

Attitude Estimation Using a Quaternion-Based Kalman Filter With Adaptive and Norm-Based Estimation of External Acceleration

Livio Bisogni

Anno Accademico 2020-2021



UNIVERSITÀ DI PISA

SCUOLA DI INGEGNERIA

DIPARTIMENTO DI
INGEGNERIA
DELL'INFORMAZIONE

Corso di Laurea Magistrale
in Ingegneria Robotica e
dell'Automazione

SISTEMI DI GUIDA E
NAVIGAZIONE

Overview

- Quaternions
 - Definition
 - Axis-Angle Representation and Quaternions of Rotation
 - Rotation Matrix (from Quaternion)
 - Quaternion (from Rotation Matrix)
 - Derivative and Multiplication
- Sensor Models
 - Gyroscope Model
 - Accelerometer Model
 - Magnetometer Model
 - Measurement Noises
 - Sensors Using Simulated Data
 - Sensors Using Real Data

Overview

- Indirect Kalman Filter
 - Introduction
 - Orientation Error Quaternion
 - State Equation
 - Propagation Stage ("Prediction")
 - Accelerometer Measurement Update Stage ("1st Correction")
 - Estimation of External Acceleration Covariance Matrix: [Sabatini]
 - Estimation of External Acceleration Covariance Matrix: [Suh]
 - Magnetometer Measurement Update Stage ("2nd Correction")
- Performance Metrics
 - Quaternion Integrations
 - Initial ($k=0$) Estimation of Euler Angles & Quaternions
 - Rotation Matrices
 - Euler Angles
 - RMSEs

Overview

- Simulated Data
 - Time Params & Sensor Characterization
 - Initial Euler Angles, State Vector and State Covariance Matrix
 - Norm-Based and Adaptive Algorithm Params & Process Noise Covariance Matrix
 - Data Generation: Angular Velocity
 - Data Generation: External Acceleration
 - Data Generation: Acceleration
 - Data Generation: Acceleration Error
 - Data Generation: Magnetic Field

Overview

- Simulated Data Analysis
 - Quaternion Estimation
 - Quaternion Estimation Error
 - Gyroscope Bias Estimation
 - Angular Velocity Estimation
 - Angular Velocity Estimation Error
 - Accelerometer Bias Estimation
 - Internal Acceleration Estimation
 - Internal Acceleration Estimation Error
 - External Acceleration Estimation
 - External Acceleration Estimation Error
 - Magnetic Field Estimation
 - Magnetic Field Estimation Error

Overview

- Simulated Data Analysis [continued]
 - Euler Angles Estimation [Sabatini]
 - Euler Angles Estimation Error [Sabatini]
 - Euler Angles Estimation [Suh]
 - Euler Angles Estimation Error [Suh]
 - Process Noise Covariance Matrix: Suh vs. Sabatini
 - A More Extreme Example: Angular Velocity
 - A More Extreme Example: Euler Angles Estimation [Sabatini]
 - A More Extreme Example: Euler Angles Estimation Error [Sabatini]
 - A More Extreme Example: Euler Angles Estimation [Suh]
 - A More Extreme Example: Euler Angles Estimation Error [Suh]

Overview

- Real Data
 - Time Params & Sensor Characteristics
 - Initial State Vector and State Covariance Matrix
 - Norm-Based and Adaptive Algorithm Parameters & Process Noise Cov. Matrix
 - Time Step Discontinuities
- Real Data Analysis
 - Angular Velocity
 - Acceleration
 - Magnetic Field
 - Gyroscope Bias Estimation
 - Accelerometer Bias Estimation
 - Euler Angles Estimation [Sabatini]
 - Euler Angles Estimation Error [Sabatini]
 - Euler Angles Estimation [Suh]
 - Euler Angles Estimation Error [Suh]
- Conclusions

References

1. Suh, Y. S. Orientation Estimation Using a Quaternion-Based Indirect Kalman Filter With Adaptive Estimation of External Acceleration. *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 12, 3296-3305, Dec. 2010. doi:10.1109/TIM.2010.2047157.
2. Trawny, N.; Roumeliotis, S.I. Indirect Kalman filter for 3D attitude estimation. In Technical Report; Department of Computer Science and Engineering, University of Minnesota: Minneapolis, MN, USA, 2005.
3. Nez, A.; Fradet, L.; Marin, F.; Monnet, T.; Lacouture, P. Identification of Noise Covariance Matrices to Improve Orientation Estimation by Kalman Filter. *Sensors* 2018, 18, 3490. doi:10.3390/s18103490
4. Sabatini, A. M. Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing. *IEEE Trans. Biomed. Eng.*, vol. 53, no. 7, 1346–1356, Jul. 2006.
5. Schneider, R; Georgakis, C. How To NOT Make the Extended Kalman Filter Fail. *Industrial & Engineering Chemistry Research* 2013 52 (9), 3354-3362. doi:10.1021/ie300415d
6. Bisogni, L.; Mollaiyan, R.; Pettinari, M.; Neri, P.; Gabiccini, M. Automatic Calibration of a Two-Axis Rotary Table for 3D Scanning Purposes. *Sensors* 2020, 20, 7107. doi:10.3390/s20247107

Quaternions

Definition

The quaternion q is defined as:

$$q = q_0 + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

where:

- q_0, q_1, q_2 , and q_3 are real numbers
- \mathbf{i}, \mathbf{j} , and \mathbf{k} are hyperimaginary numbers satisfying:

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$$

The quaternion can also be written in a four-dimensional column vector:

$$q = \begin{bmatrix} q_0 \\ \mathbf{q} \end{bmatrix}$$

where:

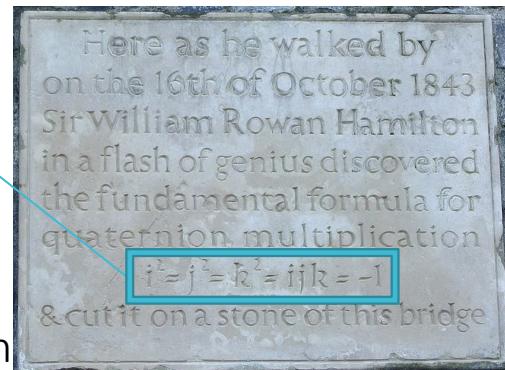
- q_0 is the real or scalar part of the quaternion
- $\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$ is the imaginary or vector part of the quaternion

Note that this does correspond to the Hamilton notation. Other authors prefer instead to use the JPL Proposed Standard Conventions:

$$q = q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k} + q_4 = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix}$$

$$\mathbf{ijk} = +1$$

All of the following formulas involving quaternions are valid with the Hamilton convention only.



Quaternions

Axis-Angle Representation and Quaternions of Rotation

If the quantities q_0 and \mathbf{q} fulfill:

$$q_0 = \cos(\theta/2), \quad \mathbf{q} = \begin{bmatrix} k_x \sin(\theta/2) \\ k_y \sin(\theta/2) \\ k_z \sin(\theta/2) \end{bmatrix} = \mathbf{k} \sin(\theta/2)$$

the elements q_0 , q_1 , q_2 , and q_3 are called quaternion of rotation or Euler symmetric parameters. In this notation, the unit vector \mathbf{k} indicates the direction of an axis of rotation and the angle θ describes the magnitude of the rotation about the axis.

The quaternion of rotation is a unit quaternion, i.e., satisfying:

$$|q| = \sqrt{qq^*} = \sqrt{q^*q} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1$$

where $q^* = q_0 - q_1\mathbf{i} - q_2\mathbf{j} - q_3\mathbf{k}$ is the conjugate of q .

Henceforth, the term quaternion will refer to a quaternion of rotation.

Quaternions

Rotation Matrix (from Quaternion)

The quaternion q is used as an orientation quaternion in the navigation frame. A point p_n in the navigation frame ($\{n\}$) and a point p_b in the body frame ($\{b\}$) are related as follows:

$$p_b = C_n^b p_n = C(q) p_n$$

where the rotation matrix $C(q)$ is defined by:

$$C(q) = \begin{bmatrix} 2q_0^2 + 2q_1^2 - 1 & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 - 2q_0q_2 \\ 2q_1q_2 - 2q_0q_3 & 2q_0^2 + 2q_2^2 - 1 & 2q_2q_3 + 2q_0q_1 \\ 2q_1q_3 + 2q_0q_2 & 2q_2q_3 - 2q_0q_1 & 2q_0^2 + 2q_3^2 - 1 \end{bmatrix}$$

Since q is a quaternion of rotation, i.e., it is a unit quaternion, $C(q)$ is guaranteed to be an orthogonal matrix, i.e., $C(q)C'(q) = C'(q)C(q) = I_{3 \times 3}$.

Quaternions

Quaternion (from Rotation Matrix)

The inverse problem is to determine q as a function of C :

$$\begin{aligned}
 1. \quad T &= \text{trace}(C) = c_{1,1} + c_{2,2} + c_{3,3} = 4q_0^2 - 1 \\
 2. \quad q &= \begin{bmatrix} \sqrt{1+T}/2 \\ c_{2,3} - c_{3,2}/4q_1 \\ c_{3,1} - c_{1,3}/4q_1 \\ c_{1,2} - c_{2,1}/4q_1 \end{bmatrix} = \begin{bmatrix} c_{2,3} - c_{3,2}/4q_1 \\ \sqrt{1+2c_{1,1}-T}/2 \\ c_{1,2} + c_{2,1}/4q_1 \\ c_{1,3} + c_{3,1}/4q_1 \end{bmatrix} \\
 &= \begin{bmatrix} c_{3,1} - c_{1,3}/4q_2 \\ c_{1,2} + c_{2,1}/4q_2 \\ \sqrt{1+2c_{2,2}-T}/2 \\ c_{2,3} + c_{3,2}/4q_2 \end{bmatrix} = \begin{bmatrix} c_{1,2} - c_{2,1}/4q_3 \\ c_{1,3} + c_{3,1}/4q_3 \\ c_{2,3} + c_{3,2}/4q_3 \\ \sqrt{1+2c_{3,3}-T}/2 \end{bmatrix}
 \end{aligned}$$

3. Each of these four solutions will be singular when the pivotal element is zero (q_0 , q_1 , q_2 , and q_3 , respectively), but at least one will not be (since otherwise q could not have unit norm). For maximum numerical accuracy, the form with the largest pivotal element should be used. The maximum of ($|q_0|$, $|q_1|$, $|q_2|$, $|q_3|$) corresponds to the most positive of (T , $c_{1,1}$, $c_{2,2}$, $c_{3,3}$), respectively.

Quaternions

Derivative and Multiplication

The orientation estimation problem is to estimate quaternion q . The derivative of q is given by:

$$\frac{dq}{dt} = \frac{1}{2} q \otimes \tilde{\omega}$$

where $\tilde{\omega} = \begin{bmatrix} 0 \\ \omega \end{bmatrix}$, $\omega \in \mathbb{R}^3$ is the angular velocity, and \otimes represents quaternion multiplication:

$$\begin{aligned} q \otimes p &= (q_0 + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k})(p_0 + p_1 \mathbf{i} + p_2 \mathbf{j} + p_3 \mathbf{k}) \\ &= \begin{bmatrix} q_0 p_0 - q_1 p_1 - q_2 p_2 - q_3 p_3 \\ q_0 p_1 + q_1 p_0 + q_2 p_3 - q_3 p_2 \\ q_0 p_2 - q_1 p_3 + q_2 p_0 + q_3 p_1 \\ q_0 p_3 + q_1 p_2 - q_2 p_1 + q_3 p_0 \end{bmatrix} \end{aligned}$$

It should be noted that the quaternion multiplication is not commutative, i.e., $q \otimes p \neq p \otimes q$.

Sensor Models

Gyroscope Model

The gyroscope output, expressed in the body frame, is modeled as:

$$y_g = \omega + b_g + n_g$$

where:

- ω is the angular velocity
- b_g is the gyroscope bias (assumed nearly constant)
- n_g is the gyroscope noise

Sensor Models

Accelerometer Model (1/2)

The accelerometer output, expressed in the body frame, is modeled as:

$$y_a = C(q)\tilde{g} + b_a + n_a + a_b$$

where:

- $C(q)$ is the rotation matrix
- $\tilde{g} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}$ is the gravitational field, expressed in the navigation frame
- g is the intensity of the gravitational field
- b_a is the accelerometer bias (assumed nearly constant)
- n_a is the accelerometer noise
- a_b is the external acceleration

Two different approaches to estimate a_b will be shown:
a. a norm-based algorithm [A. M. Sabatini 2006]
b. an adaptive algorithm [Y. S. Suh 2010]

Sensor Models

Accelerometer Model (2/2)



The intensity of the gravitational field (both for simulated and real data) was chosen according to the WGS84 model (<http://walter.bislins.ch/bloge/index.asp?page=Earth+Gravity+Calculator>) at the proper location (43°52'52"N, 10°14'6"E, Viareggio, IT) where real data have been acquired:

- $g = 9.805185 \text{ m/s}^2$

Sensor Models

Magnetometer Model (1/2)

The magnetometer output, expressed in the body frame, is modeled as:

$$y_m = C(q)\tilde{m} + n_m$$

where:

- $C(q)$ is the rotation matrix
- $\tilde{m} = \begin{bmatrix} \cos \alpha \\ 0 \\ -\sin \alpha \end{bmatrix} \cdot m$ is the magnetic field, expressed in the navigation frame
- α is the dip angle or magnetic inclination
- m is the intensity of the magnetic field
- n_m is the magnetometer noise

Sensor Models

Magnetometer Model (2/2)

The values for the magnetic field were chosen according to the most recent (2019-2024) World Magnetic Model (WMM) (<https://www.ngdc.noaa.gov/geomag/calculators/magcalc.shtml#igrfwmm>), at the proper date (February 13th, 2020) and location (43°52'52"N, 10°14'6"E, Viareggio, IT) in which real data have been acquired:

- $\alpha = 60.10803^\circ$
- $m = 47.1179 \mu T$



Sensor Models

Measurement Noises (1/2)

Measurement noises are assumed to be zero-mean white Gaussian noises satisfying:

- $E\{n_g\} = 0_{3 \times 1}$
- $E\{n_a\} = 0_{3 \times 1}$
- $E\{n_m\} = 0_{3 \times 1}$
- $E\{n_g(t + \tau)n'_g(t)\} = R_g\delta(\tau)$
- $E\{n_a(t + \tau)n'_a(t)\} = R_a\delta(\tau)$
- $E\{n_m(t + \tau)n'_m(t)\} = R_m\delta(\tau)$
- $E\{n_g(t + \tau)n'_a(t)\} = 0_{3 \times 1}$
- $E\{n_g(t + \tau)n'_m(t)\} = 0_{3 \times 1}$
- $E\{n_a(t + \tau)n'_m(t)\} = 0_{3 \times 1}$

Sensor Models

Measurement Noises (2/2)

For simplicity, measurement noises will be assumed to be equal in all three spatial directions, i.e.:

- $R_g = \sigma_g^2 \cdot I_{3 \times 3}$ is the covariance measurement matrix of the gyroscope
- $R_a = \sigma_a^2 \cdot I_{3 \times 3}$ is the covariance measurement matrix of the accelerometer
- $R_m = \sigma_m^2 \cdot I_{3 \times 3}$ is the covariance measurement matrix of the magnetometer

where:

- σ_g is the standard deviation of the gyroscope global errors (noises, external perturbations, calibration defects, etc.)
- σ_a is the standard deviation of the accelerometer global errors (noises, external perturbations, calibration defects, etc.)
- σ_m is the standard deviation of the magnetometer global errors (noises, external perturbations, calibration defects, etc.)

Sensor Models

Sensors Using Simulated Data

Gyroscope bias b_g and accelerometer bias b_a are assumed to be nearly constant, whereas magnetometer bias b_m is assumed to be null.

The characteristics of the simulated sensors are summarized in the following table:

	Bias	Standard deviation
Gyroscope [rad/s]	$b_g = [-0.019 \quad 0.013 \quad -0.006]'$	$\sigma_g = 0.006$
Accelerometer [m/s²]	$b_a = [0.07 \quad 0.033 \quad -0.044]'$	$\sigma_a = 0.048$
Magnetometer [μT]	$b_m = [0 \quad 0 \quad 0]'$	$\sigma_m = 2$

Sensor Models

Sensors Using Real Data

Biases of real sensors are unknown, and they will be estimated by the implemented Kalman Filter. Hence, they are not artificially added.

The gyro and accelerometer biases taken as reference (and reported in the following table) were obtained averaging angular velocity and accelerometer measurements, respectively, during the first quiet period (about 5 minutes long), when gyroscope and accelerometer output, respectively, were almost static.

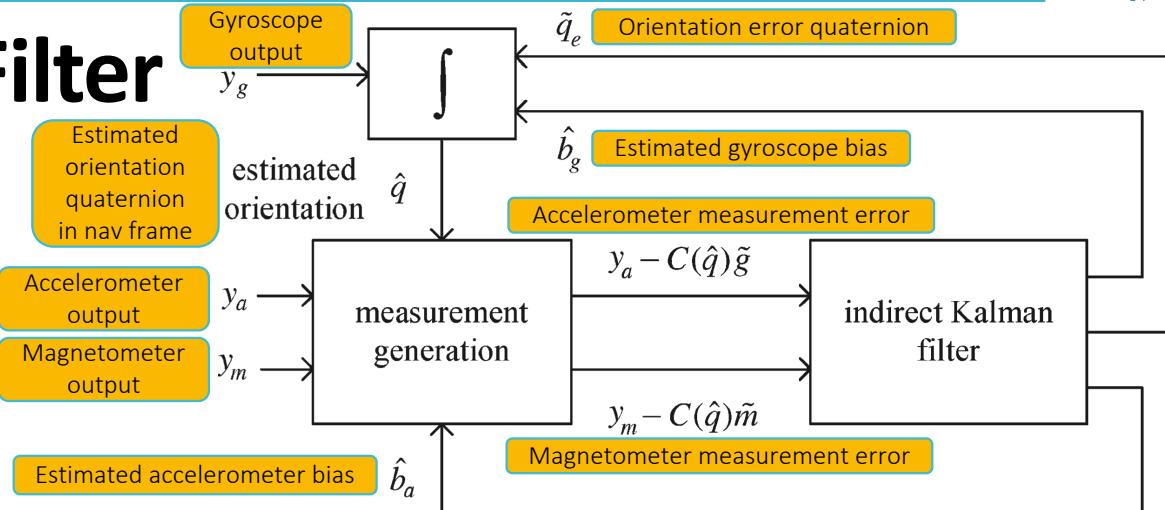
Regarding the measurement noise, the vendor did not provide reliable datasheets. After much trial and error, reasonable values were found (and reported in the following table).

It should be noted that, actually, these standard deviations include gyroscope, accelerometer, and magnetometer, respectively, global errors, which are due to measurement noises, external perturbations, calibration defects, etc..

