\right) c\left(\frac{\varphi}{2}\right) + s\left(\frac{\psi}{2}\right) c\left(\frac{\theta}{2}\right) s\left(\frac{\varphi}{2}\right) \\ c\left(\frac{\psi}{2}\right) c\left(\frac{\theta}{2}\right) s\left(\frac{\varphi}{2}\right) - s\left(\frac{\psi}{2}\right) s\left(\frac{\theta}{2}\right) c\left(\frac{\varphi}{2}\right) \\ -c\left(\frac{\psi}{2}\right) s\left(\frac{\theta}{2}\right) s\left(\frac{\varphi}{2}\right) + s\left(\frac{\psi}{2}\right) c\left(\frac{\theta}{2}\right) c\left(\frac{\varphi}{2}\right) \end{bmatrix}$$

Quaternion to Euler Angles:

$\theta = -\arcsin(2q_2q_3 - 2q_0q_1)$

$\varphi = arctan2(2q_1q_3 + 2q_0q_2, q_0^2 - q_1^2 - q_2^2 + q_3^2)$

$\psi = arctan2(2q_1q_2 + 2q_0q_3, q_0^2 - q_1^2 + q_2^2 - q_3^2)$

Rigid Body Rotation – Integration of Quaternion

Quaternion derivative is related to angular velocity.

Angular velocity: $\dot{q} = \frac{dq}{dt} = \frac{1}{2}\omega q$ => Change in rotation: $\int \left(\frac{1}{2}\omega q\right) dt$

In order to solve the integration, Runge-Kutta Method is applied.

$$\frac{dq}{dt} = \frac{1}{2}\begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix}\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} \omega_z q_1 - \omega_y q_2 + \omega_x q_3 \\ -\omega_z q_0 + \omega_x q_2 + \omega_y q_3 \\ \omega_y q_0 - \omega_x q_1 + \omega_z q_3 \\ -\omega_x q_0 - \omega_y q_1 - \omega_z q_2 \end{bmatrix} = q_t$$

$$K_1 = f(t, q_t)$$

$$K_2 = f\left(t + \frac{1}{2}h, q_t + \frac{1}{2}K_1 h\right)$$

$$K_3 = f\left(t + \frac{1}{2}h, q_t + \frac{1}{2}K_2 h\right)$$

$$K_4 = f(t + h, q_t + K_3 h)$$

$$q_{t+1} = q_t + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)h$$

Rigid Body Rotation – Quaternion to Rotation Matrix

Quaternion:

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = [q_0 \quad q_1 \quad q_2 \quad q_3]^T$$

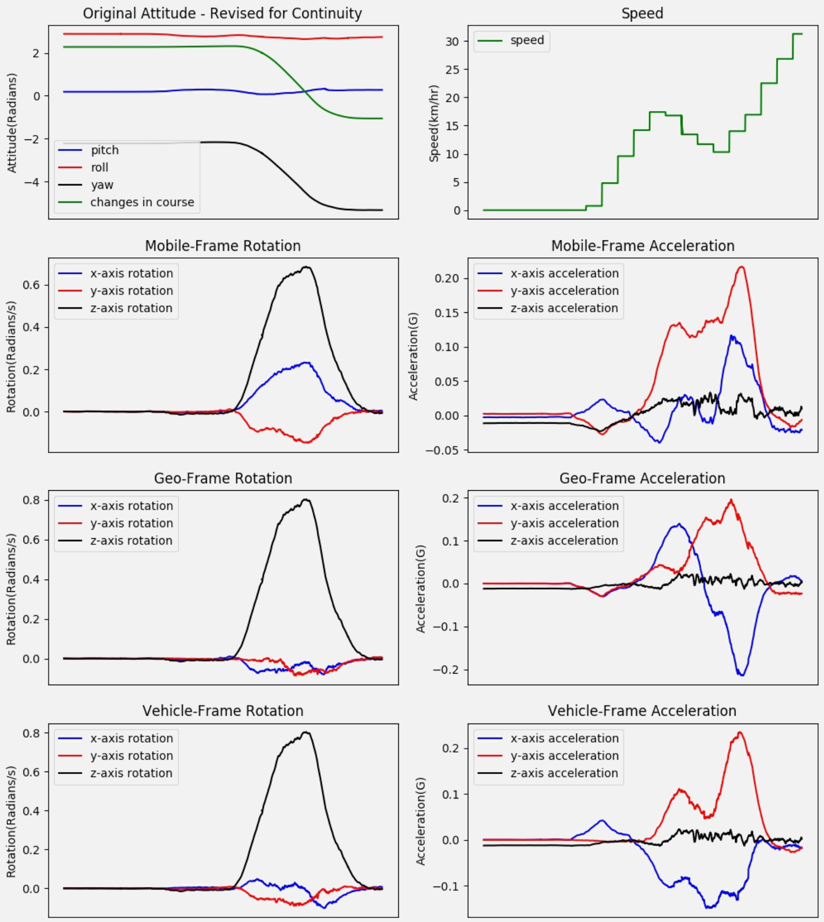
Convert quaternion to rotation matrix:

$$R_q(q) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 - 2q_0q_2 \\ 2q_1q_2 - 2q_0q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 + 2q_0q_1 \\ 2q_1q_3 + 2q_0q_2 & 2q_2q_3 - 2q_0q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

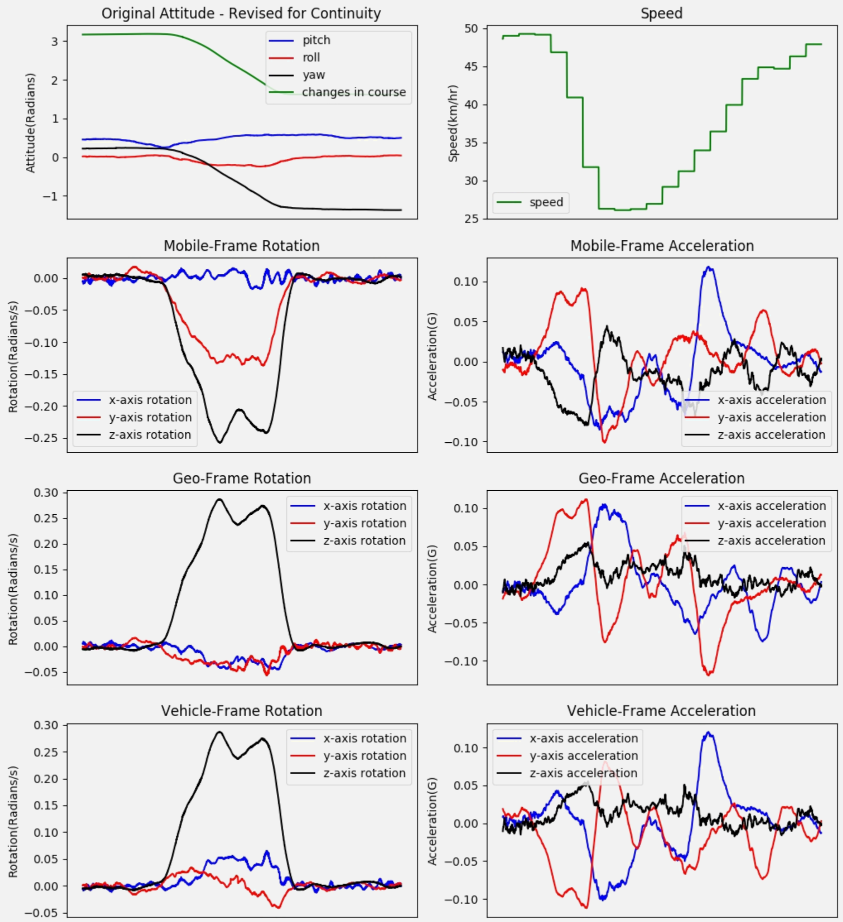
Converting from inertial frame to body frame: $z_B = R_q(q) z_I$

Converting from body frame to inertial frame: $z_I = R_q(q)^T z_B$

Frame Implementation Errors



Test Data 1: y-axis acceleration represents longitudinal force of a vehicle. In this case, it is more or less in alignment with the speed.



Test Data 2: In this case, when speed goes up, y-axis accelerations are below zero, which is inconsistent.