	Bias	Standard deviation
Gyroscope [rad/s]	$b_g = [-0.0640 \quad -0.0240 \quad 0.0205]'$	$\sigma_g = 0.001$
Accelerometer [m/s²]	$b_a = [0.5875 \quad -0.0000036 \quad 0.0312]'$	$\sigma_a = 0.039$
Magnetometer [µT]	$b_m = [0 \quad 0 \quad 0]'$	$\sigma_m = 2$

Indirect Kalman Filter

Introduction



The objective of the proposed filter is to estimate the attitude quaternion q , i.e., \hat{q} , from the sensor outputs y_g , y_a , and y_m .

It consists of two phases which alternate:

1. The *propagation stage*, where the filter produces a prediction of the attitude \hat{q} based on the previous estimate of the state $x = [q_e \quad b_g \quad b_g]'$, i.e., $\hat{x} = [\tilde{q}_e \quad \hat{b}_g \quad \hat{b}_g]'$, and the most recent gyroscope output y_g .
2. The *update stage*, which corrects the values predicted in the propagation stage. It consists of a two-step measurement update:
 - a. An *accelerometer measurement update*, where the most recent accelerometer output y_a is used.
 - b. A *magnetometer measurement update*, where the most recent magnetometer output y_m is used.

Indirect Kalman Filter

Orientation Error Quaternion

An attitude quaternion estimate \hat{q} is computed from the integration of the following equation:

$$\frac{d\hat{q}}{dt} = \frac{1}{2} \hat{q} \otimes y_g$$

Since $y_g \neq \omega$ (due to gyro noise n_g and gyro bias b_g), $\hat{q} \neq q$, i.e., \hat{q} contains an orientation error. We introduce \tilde{q}_e , the orientation error quaternion, to denote a small error in \hat{q} so that:

$$q = \hat{q} \otimes \tilde{q}_e$$

It should be noted that \tilde{q}_e does not depend on ω (angular velocity) but depends on gyro noise n_g and gyro bias b_g , which can be assumed to be small. Thus, even if the rotations are large, we can assume that \tilde{q}_e is small. Assuming \tilde{q}_e is small, we can approximate it as follows:

$$\tilde{q}_e \approx \begin{bmatrix} 1 \\ q_e \end{bmatrix}$$

In an Indirect Kalman Filter, \tilde{q}_e is estimated, and q can be estimated through $q = \hat{q} \otimes \tilde{q}_e$ instead of directly estimating it.

Indirect Kalman Filter

State Equation (1/2)

Defining the state x by:

$$x = \begin{bmatrix} q_e \\ b_g \\ b_a \end{bmatrix} \in \mathbb{R}^{9 \times 1}$$

we have the following process equation for the Kalman Filter:

$$\dot{x}(t) = Ax(t) + w = Ax(t) + \begin{bmatrix} -0.5n_g \\ n_{b_g} \\ n_{b_a} \end{bmatrix}$$

where:

$$\bullet A = \begin{bmatrix} -[y_g \times] & -0.5I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}$$

For a vector $p \in \mathbb{R}^3$, $p = [p_1 \ p_2 \ p_3]'$, $[p \times]$ is the skew-symmetric matrix operator which is defined as:

$$[p \times] = \begin{bmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{bmatrix}$$

Indirect Kalman Filter

State Equation (2/2)

- n_{b_g} and n_{b_a} represent the process noises; they are added so that the bias estimation is not stopped soon. In fact, if $n_{b_g} = 0$ and $n_{b_a} = 0$, the error covariance of the filter corresponding to the state components b_g and b_a will be very small after some time, and thus, no bias update is done afterward.

It is assumed that:

$$E\{w(t + \tau)w'(t)\} = Q\delta(\tau)$$

where Q is the continuous-time process noise covariance matrix defined by:

$$Q = \begin{bmatrix} 0.25R_g & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & Q_{b_g} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & Q_{b_a} \end{bmatrix}$$

$R_g = \sigma_g^2 \cdot I_{3 \times 3}$ is the covariance measurement matrix of the gyroscope

Q_{b_g} is the covariance noise matrix of the gyroscope

Q_{b_a} is the covariance noise matrix of the accelerometer

Indirect Kalman Filter

Propagation Stage (“Prediction”) (1/3)

- Obtain the predicted quaternion \hat{q}_{k+1} integrating:

$$\frac{d\hat{q}}{dt} = \frac{1}{2} \hat{q} \otimes y_g$$

The quaternion integration is approximated by a discrete equation, where the third-order local linearization algorithm is used:

$$\hat{q}_{k+1} = \left(I + \frac{3}{4} \Omega_k T_s - \frac{1}{4} \Omega_{k-1} T_s - \frac{1}{6} \|y_g\|_2^2 T_s^2 - \frac{1}{24} \Omega_k \Omega_{k-1} T_s^2 - \frac{1}{48} \|y_g\|_2^2 T_s^3 \right) \hat{q}_k$$

where:

- T_s is the sampling period
 - $T_s = 0.005$ s with simulated data
 - $T_s = 0.02$ s with real data

$$\bullet \quad \Omega_k = \begin{bmatrix} 0 & -y_{g,x}(kT_s) & -y_{g,y}(kT_s) & -y_{g,z}(kT_s) \\ y_{g,x}(kT_s) & 0 & y_{g,z}(kT_s) & -y_{g,y}(kT_s) \\ y_{g,y}(kT_s) & -y_{g,z}(kT_s) & 0 & y_{g,x}(kT_s) \\ y_{g,z}(kT_s) & y_{g,y}(kT_s) & -y_{g,x}(kT_s) & 0 \end{bmatrix}$$

Indirect Kalman Filter

Propagation Stage (“Prediction”) (2/3)

2. Compute the state transition matrix:

$$\phi_k = e^{AT_s}$$

and the discrete-time process noise covariance matrix:

$$Q_{d,k} = E\{w_k w_k'\} = \int_{kT_s}^{(k+1)T_s} e^{At} Q (e^{At})' dt$$

These matrices can be approximated as follows:

- a. $\phi_k \approx I + A(kT_s)T_s + \frac{1}{2}A(kT_s)^2 T_s^2$
- b. $Q_{d,k} \approx QT_s + \frac{1}{2}A(kT_s)Q + \frac{1}{2}QA(kT_s)'$

This is useful as ϕ_k and $Q_{d,k}$ should be computed at every step, since A is time-varying. In fact, in real-time applications, it could be computationally demanding to compute exact values of ϕ_k and $Q_{d,k}$.

Indirect Kalman Filter

Propagation Stage (“Prediction”) (3/3)

3. The discrete model is:

$$x_{k+1} = \phi_k x_k + w_k$$

Based on it, a standard project ahead algorithm is used.

- a. Therefore, project the state ahead:

$$\hat{x}_{k+1}^- = \phi_k \hat{x}_k$$

- b. and project the state covariance matrix ahead:

$$P_{k+1}^- = \phi_k P_k \phi_k' + Q_{d,k}$$

Indirect Kalman Filter

Accelerometer Measurement Update Stage (“1st Correction”) (1/3)

1. Compute accelerometer measurement matrix:

$$H_{a,k} = [2[C(\hat{q}_k)\tilde{g}\times] \quad 0_{3\times 3} \quad I_{3\times 3}]$$

2. Compute accelerometer measurement error according to:

$$z_{a,k} = y_{a,k} - C(\hat{q}_k)\tilde{g}$$

3. Estimate external acceleration covariance matrix $\hat{Q}_{a_b,k}$ with:
 - a norm-based algorithm [A. M. Sabatini 2006]
 - an adaptive algorithm [Y. S. Suh 2010]

We'll talk about that later on.

Indirect Kalman Filter

Accelerometer Measurement Update Stage (“1st Correction”) (2/3)

4. Compute accelerometer residual covariance matrix:

$$S_{a,k} = H_{a,k} P_k^- H_{a,k}' + R_a + \hat{Q}_{a_b,k}$$

5. Compute accelerometer Kalman gain:

$$K_{a,k} = P_k^- H_{a,k}' S_{a,k}^{-1}$$

6. Compute residual in accelerometer measurement update:

$$r_{a,k} = z_{a,k} - H_{a,k} \hat{x}_k^-$$

7. Update the state vector:

$$\hat{x}_{a,k} = \hat{x}_k^- + K_{a,k} r_{a,k}$$

All nine elements of the state are updated in this step.

Indirect Kalman Filter

Accelerometer Measurement Update Stage (“1st Correction”) (3/3)

8. Update the state covariance matrix:

$$P_{a,k} = (I_{3 \times 3} - K_{a,k} H_{a,k}) P_k^- (I_{3 \times 3} - K_{a,k} H_{a,k})' + K_{a,k} (R_a + \hat{Q}_{a_b,k}) K_{a,k}'$$

9. Update the estimated attitude quaternion \hat{q} as follows:

- a. $q_e = \hat{x}_{a,k}(1:3)$, i.e., extract the imaginary part of the attitude error quaternion from the state vector
- b. $\tilde{q}_e = \begin{bmatrix} 1 \\ q_e \end{bmatrix}$, i.e., create the attitude error quaternion
- c. $\hat{q}_k = \hat{q}_k \otimes \tilde{q}_e$, i.e., compute the new attitude quaternion
- d. $\hat{q}_k = \frac{\hat{q}_k}{\|\hat{q}_k\|}$, i.e., normalize the quaternion
- e. $\hat{x}_{a,k}(1:3) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$, i.e., set the first 3 elements of the state array to 0



Indirect Kalman Filter

Estimation of Ext. Acceleration Covariance Matrix: [Sabatini]

The first approach implemented for estimating $Q_{a_b,k}$ — the time-varying covariance matrix of the external acceleration a_b — is based on a norm-based algorithm proposed by A. M. Sabatini:

$$\hat{Q}_{a_b,k} = \begin{cases} 0_{3 \times 3}, & \left\| \|y_{a,k}\|_2 - g \right\| < \varepsilon_a \\ sI_{3 \times 3}, & \text{otherwise} \end{cases}$$

No external acceleration detected

External acceleration detected

where $\varepsilon_a, s \in \mathbb{R}_{\geq 0}$ are two constants. A possible choice is $\varepsilon_a = 0.25 \text{m/s}^2$, $s = 10$.

In case the latter condition is met — i.e., when an external acceleration is detected — less weights are given to all three outputs of the accelerometer (hence, without differentiating among the three axes).



Indirect Kalman Filter

Estimation of Ext. Acceleration Covariance Matrix: [Suh] (1/2)

The second approach used is based on an adaptive algorithm proposed by Y. S. Suh to estimate the external acceleration from the residual.

1. Residual in accelerometer measurement update was defined by:

$$r_{a,k} = z_{a,k} - H_{a,k} \hat{x}_k^-$$

2. Its covariance can be approximated as:

$$E\{r_{a,k} r_{a,k}'\} \approx U_k = \frac{1}{M_1} \sum_{i=0}^{M_1-1} r_{a,k-i} r_{a,k-i}'$$

where $M_1 \in \mathbb{N}_{>0}$ is a constant.

3. Since U_k is symmetric, there are orthonormal eigenvectors $u_{i,k} \in \mathbb{R}^{3 \times 1}$ ($i = 1, 2, 3$) and corresponding eigenvalues $\lambda_{i,k} \in \mathbb{R}$ ($i = 1, 2, 3$) so that U_k can be expressed as follows:

$$U_k = \sum_{i=1}^3 \lambda_{i,k} u_{i,k} u_{i,k}'$$

4. Let $\mu_{i,k}$ ($i = 1, 2, 3$) be defined by:

$$\mu_{i,k} = u_{i,k}' (H_{a,k} P_k^- P_{a,k}' + R_a) u_{i,k}$$



Indirect Kalman Filter

Estimation of Ext. Acceleration Covariance Matrix: [Suh] (2/2)

5. Finally, an adaptive estimation algorithm of $\hat{Q}_{a_b,k}$ is given in pseudocode:

- if $\max_i(\lambda_{i,j} - \mu_{i,j}) < \gamma$ ($j = k, k-1, \dots, k-M_2$), where $\gamma \in \mathbb{R}_{\geq 0}$, $M_2 \in \mathbb{N}$ are two constants, use *Mode 1* (no external acceleration mode):

$$\hat{Q}_{a_b,k} = 0$$

No external
acceleration detected

- else, use *Mode 2* (external acceleration mode):

$$\hat{Q}_{a_b,k} = \sum_{i=1}^3 \max_k(\lambda_{i,k} - \mu_{i,k}, 0) u_{i,k} u'_{i,k}$$

External acceleration
detected

- The condition $\max_i(\lambda_{i,j} - \mu_{i,j}) < \gamma$ is introduced so that *Mode 2* is only used when U_k is significantly greater than $H_{a,k}P_k^-P'_{a,k} + R_a$ in any direction. This is to prevent $\hat{Q}_{a_b,k}$ from being affected by normal fluctuation of accelerometer noises.
- M_2 is introduced so that transition from *Mode 2* to *Mode 1* occurs only if $\max_i(\lambda_{i,j} - \mu_{i,j}) < \gamma$ is satisfied for $M_2 + 1$ consecutive times. This is to prevent falsely entering *Mode 1* when there is external acceleration.
- There is no delay in transition from *Mode 1* to *Mode 2*, so that external acceleration is quickly estimated.
- Typical values for M_1 , M_2 , and γ , are $M_1 = 3$, $M_2 = 2$, and $\gamma = 0.1$.

Indirect Kalman Filter

Magnetometer Measurement Update Stage (“2nd Correction”) (1/3)

1. Compute magnetometer measurement matrix:

$$H_{m,k} = [2[C(\hat{q}_k)\tilde{m}\times] \quad 0_{3 \times 3} \quad 0_{3 \times 3}]$$

2. Compute magnetometer measurement error according to:

$$z_{m,k} = y_{m,k} - C(\hat{q}_k)\tilde{m}$$

3. Update the covariance matrix:

$$P_{m,k}^- = \begin{bmatrix} P_{a,k}(1:3,1:3) & 0_{3 \times 6} \\ 0_{6 \times 3} & 0_{6 \times 3} \end{bmatrix}$$

4. Compute magnetometer residual covariance matrix:

$$S_{m,k} = H_{m,k} P_{m,k}^- H_{m,k}' + R_m$$

Indirect Kalman Filter

Magnetometer Measurement Update Stage (“2nd Correction”) (2/3)

5. To limit the effect of this correction only to the yaw component, the computation of the magnetometer Kalman gain is modified as following:

$$K_{m,k} = \begin{bmatrix} r_3 r'_3 & 0_{3 \times 6} \\ 0_{6 \times 3} & 0_{6 \times 6} \end{bmatrix} P_{m,k}^- H'_{m,k} S_{m,k}^{-1}$$

where:

$$r_3 = C(\hat{q}_k) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

6. Compute residual in the magnetometer measurement update:

$$r_{m,k} = z_{m,k} - H_{m,k} \hat{x}_{a,k}$$

Indirect Kalman Filter

Magnetometer Measurement Update Stage (“2nd Correction”) (3/3)

7. Update the state vector:

$$\hat{x}_k = \hat{x}_{a,k} + K_{m,k} r_{m,k}$$

8. Update the state covariance matrix:

$$P_k = (I_{3 \times 3} - K_{m,k} H_{m,k}) P_{a,k} (I_{3 \times 3} - K_{m,k} H_{m,k})' + K_{m,k} R_m K'_{m,k}$$

From the structure of $P_{m,k}$ and $K_{m,k}$ it can be noticed that only the q_e part in $\hat{x}_{a,k}$ is updated, i.e., b_g and b_a are not updated in this stage.

Moreover, due to the particular construction of r_3 , it can be proved that only yaw angles are modified in this stage, which can be helpful as magnetic disturbances sometimes could be very large.

Performance Metrics

Quaternion Integrations

The performance metric chosen is given by the Root-Mean-Square Error (RMSE) of Euler angles, which are computed from rotation matrices obtained from quaternions yielded by integration of angular velocities.



In more detail:

1. Compute orientation quaternions by integrating angular velocities:
 - a. True angular velocities ω yields true quaternions q :
$$q_{k+1} = \left(I + \frac{3}{4}\Omega_k T_s - \frac{1}{4}\Omega_{k-1} T_s - \frac{1}{6}\|\omega\|_2^2 T_s^2 - \frac{1}{24}\Omega_k \Omega_{k-1} T_s^2 - \frac{1}{48}\|\omega\|_2^2 T_s^3 \right) q_k$$
 - b. Gyroscope measurements y_g yields estimated-from-measurement quaternions \bar{q} :
$$\bar{q}_{k+1} = \left(I + \frac{3}{4}\Omega_k T_s - \frac{1}{4}\Omega_{k-1} T_s - \frac{1}{6}\|y_g\|_2^2 T_s^2 - \frac{1}{24}\Omega_k \Omega_{k-1} T_s^2 - \frac{1}{48}\|y_g\|_2^2 T_s^3 \right) \bar{q}_k$$
 - c. Gyroscope measurements y_g — and using quaternion corrections — yields estimated-from-Kalman-filter quaternions \hat{q} :
$$\hat{q}_{k+1} = \left(I + \frac{3}{4}\Omega_k T_s - \frac{1}{4}\Omega_{k-1} T_s - \frac{1}{6}\|y_g\|_2^2 T_s^2 - \frac{1}{24}\Omega_k \Omega_{k-1} T_s^2 - \frac{1}{48}\|y_g\|_2^2 T_s^3 \right) \hat{q}_k$$

Finally, \hat{q}_{k+1} is corrected in the accelerometer measurement update stage.

Performance Metrics

Initial ($k=0$) Estimation of Euler Angles & Quaternions

- Unfortunately, initial estimated orientation quaternions $\bar{q}(0)$ and $\hat{q}(0)$ are neither known nor can they be integrated from previous quaternions and angular velocities. The procedure to estimate them is as follows:
 - Given accelerometer and magnetometer measurements y_a and y_m , it is possible to estimate initial Euler angles:
 - $\bar{\phi}(0) = \hat{\phi}(0) = \text{atan2}(y_{a,y}(0), y_{a,z}(0))$ is the estimated roll
 - $\bar{\theta}(0) = \hat{\theta}(0) = \text{atan2}\left(y_{a,x}(0), \sqrt{y_{a,y}^2 + y_{a,z}^2}\right)$ is the estimated pitch
 - $\bar{\psi}(0) = \hat{\psi}(0) = \text{atan2}\left(-\hat{C}_n^b(0) \cdot y_{m,y}(0), \hat{C}_n^b(0) \cdot y_{m,x}(0)\right)$ is the estimated yaw, where: $\hat{C}_n^b(0) = C_Y(\hat{\theta}(0)) \cdot C_X(\hat{\phi}(0))$
 - Then, initial estimated orientation quaternions are computed as follows:
 - $\bar{q}(0) = \hat{q}(0) = \text{RotMatToQuat}(C_Z(\hat{\psi}(0)) \cdot C_Y(\hat{\theta}(0)) \cdot C_X(\hat{\phi}(0)))$ (cf. [Quaternion \(from Rotation Matrix\)](#))

$\bar{q}(k), \hat{q}(k), k = 1, \dots, N - 1$ are computed as shown in [Quaternion Integrations](#).

Performance Metrics

Rotation Matrices

2. Compute direction cosine matrices $C(q)$ from previously obtained quaternions:

$$C(q) = \begin{bmatrix} 2q_0^2 + 2q_1^2 - 1 & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 - 2q_0q_2 \\ 2q_1q_2 - 2q_0q_3 & 2q_0^2 + 2q_2^2 - 1 & 2q_2q_3 + 2q_0q_1 \\ 2q_1q_3 + 2q_0q_2 & 2q_2q_3 - 2q_0q_1 & 2q_0^2 + 2q_3^2 - 1 \end{bmatrix}$$

In particular:

- Using q yields $C(q) = C_n^b$
- Using \bar{q} yields $C(\bar{q}) = \bar{C}_n^b$
- Using \hat{q} yields $C(\hat{q}) = \hat{C}_n^b$

Performance Metrics

Euler Angles

3. Convert direction cosine matrix $C(q)$ to Euler angles:

- $\phi = \text{atan2}(c_{2,3}, c_{3,3})$ is the roll
- $\theta = \text{atan2}\left(-c_{1,3}, \sqrt{c_{1,1}^2 + c_{1,2}^2}\right)$ is the pitch
- $\psi = \text{atan2}(\sin \phi \cdot c_{3,1} - \cos \phi \cdot c_{2,1}, \cos \phi \cdot c_{2,2} - \sin \phi \cdot c_{3,2})$ is the yaw

In particular:

- Using $C(q) = C_n^b$ yields (for the simulated data set) true Euler angles ϕ, θ, ψ
- Using $C(\bar{q}) = \bar{C}_n^b$ yields estimated-from-measurement Euler angles $\bar{\phi}, \bar{\theta}, \bar{\psi}$
- Using $C(\hat{q}) = \hat{C}_n^b$ yields estimated-from-Kalman-Filter Euler angles $\hat{\phi}, \hat{\theta}, \hat{\psi}$

Regarding the real data set analyzed, the roll, pitch, and yaw angles estimated by another Kalman filter were considered as the reference for the purpose of error estimation. Indeed, it is worthwhile to mention that Euler angles estimated by the implemented Kalman Filter may even be more accurate.

Performance Metrics

RMSEs

4. Compute Root-Mean-Square Errors among all N samples:

a. Between true and estimated-from-Kalman-Filter Euler angles:

- $RMSE_{\hat{\phi}} = \sqrt{\frac{\sum_{k=0}^{N-1}(\hat{\phi}_k - \phi_k)^2}{N-1}}$, where $\hat{\phi}_k - \phi_k$ is wrapped in $[-180^\circ, 180^\circ]$
- $RMSE_{\hat{\theta}} = \sqrt{\frac{\sum_{k=0}^{N-1}(\hat{\theta}_k - \theta_k)^2}{N-1}}$, where $\hat{\theta}_k - \theta_k$ is wrapped in $[-90^\circ, 90^\circ]$
- $RMSE_{\hat{\psi}} = \sqrt{\frac{\sum_{k=0}^{N-1}(\hat{\psi}_k - \psi_k)^2}{N-1}}$, where $\hat{\psi}_k - \psi_k$ is wrapped in $[-180^\circ, 180^\circ]$

b. Between true and estimated-from-measurement Euler angles:

- $RMSE_{\bar{\phi}} = \sqrt{\frac{\sum_{k=0}^{N-1}(\bar{\phi}_k - \phi_k)^2}{N-1}}$, where $\bar{\phi}_k - \phi_k$ is wrapped in $[-180^\circ, 180^\circ]$
- $RMSE_{\bar{\theta}} = \sqrt{\frac{\sum_{k=0}^{N-1}(\bar{\theta}_k - \theta_k)^2}{N-1}}$, where $\bar{\theta}_k - \theta_k$ is wrapped in $[-90^\circ, 90^\circ]$
- $RMSE_{\bar{\psi}} = \sqrt{\frac{\sum_{k=0}^{N-1}(\bar{\psi}_k - \psi_k)^2}{N-1}}$, where $\bar{\psi}_k - \psi_k$ is wrapped in $[-180^\circ, 180^\circ]$

Simulated Data

Time Params & Sensor Characteristics

- Simulation duration: $t_f = 200$ s
- Sampling frequency: $f_s = 200$ Hz → Sampling time: $T_s = 1/f_s = 0.005$ s
- Number of samples: $N = t_f \cdot f_s = 40000$
- Gravitational field parameter: $g = 9.805185$ m/s²
- Magnetic field parameters: $\alpha = 60.10803^\circ$, $m = 47.1179$ µT
- Values of biases and standard deviations used for simulating sensor behavior:

Sampling frequency and sampling time are assumed to be constant, and the same for all the sensors.

Viareggio,
February 13th, 2020

	Bias	Standard deviation
Gyroscope [rad/s]	$b_g = [-0.019 \quad 0.013 \quad -0.006]'$	$\sigma_g = 0.006$
Accelerometer [m/s²]	$b_a = [0.07 \quad 0.033 \quad -0.044]'$	$\sigma_a = 0.048$
Magnetometer [µT]	$b_m = [0 \quad 0 \quad 0]'$	$\sigma_m = 2$

Simulated Data

Initial Euler Angles, State Vector and State Cov. Matrix

- Initial true Euler angles:

$$\phi(0) = -20^\circ$$

$$\theta(0) = 15^\circ$$

$$\psi(0) = 30^\circ$$

- Initial state vector:

$$x_0 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]'$$

- Initial state covariance matrix:

$$P_0 = \begin{bmatrix} 0.04 & 0 & 0 & & & & & & \\ 0 & 0.04 & 0 & & & & & & \\ 0 & 0 & 0.04 & & & & & & \\ & & & 0_{3 \times 3} & & & & & \\ & & & & 10^{-6} & 0 & 0 & & \\ & & & & 0 & 10^{-6} & 0 & & \\ & & & & 0 & 0 & 10^{-6} & & \\ & & & & & & & 0.04 & 0 & 0 \\ & & 0_{3 \times 3} & & 0_{3 \times 3} & & & 0 & 0.04 & 0 \\ & & & & & & & 0 & 0 & 0.04 \end{bmatrix}$$

Simulated Data

Norm-Based and Adaptive Alg. Params & Process Noise Cov. Matrix

- Norm-based algorithm parameters:

$$\varepsilon_a = 0.25 \text{ m/s}^2$$

$$s = 10$$

- Adaptive estimation of external acceleration algorithm parameters:

$$M_1 = 3$$

$$M_2 = 2$$

$$\gamma = 0.1$$

- Continuous-time process noise covariance matrix:

Q_{bg} is the covariance noise matrix of the gyroscope

$R_g = \sigma_g^2 \cdot I_{3 \times 3}$ is the covariance measurement matrix of the gyroscope

$$Q_{bg} = \begin{bmatrix} 10^{-6} & 0 & 0 \\ 0 & 10^{-6} & 0 \\ 0 & 0 & 10^{-6} \end{bmatrix}, \quad Q_{ba} = \begin{bmatrix} 10^{-6} & 0 & 0 \\ 0 & 10^{-6} & 0 \\ 0 & 0 & 10^{-6} \end{bmatrix}$$

Q_{ba} is the covariance noise matrix of the accelerometer

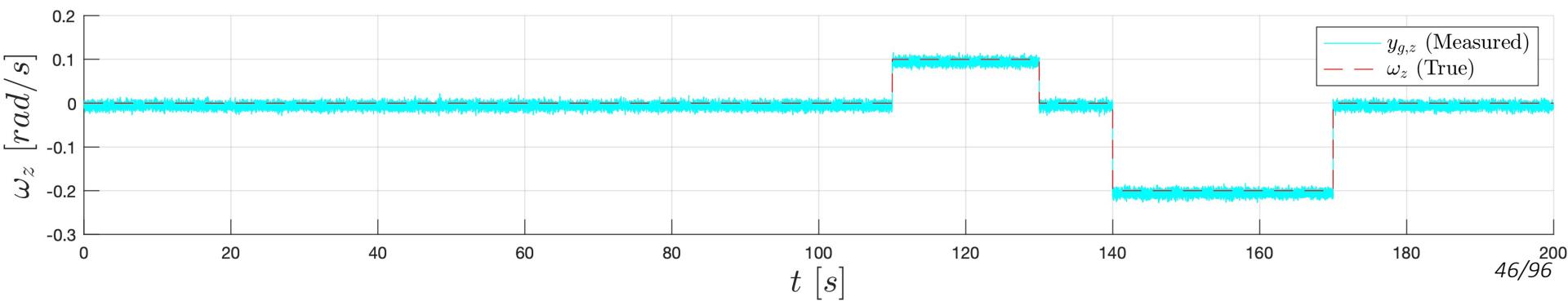
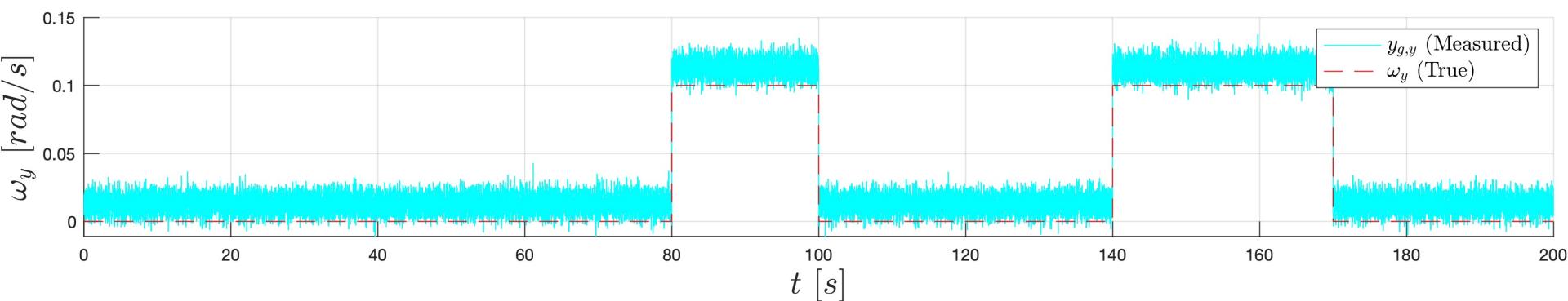
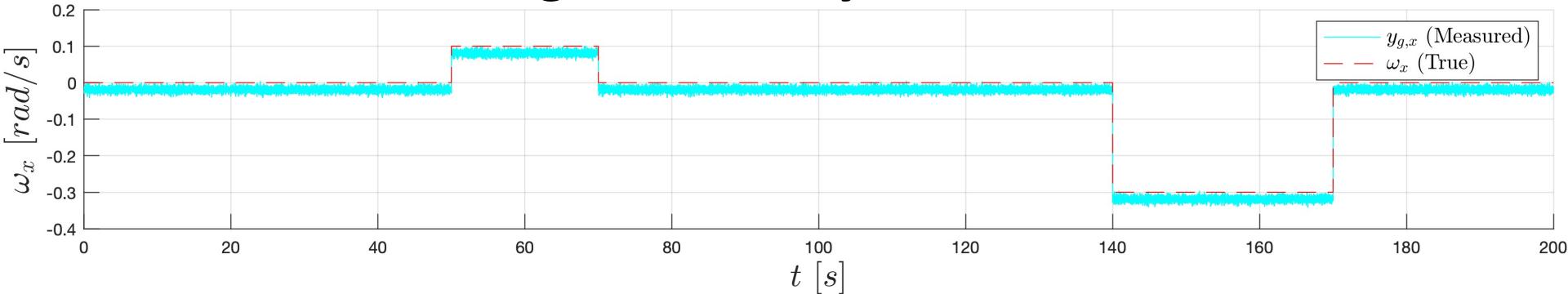
Recall that: $Q = \begin{bmatrix} 0.25R_g & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & Q_{bg} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & Q_{ba} \end{bmatrix}$

Simulated Data

Data Generation: Angular Velocity

- $y_g = \omega + b_g + n_g$
- ω

- Measured angular velocity y_g is affected by noise n_g and bias b_g



Simulated Data

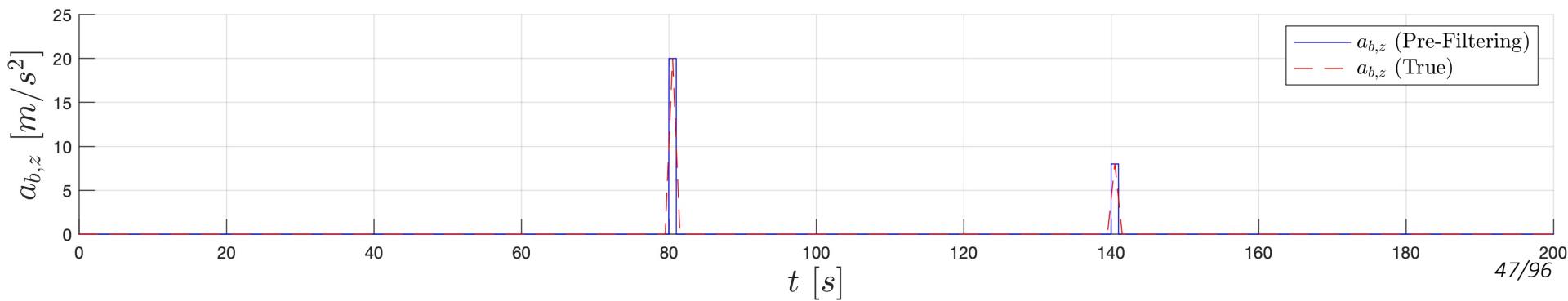
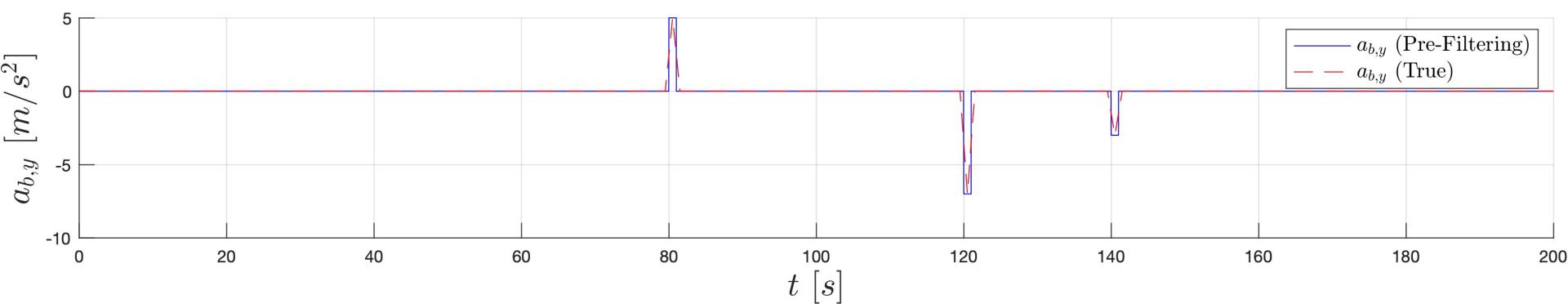
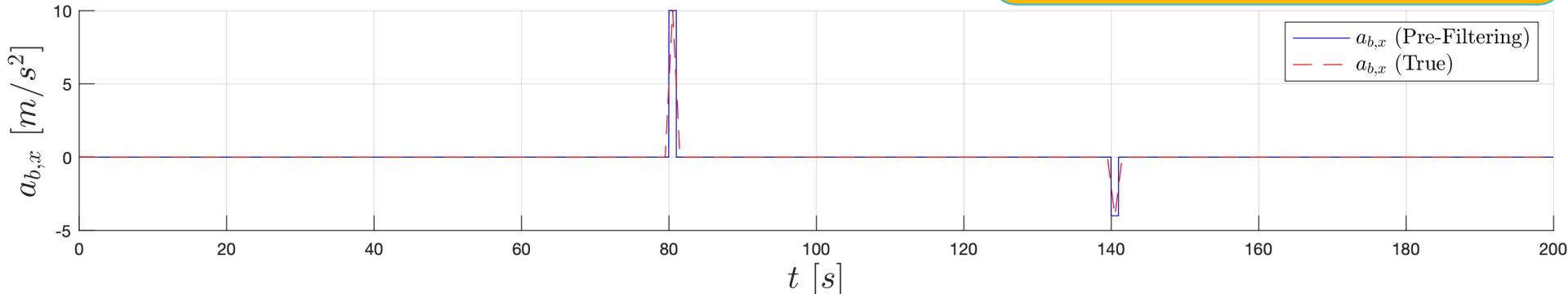
Data Generation: External Acceleration

• a_b

3 spikes of external acceleration:

- $[10 \ 5 \ 20]'$ m/s² from 80 to 81 s
- $[0 \ -7 \ 0]'$ m/s² from 120 to 121 s
- $[-4 \ -3 \ 8]'$ m/s² from 140 to 141 s

These spikes are then filtered by a moving average filter (with window size of $f_s/\text{Hz} = 200$ samples) for smoothing their shapes

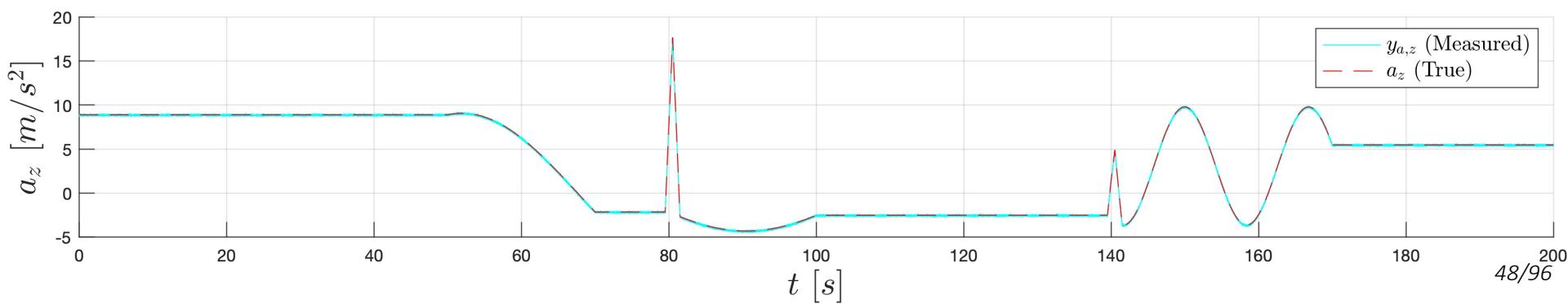
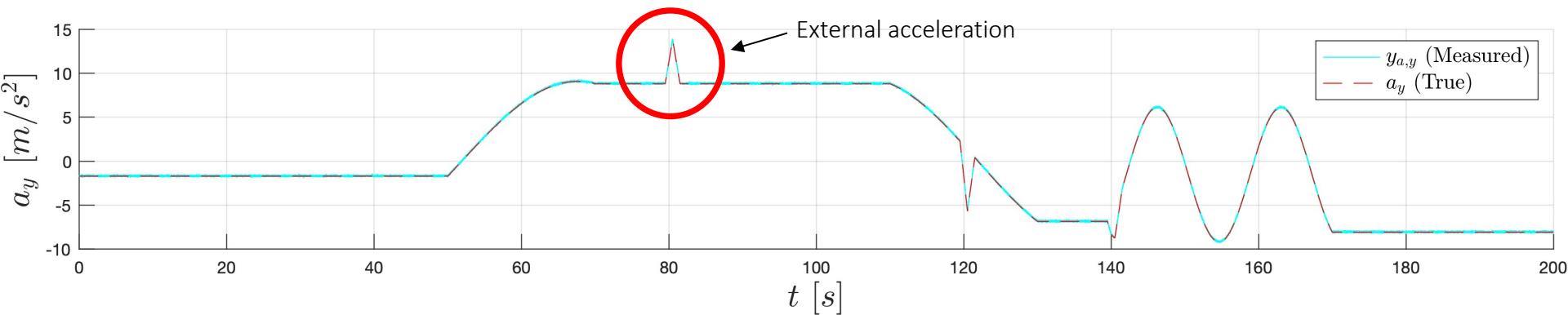
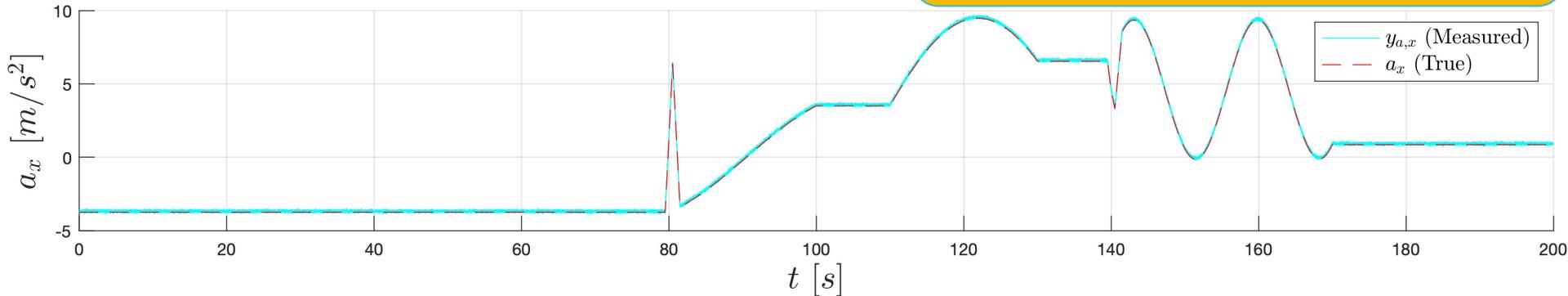


Simulated Data

Data Generation: Acceleration

- $y_a = C(q)\tilde{g} + b_a + n_a + a_b$
- $a = C(q)\tilde{g} + a_b$

- Measured acceleration y_a is affected by noise n_a and bias b_a
- Theoretical acceleration a is sum of internal acceleration $C(q)\tilde{g}$ and external acceleration a_b

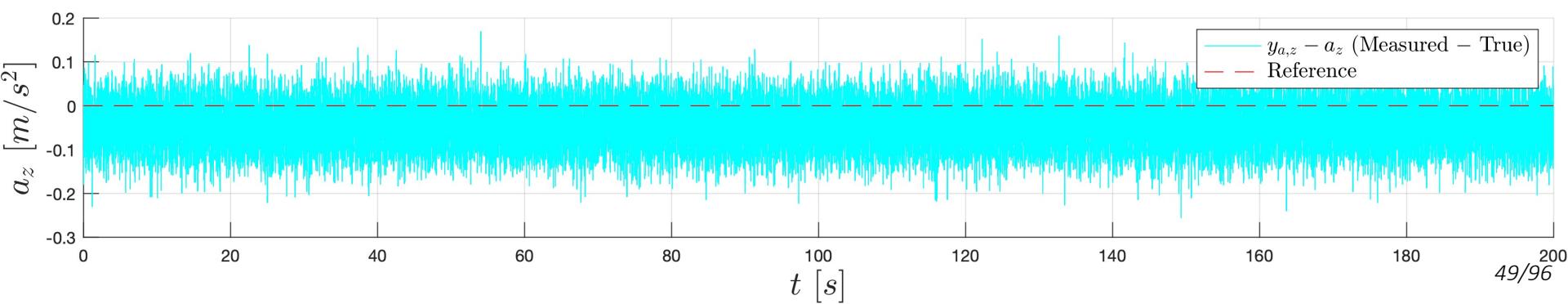
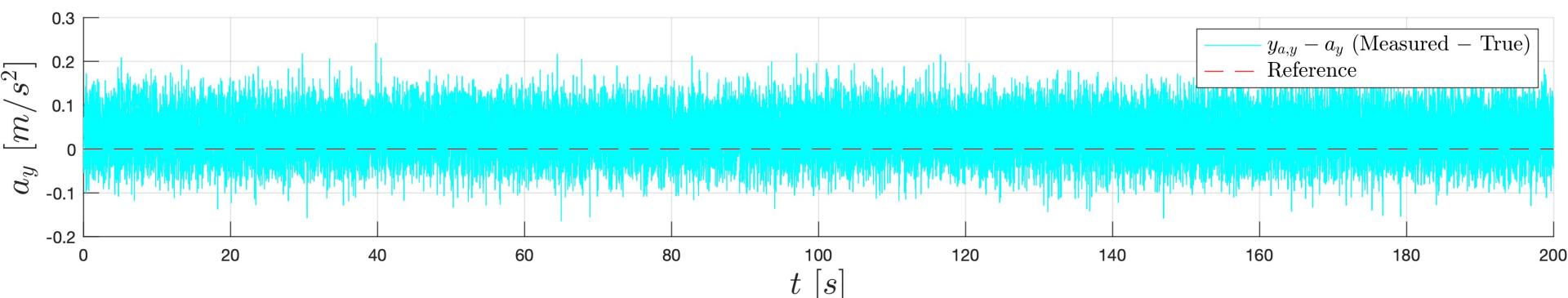
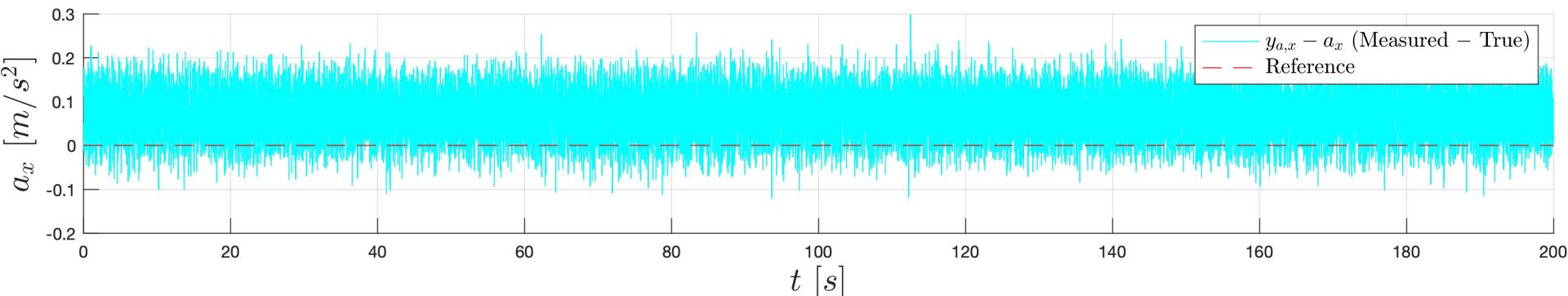


Simulated Data

Data Generation: Acceleration Error

- $y_a = C(q)\tilde{g} + b_a + n_a + a_b$
- $a = C(q)\tilde{g} + a_b$

- In this figure acceleration noise and bias are even more evident

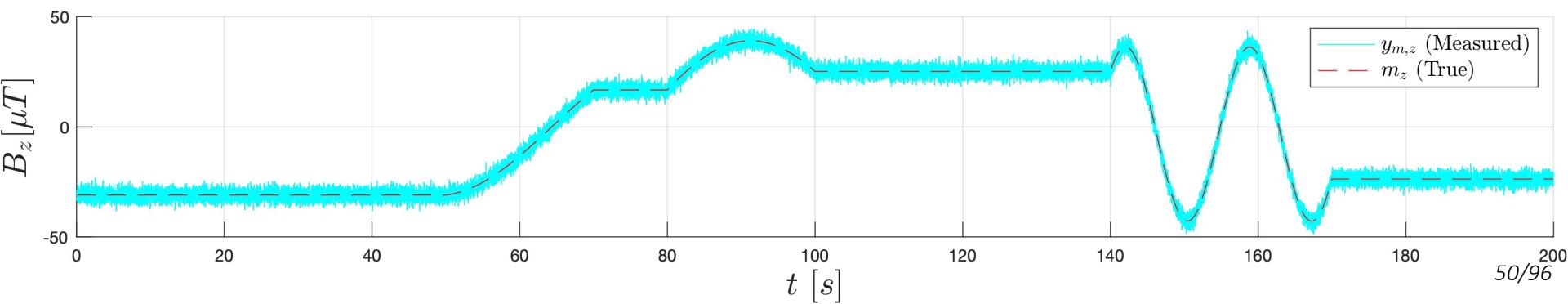
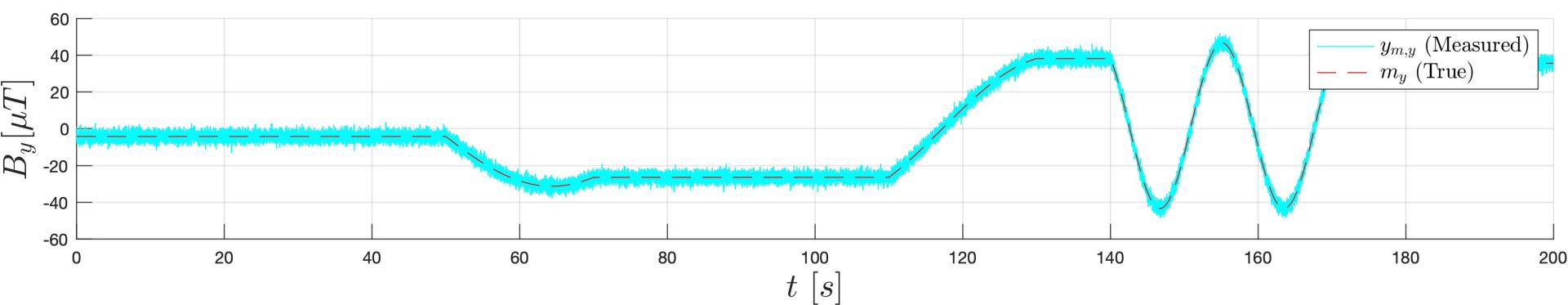
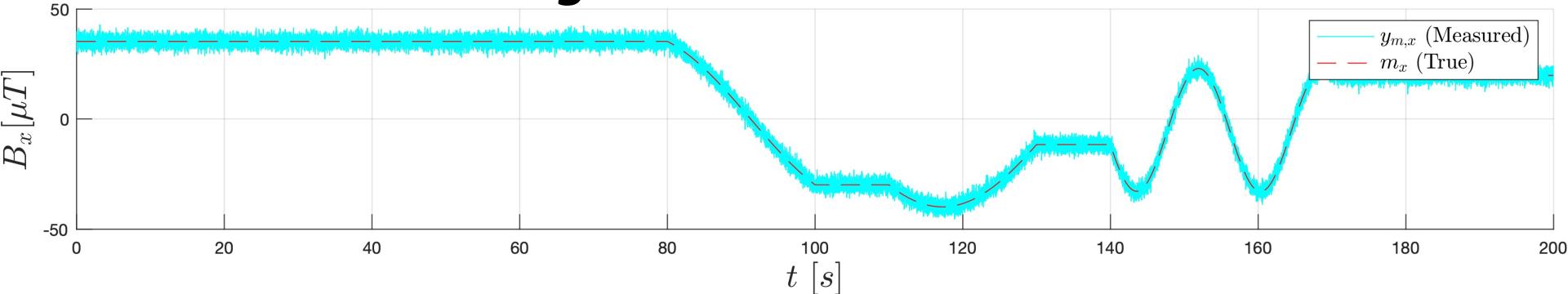


Simulated Data

Data Generation: Magnetic Field

- $y_m = C(q)\tilde{m} + n_m$
- $m = C(q)\tilde{m}$

- Measured magnetic field y_m is affected by noise n_m
- Magnetic bias b_m was supposed to be null: future works should take it into account

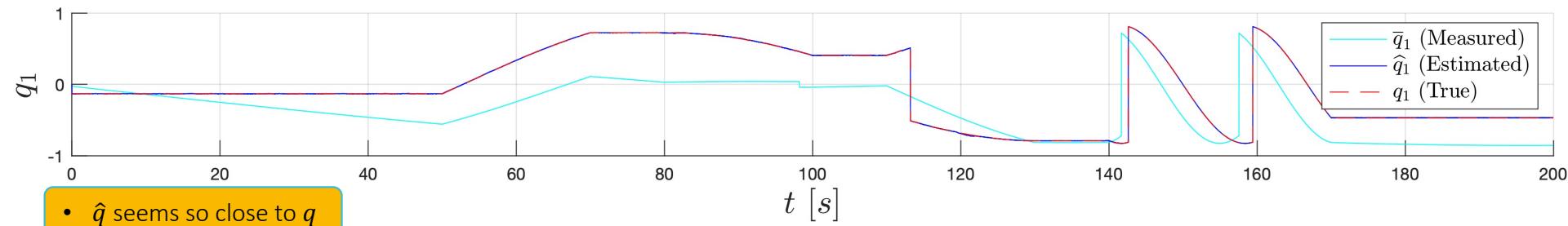
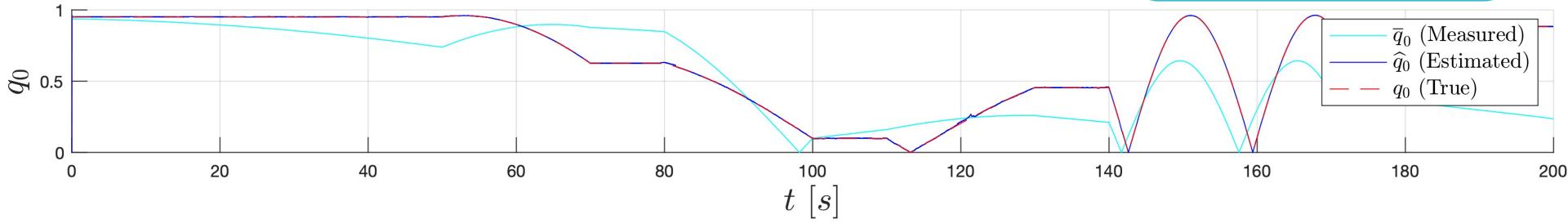


Simulated Data Analysis

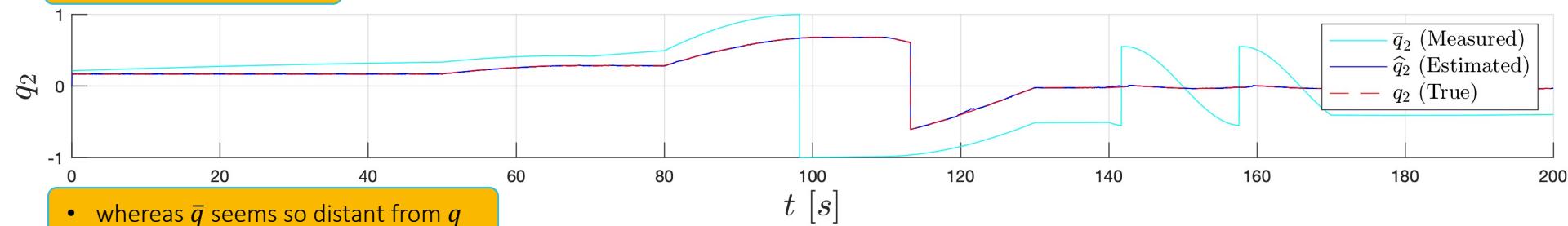
Quaternion Estimation

$$\cdot \quad q = \begin{bmatrix} q_0 \\ q \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

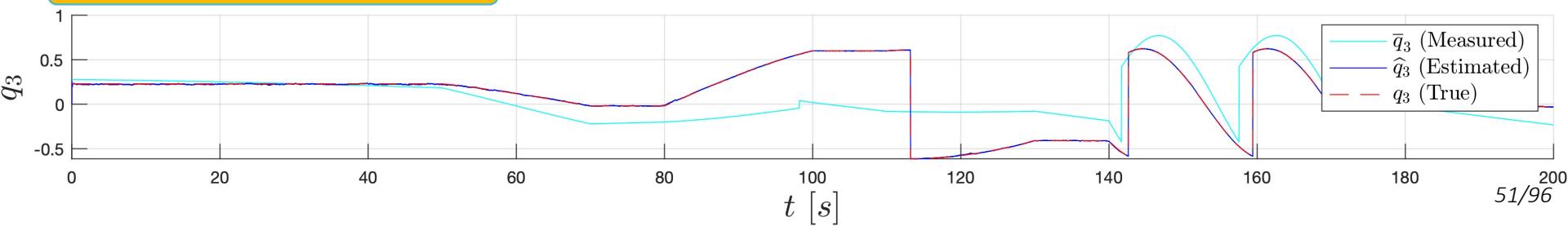
- q is obtained by integrating ω
- \bar{q} is obtained by integrating y_g
- \hat{q} is obtained by integrating y_g , then by correcting it in the accelerometer update stage



• \hat{q} seems so close to q



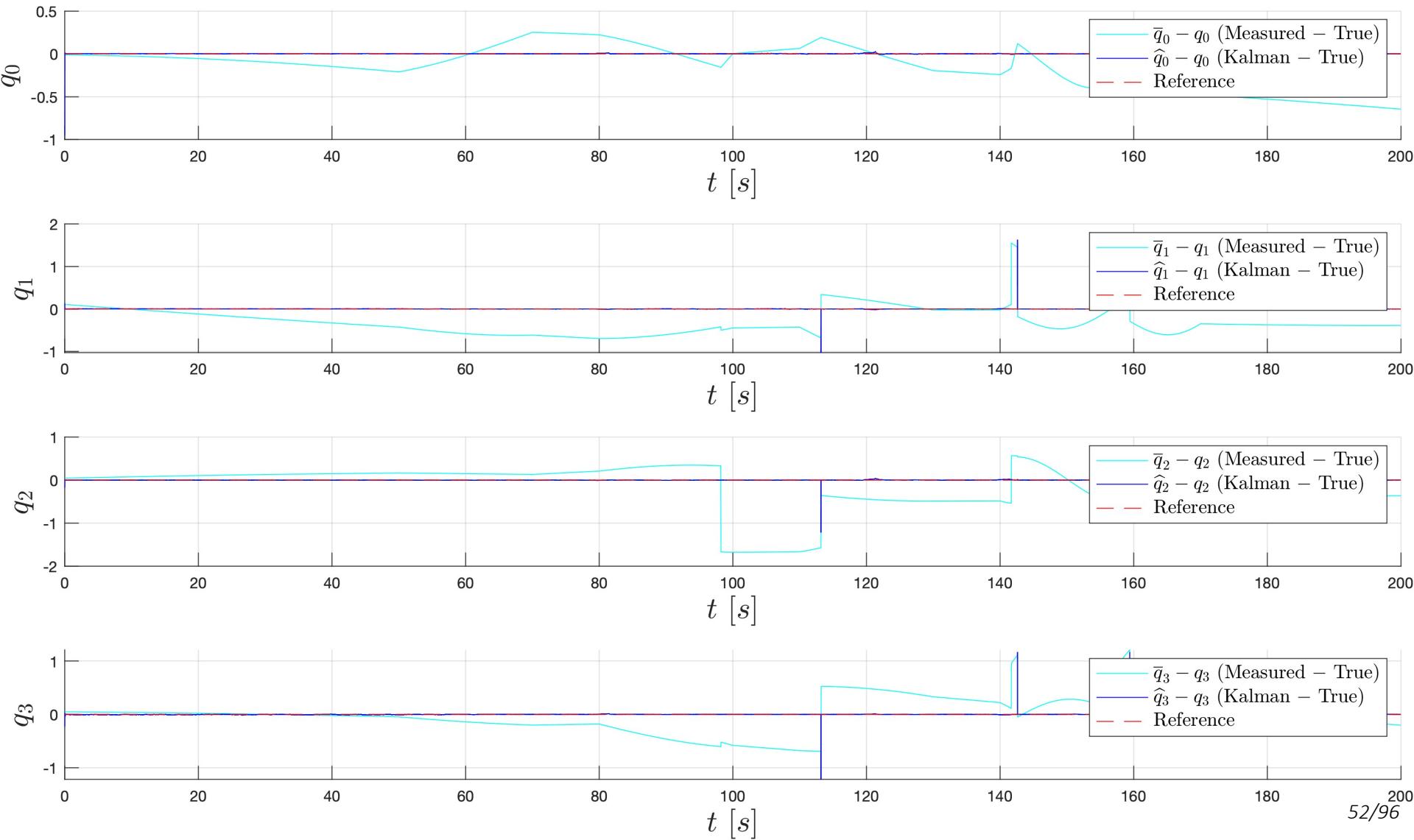
• whereas \bar{q} seems so distant from q



Simulated Data Analysis

Quaternion Estimation Error

- q , \bar{q} , and \hat{q} are all unit quaternions at every time-step, i.e.:
 $\|q(kT)\| = \|\bar{q}(kT)\| = \|\hat{q}(kT)\| = 1 \quad \forall k = 1, \dots, N$

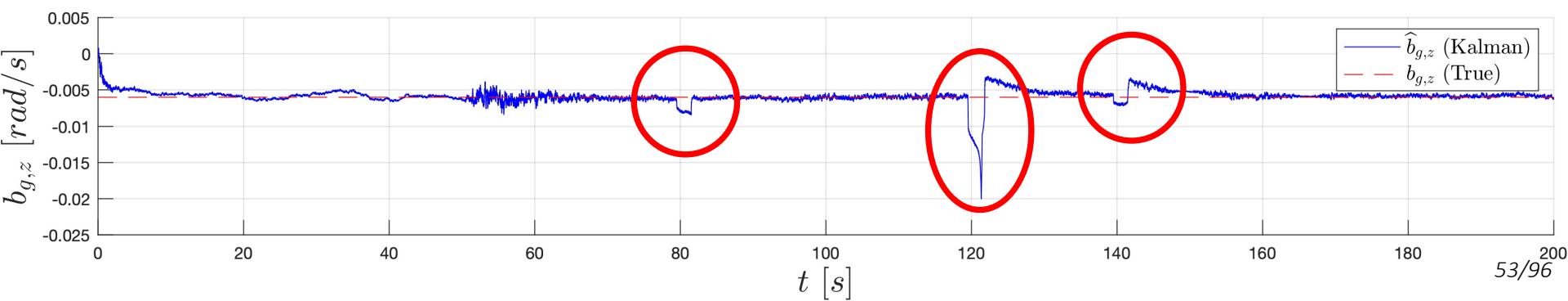
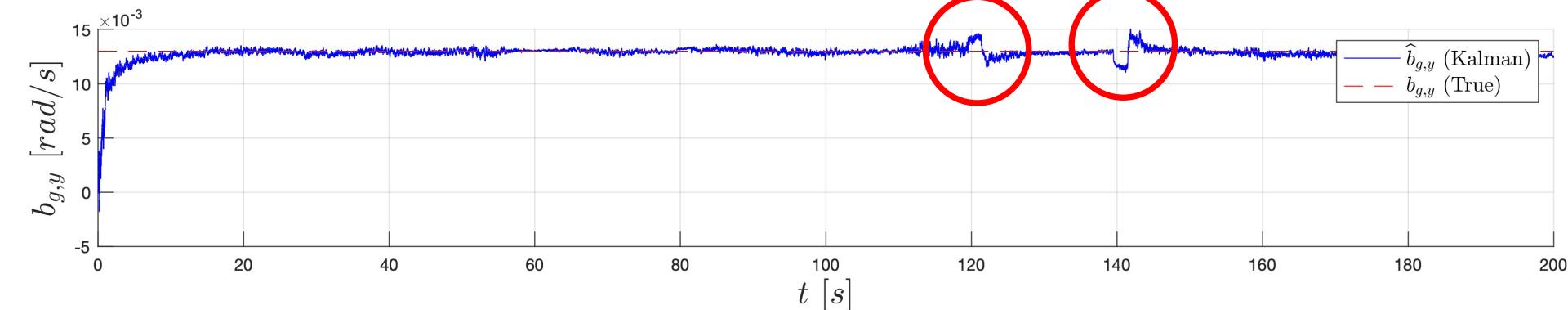
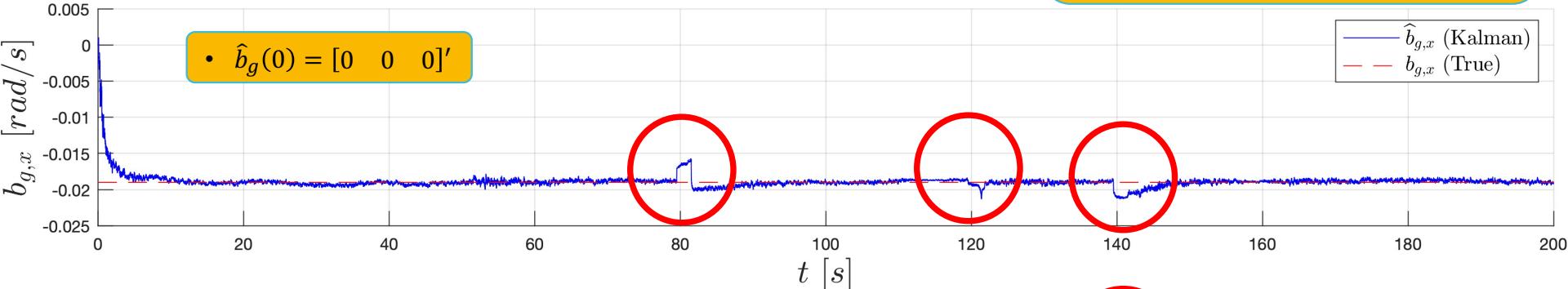


Simulated Data Analysis

Gyroscope Bias Estimation

- $\hat{b}_g = x(4:6)$
- b_g

- Estimated gyro bias \hat{b}_g converges rapidly to real bias b_g
- Estimated bias spikes (highlighted in red circles) are due to external acceleration spikes

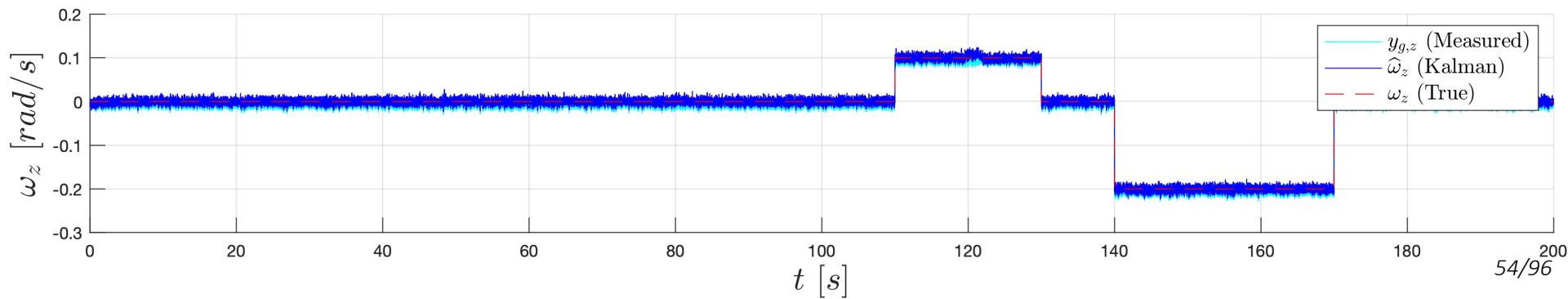
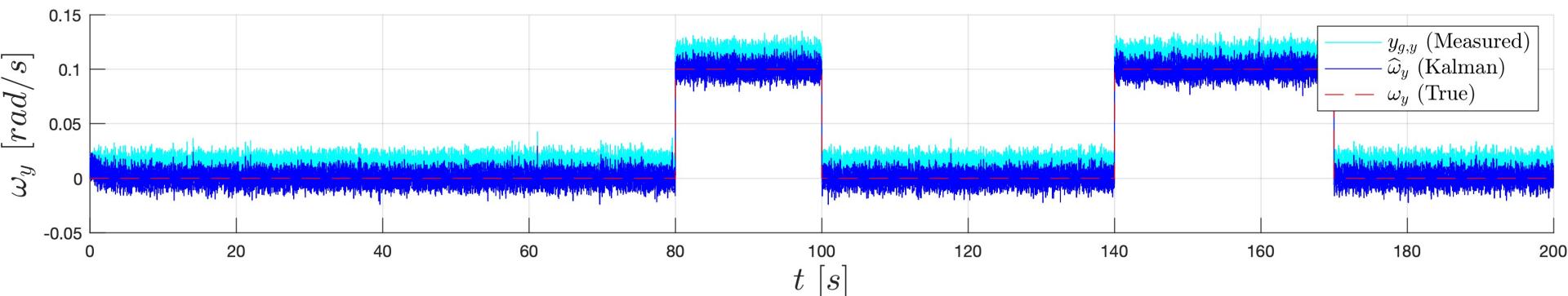
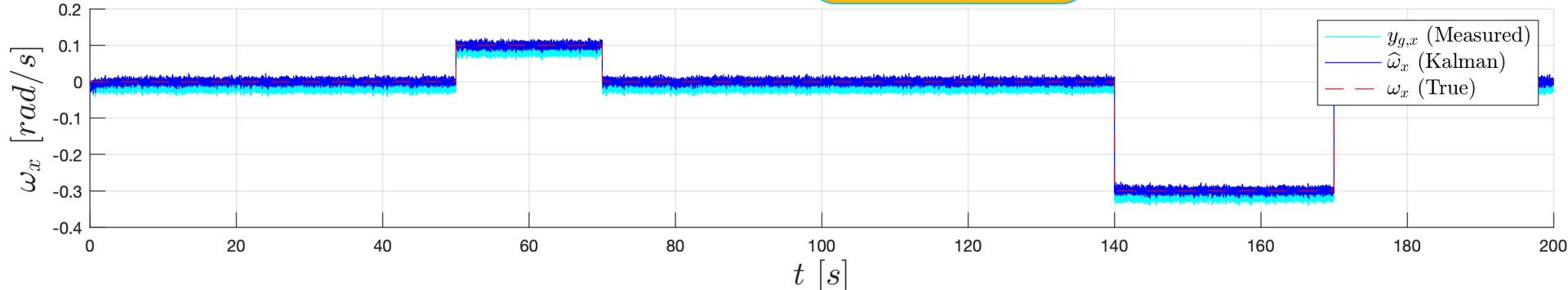


Simulated Data Analysis

Angular Velocity Estimation

- $y_g = \omega + b_g + n_g$
- $\hat{\omega} = y_g - \hat{b}_g$
- ω

- In $\hat{\omega}$ bias is almost completely removed, whereas noise is still present

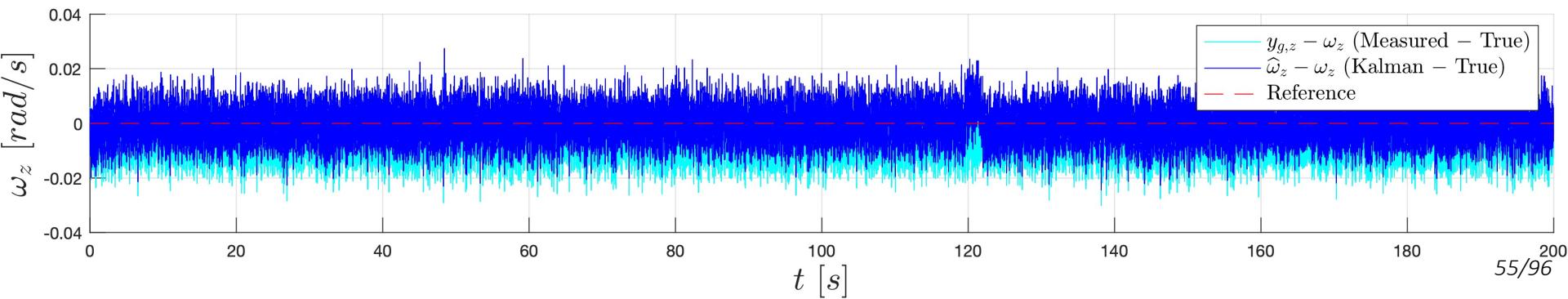
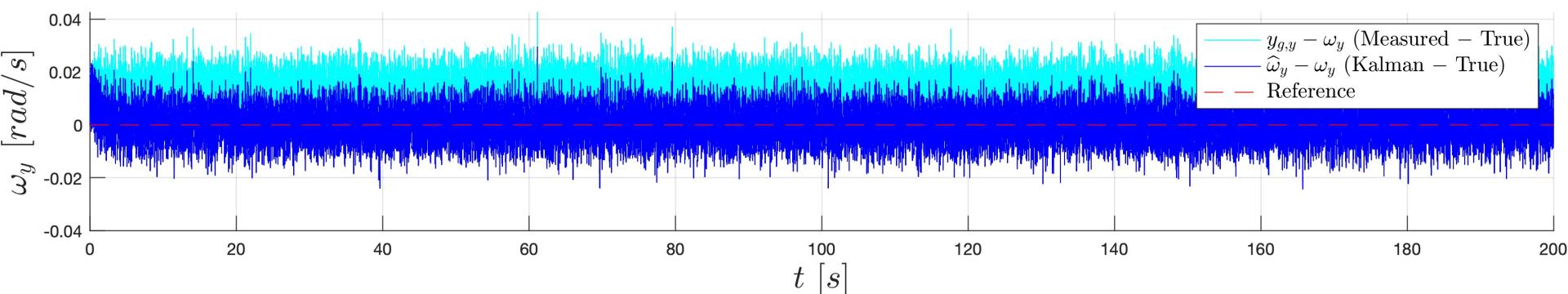
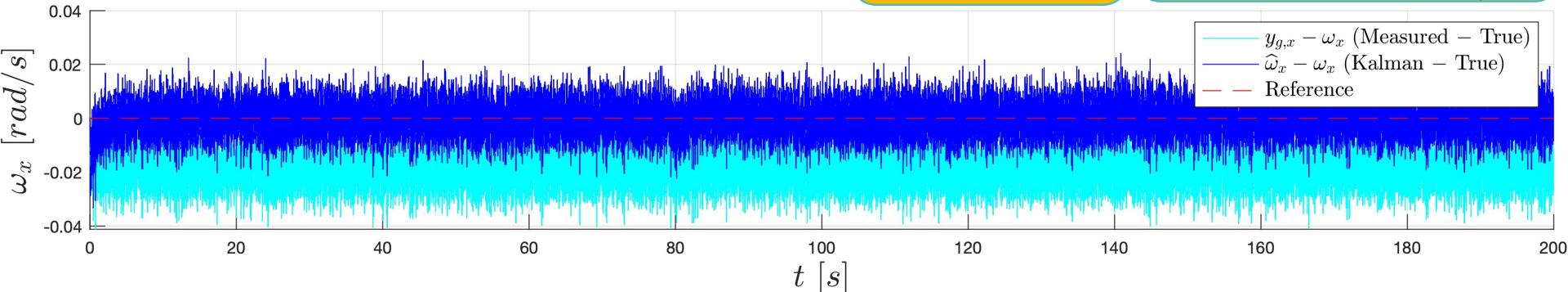


Simulated Data Analysis

Angular Velocity Estimation Error

- $y_g = \omega + b_g + n_g$
- $\hat{\omega} = y_g - \hat{b}_g$
- ω

- Gyro bias effects are noticeable in measured angular velocity y_g
- Whereas in estimated angular velocity $\hat{\omega}$ bias is almost compensated by subtracting \hat{b}_g

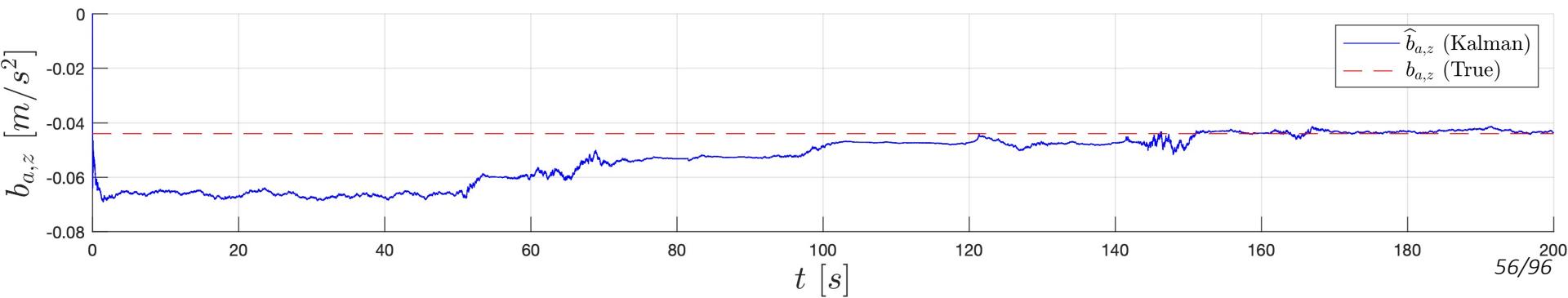
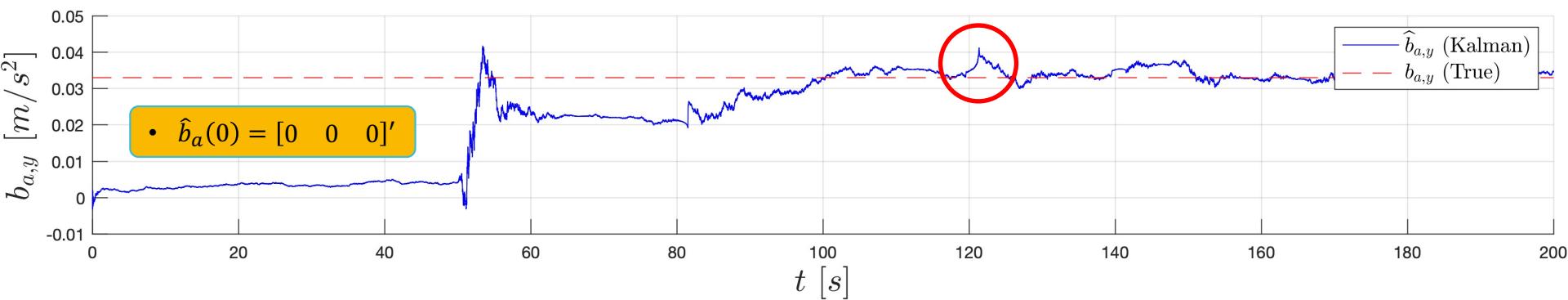
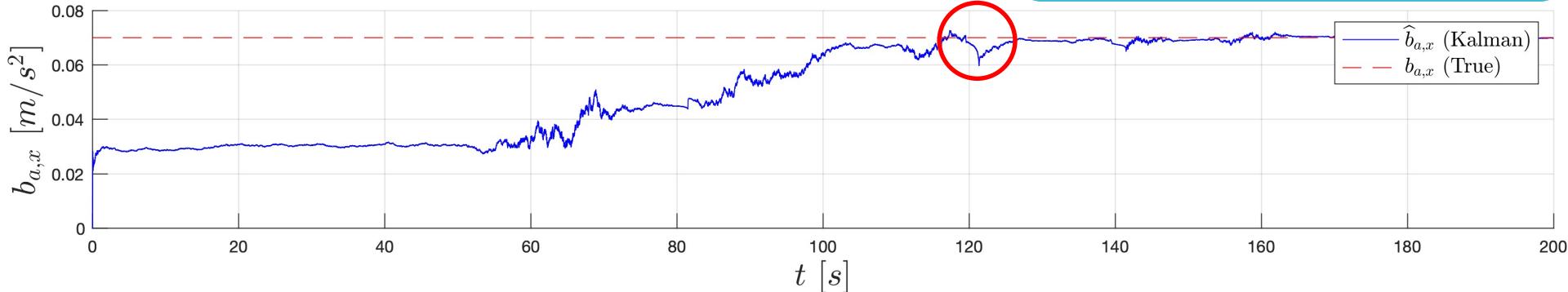


Simulated Data Analysis

Accelerometer Bias Estimation

- $\hat{b}_a = x(7:9)$
- b_a

- Estimated accelerometer bias \hat{b}_a converges to real bias b_a , indeed slower than gyro bias
- Estimated bias spikes are due to external acceleration spikes; they are less evident than in gyroscope bias estimation, though

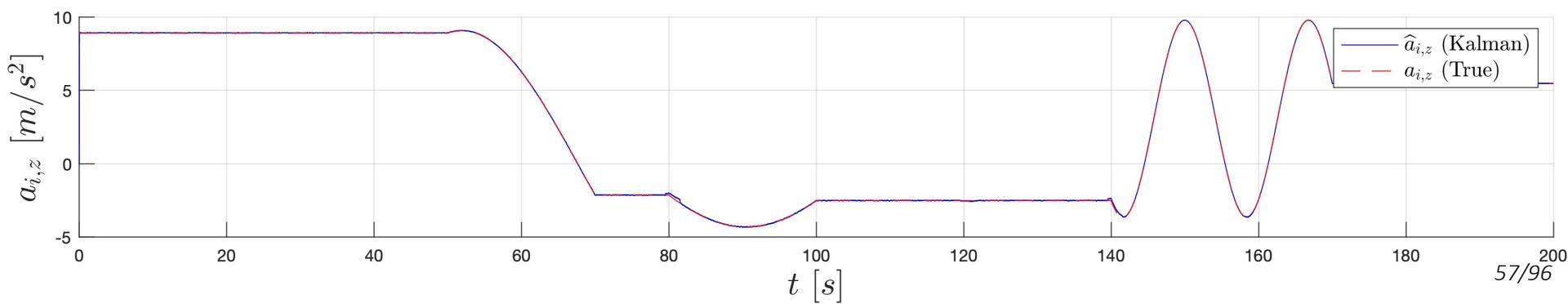
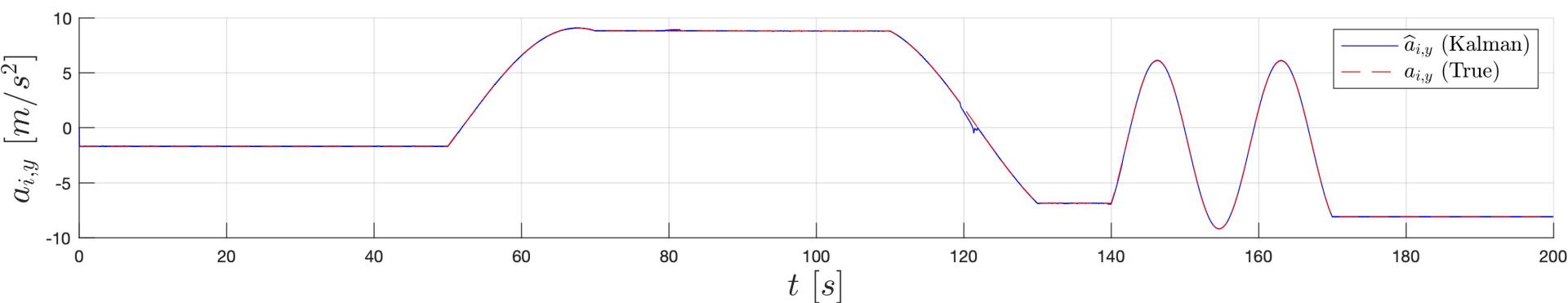
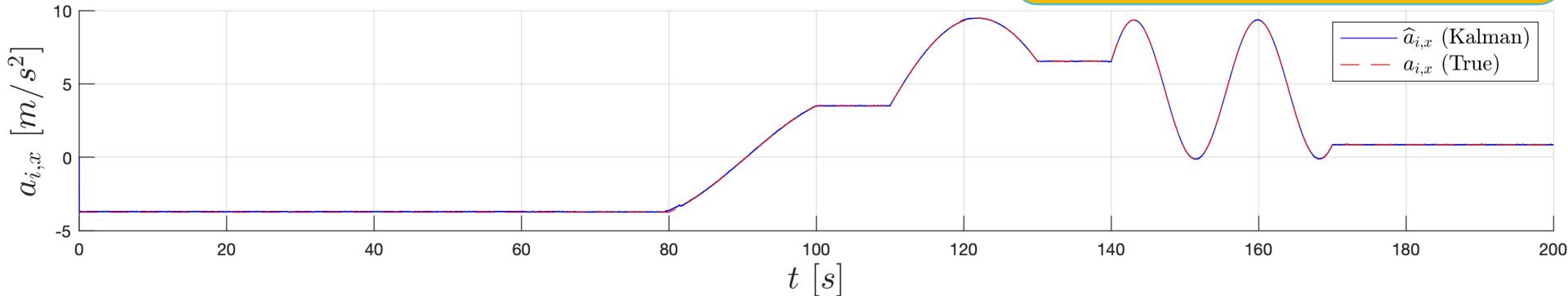


Simulated Data Analysis

Internal Acceleration Estimation

$$\begin{aligned} \hat{a}_i &= C(\hat{q})\tilde{g} \\ a_i & \end{aligned}$$

- Internal acceleration a_i is estimated rather well by \hat{a}_i
- $C(\hat{q})$ is the rotation matrix computed from the estimated quaternion \hat{q}
- \tilde{g} is the gravitational field expressed in the navigation frame

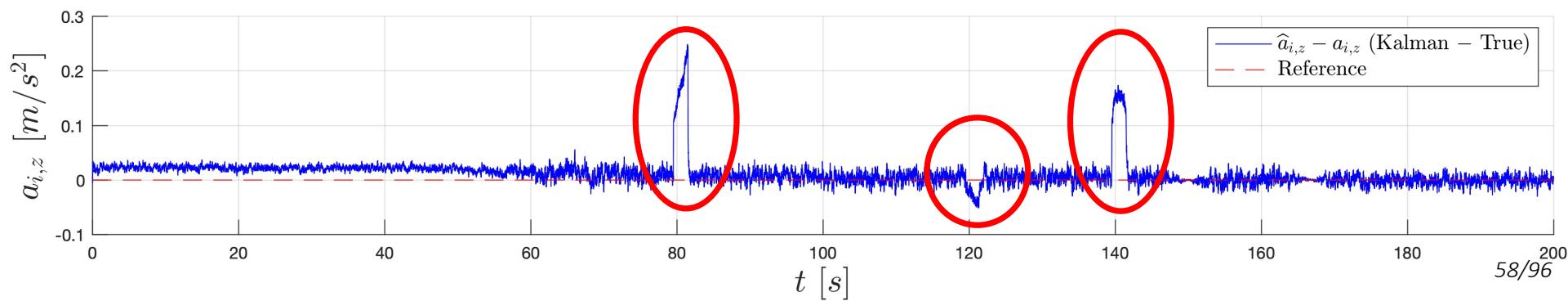
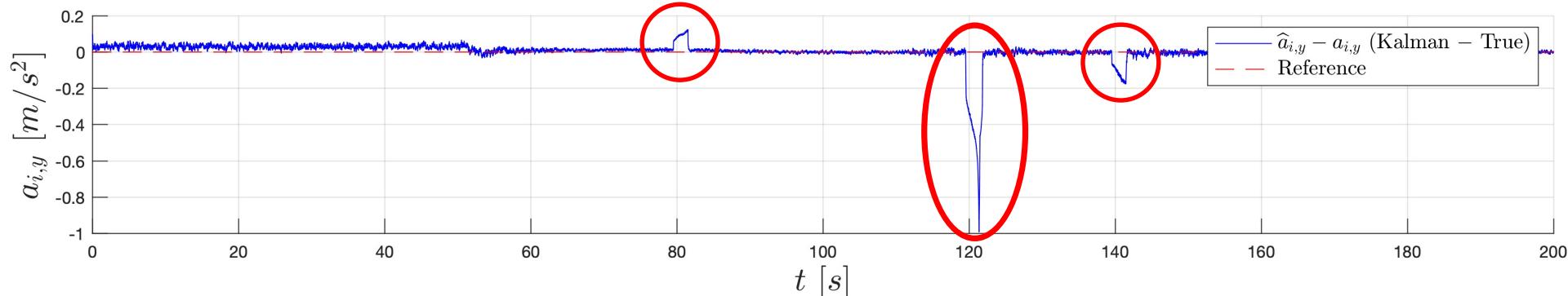
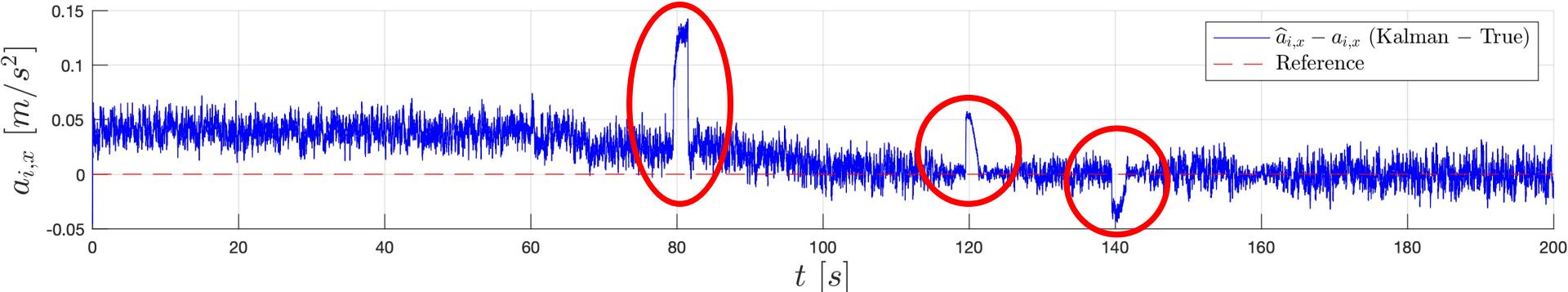


Simulated Data Analysis

Internal Acceleration Estimation Error

$$\begin{aligned} \hat{a}_i &= C(\hat{q})\tilde{g} \\ a_i \end{aligned}$$

- Estimated bias spikes are due to external acceleration spikes; they are less evident than in gyroscope bias estimation
- Yet, spikes due to external acceleration are still clear



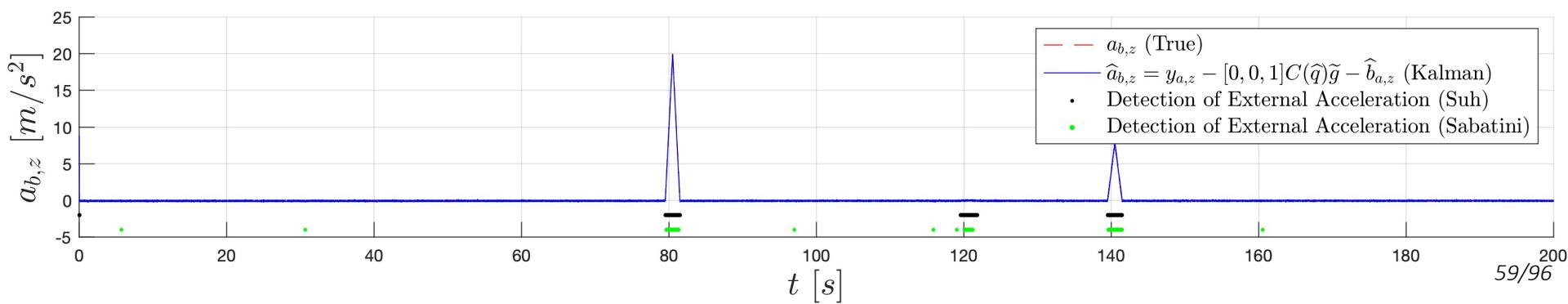
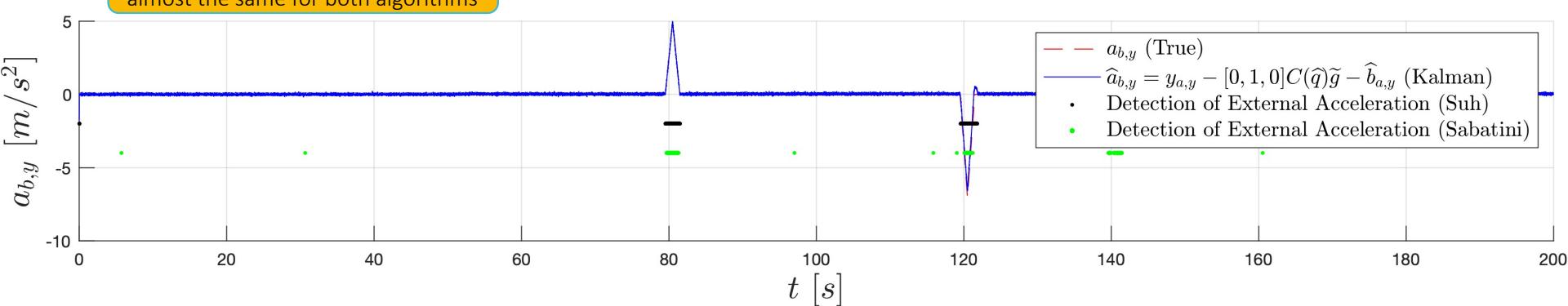
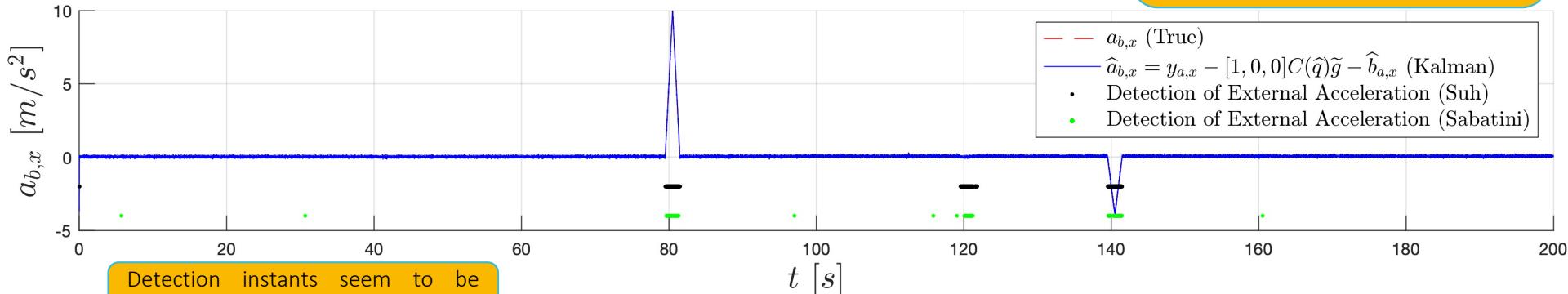
Simulated Data Analysis

External Acceleration Estimation

$$\begin{aligned} \hat{a}_b &= y_a - C(\hat{q})\tilde{g} - \hat{b}_a \\ a_b & \end{aligned}$$

Dots highlight instants where external acceleration have been detected:

- black dots when using Suh's algorithm
- green dots when using Sabatini's algorithm

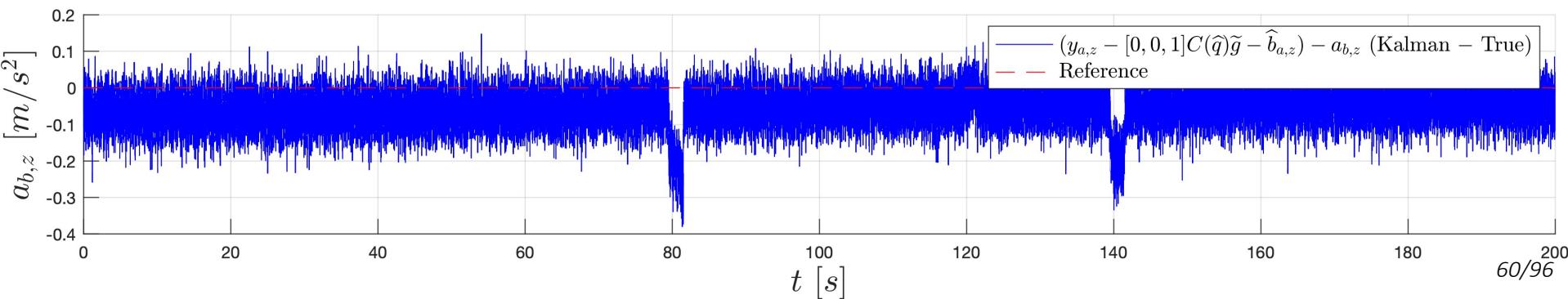
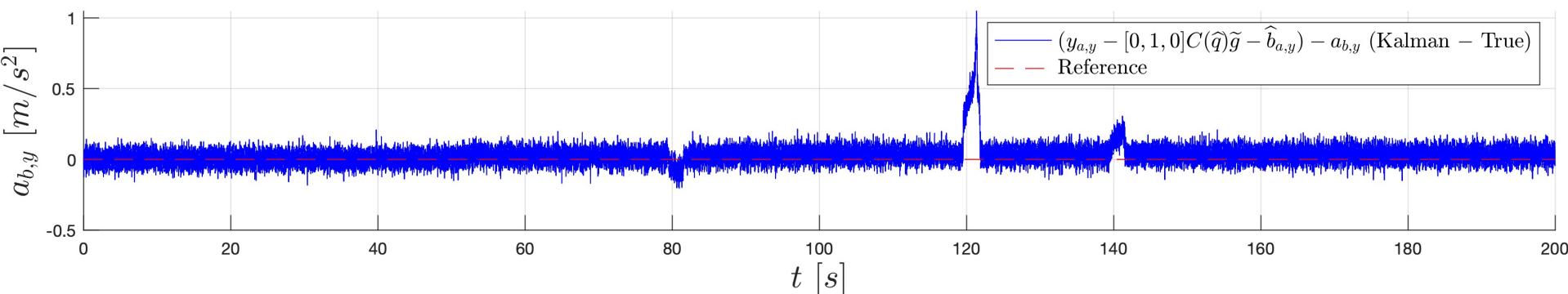
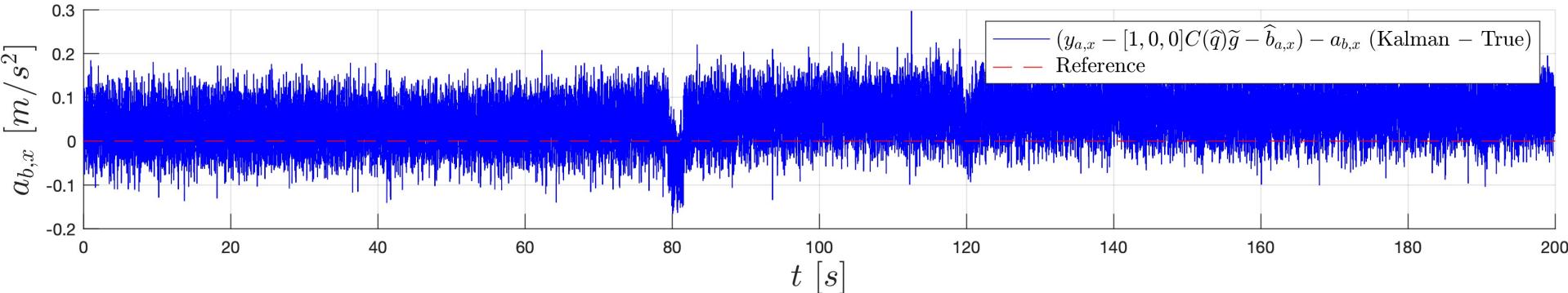


Simulated Data Analysis

External Acceleration Estimation Error

$$\begin{aligned} \hat{a}_b &= y_a - C(\hat{q})\tilde{g} - \hat{b}_a \\ a_b & \end{aligned}$$

External acceleration a_b was predicted well: even during the spikes (at 80 s, 120 s, and 140 s), errors stay rather small

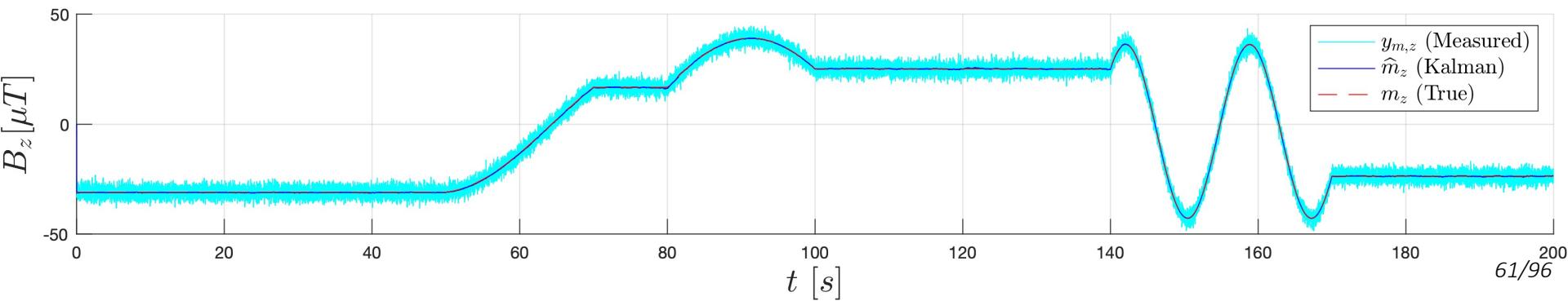
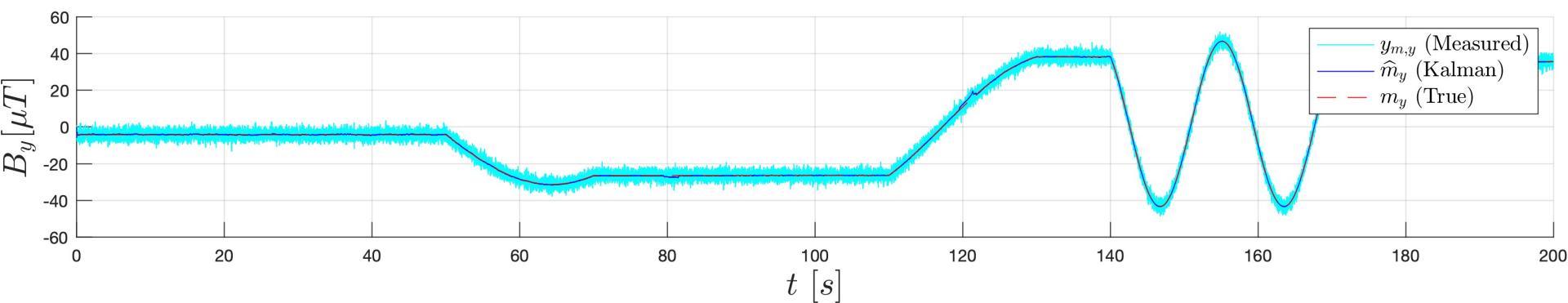
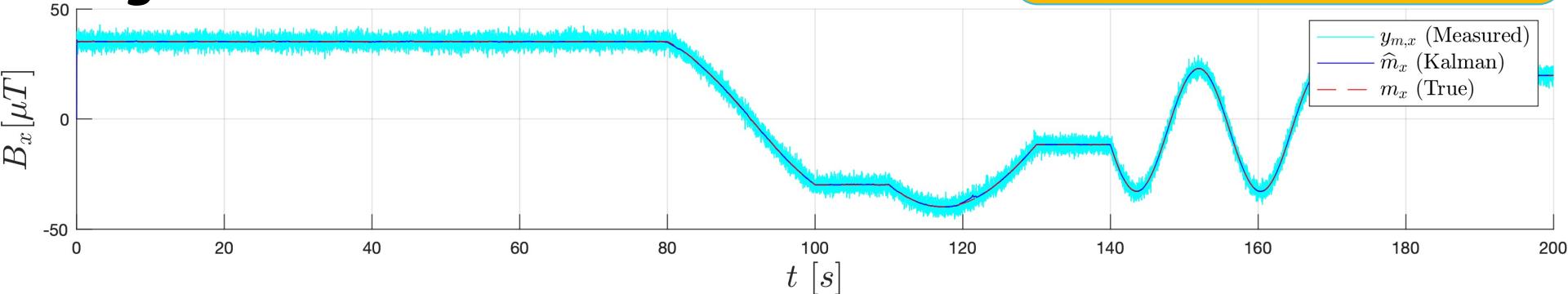


Simulated Data

Magnetic Field Estimation

- $y_m = C(q)\tilde{m} + n_m$
- $\hat{m} = C(\hat{q})\tilde{m}$
- $m = C(q)\tilde{m}$

- Magnetic field \mathbf{m} is estimated rather well by \hat{m}
- $C(\hat{q})$ is the rotation matrix from the estimated quaternion \hat{q}
- \tilde{m} is the magnetic field expressed in the navigation frame

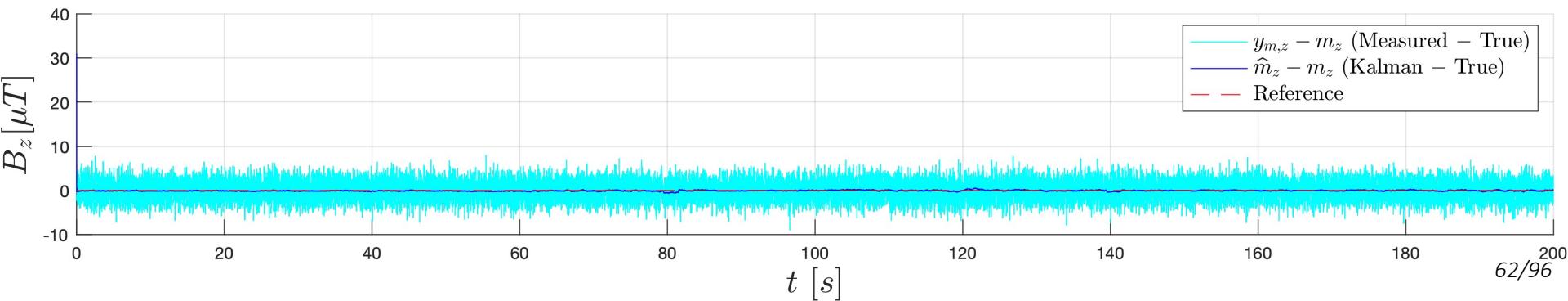
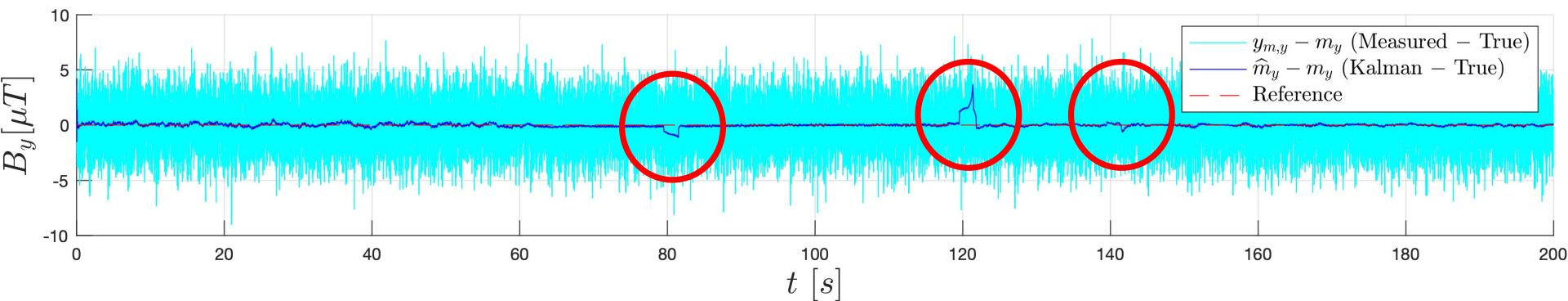
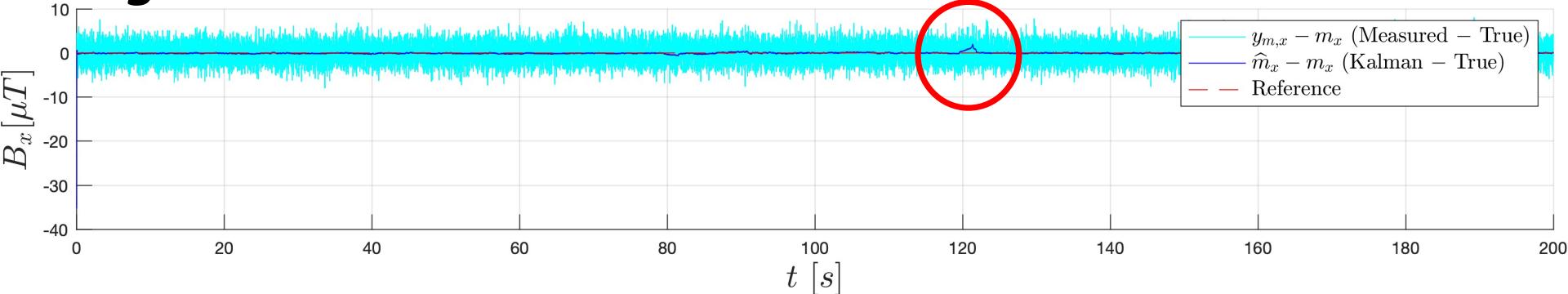


Simulated Data

Magnetic Field Estimation Error

- $y_m = C(q)\tilde{m} + n_m$
- $\hat{m} = C(\hat{q})\tilde{m}$
- $m = C(q)\tilde{m}$

- Spikes due to external acceleration are still clear

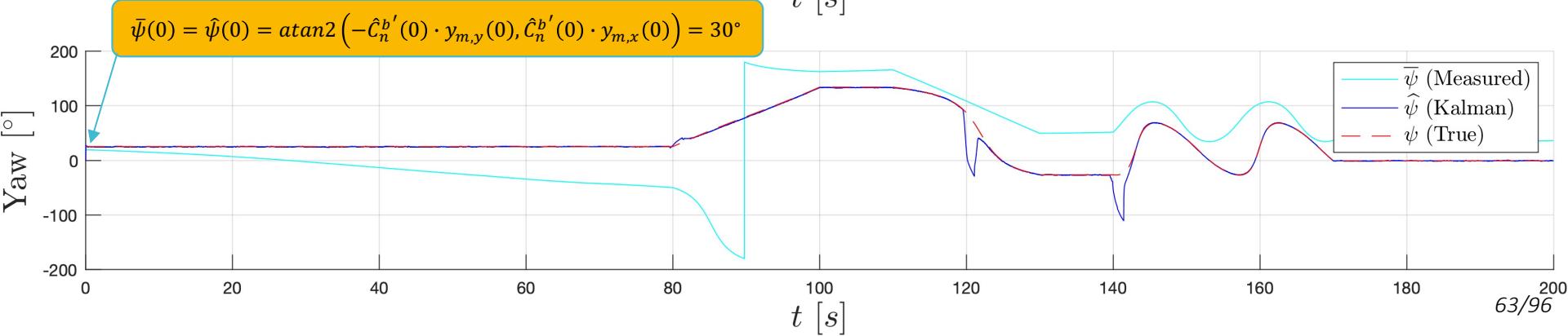
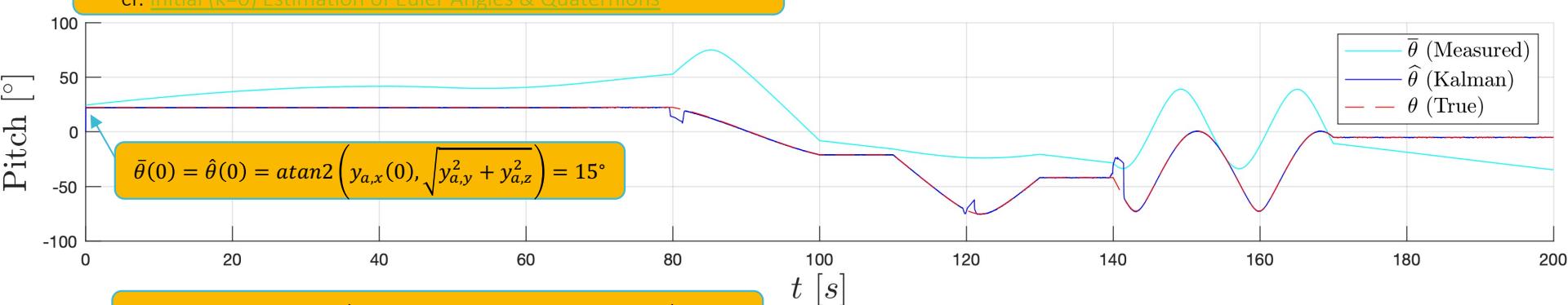
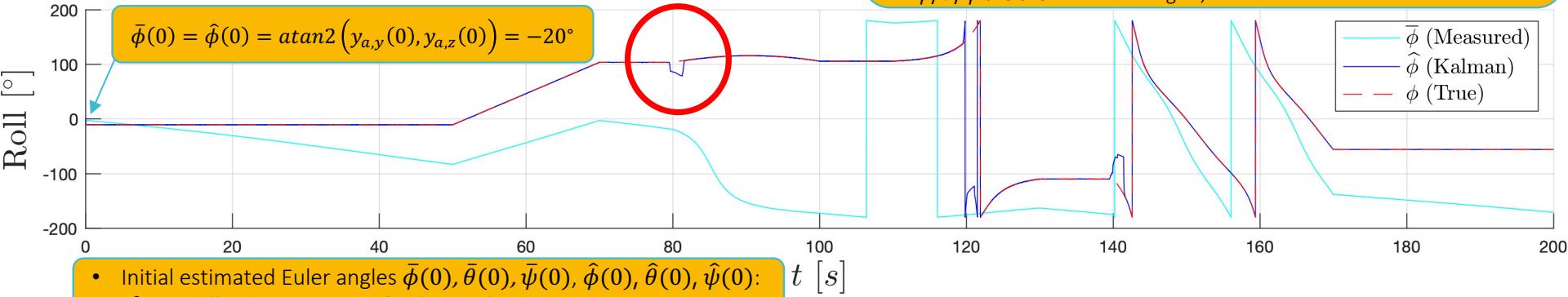


Simulated Data Analysis

Euler Angles Estimation [Sabatini]

cf. [Performance Metrics](#):

- $\bar{\phi}, \bar{\theta}, \bar{\psi}$ (estimated-from-measurement Euler angles) diverge mainly due to gyro and accelerometer biases
- $\hat{\phi}, \hat{\theta}, \hat{\psi}$ (estimated-from-Kalman-Filter Euler angles) estimate accurately; spikes due to external acceleration are noticeable, though
- ϕ, θ, ψ are the true Euler angles, taken as reference

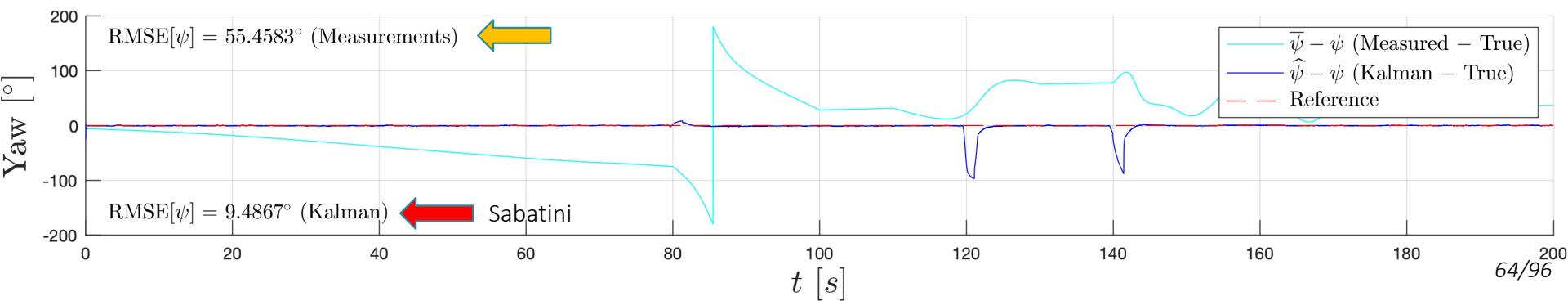
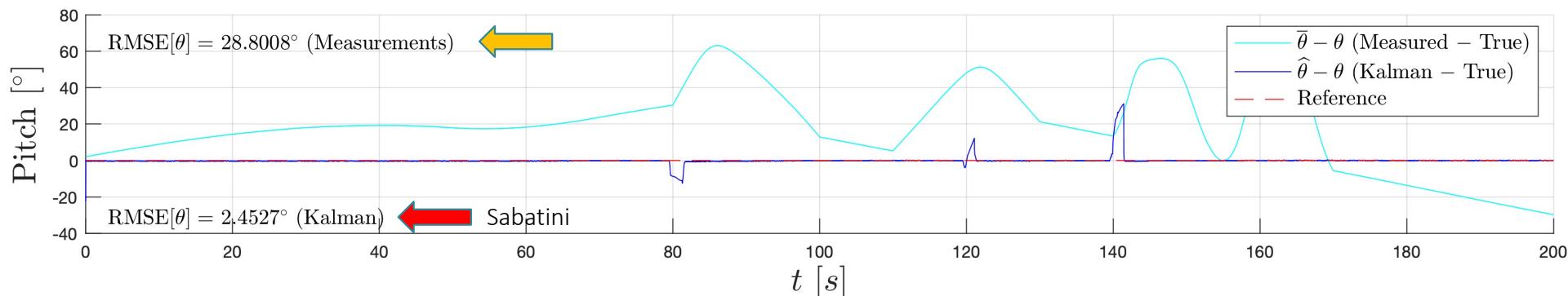
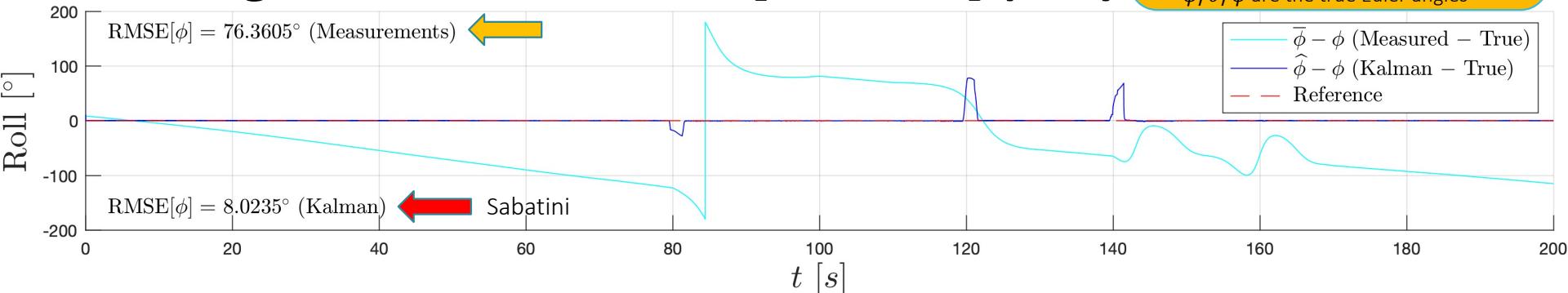


Simulated Data Analysis

Euler Angles Estimation Error [Sabatini] (1/2)

cf. [Performance Metrics](#):

- $\bar{\phi}, \bar{\theta}, \bar{\psi}$ are the estimated-from-measurement Euler angles
- $\hat{\phi}, \hat{\theta}, \hat{\psi}$ are the estimated-from-Kalman-Filter Euler angles
- ϕ, θ, ψ are the true Euler angles

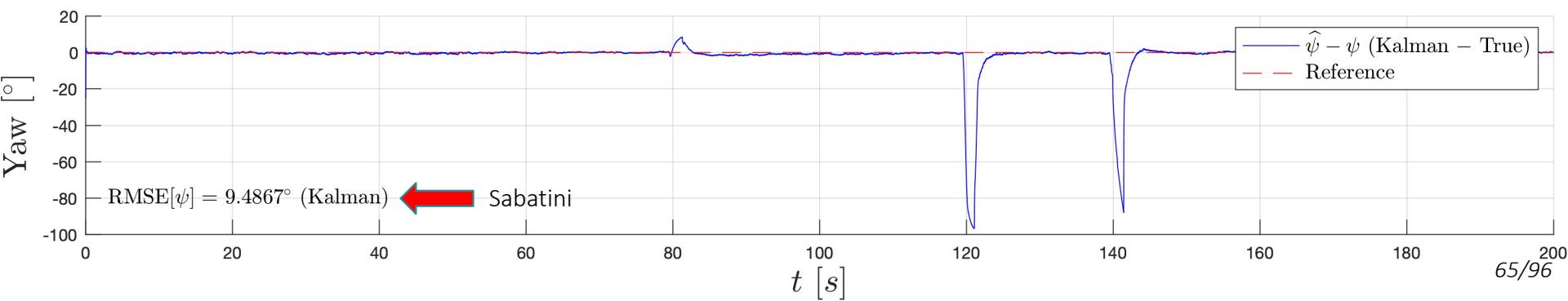
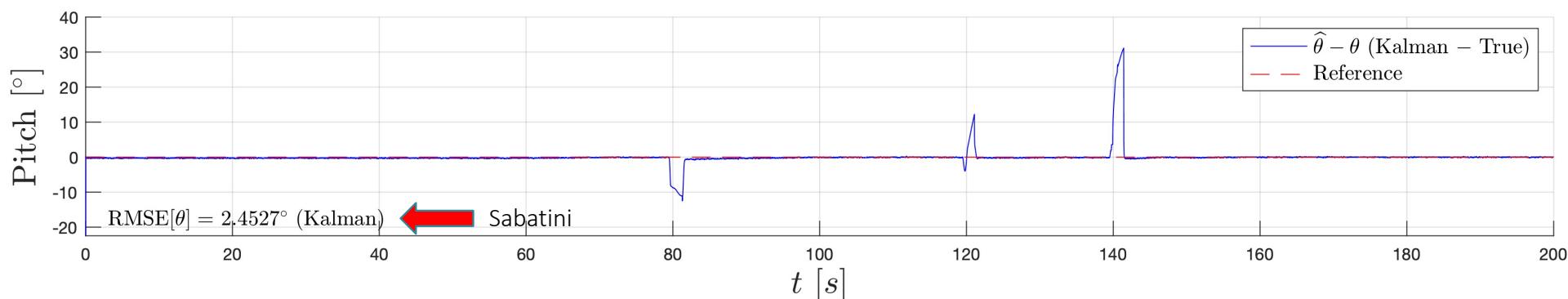
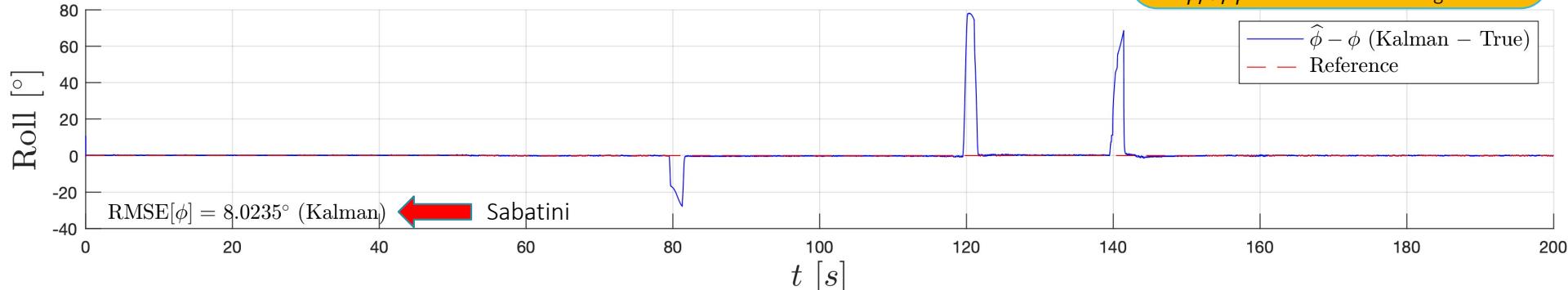


Simulated Data Analysis

Euler Angles Estimation Error [Sabatini] (2/2)

cf. [Performance Metrics](#):

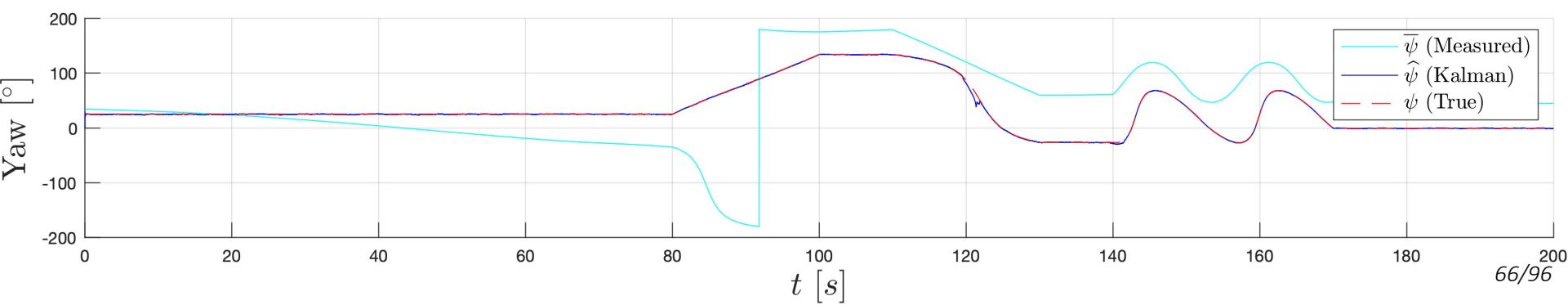
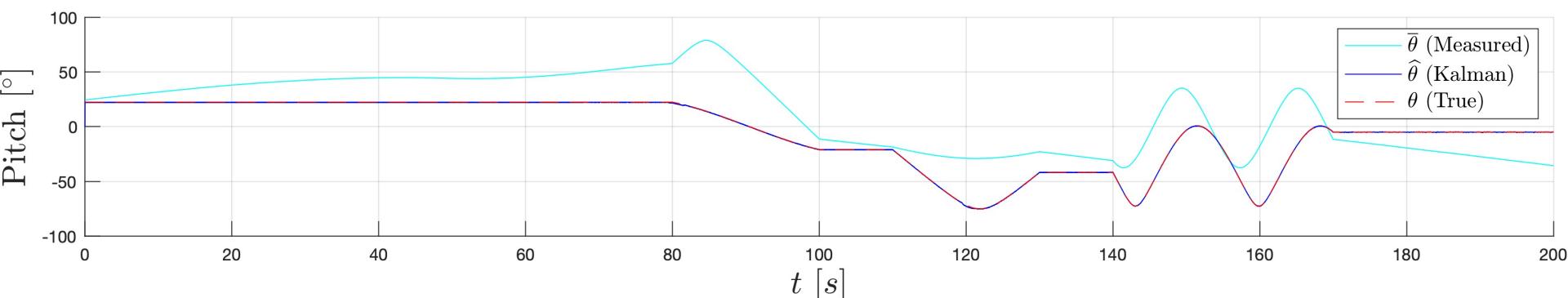
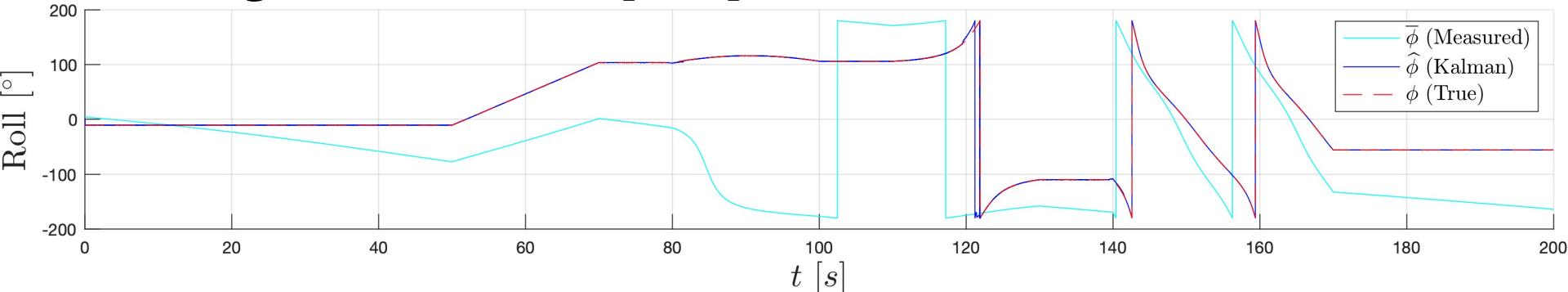
- $\bar{\phi}, \bar{\theta}, \bar{\psi}$ are the estimated-from-measurement Euler angles
- $\hat{\phi}, \hat{\theta}, \hat{\psi}$ are the estimated-from-Kalman-Filter Euler angles
- ϕ, θ, ψ are the true Euler angles



Simulated Data Analysis

Euler Angles Estimation [Suh]

Spikes due to external acceleration are barely noticeable now

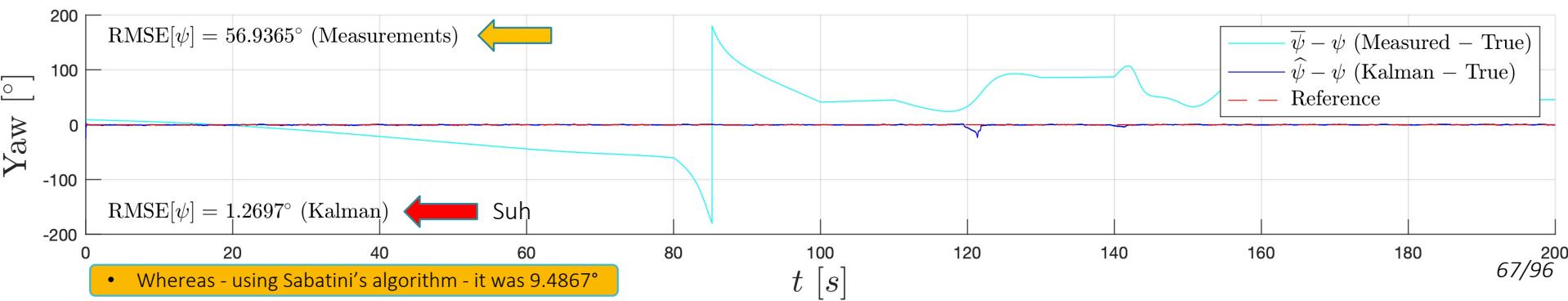
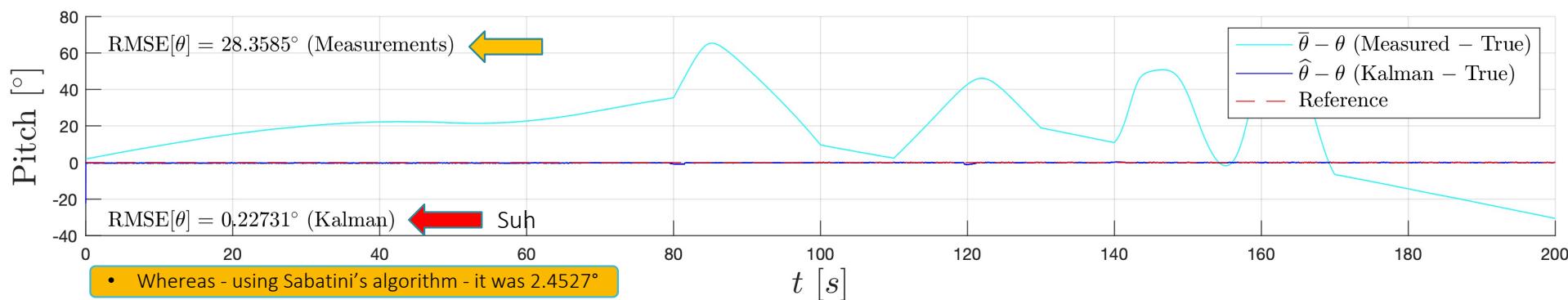
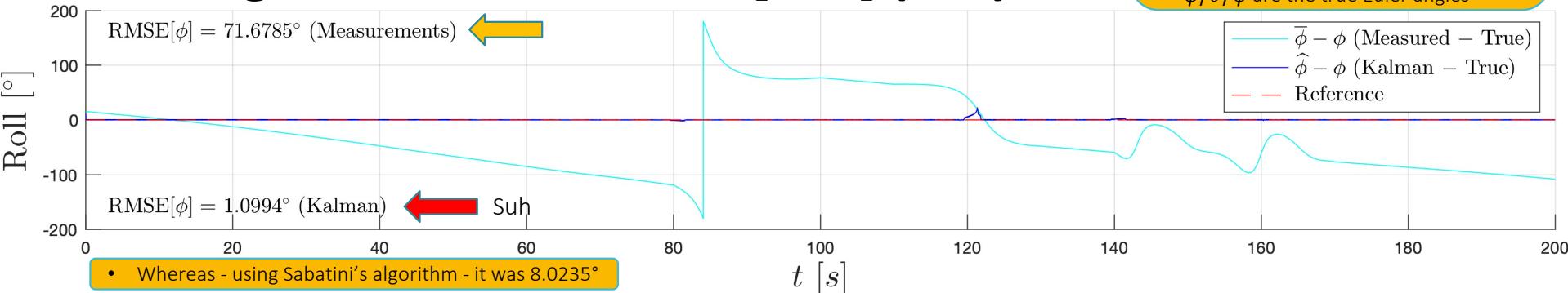


Simulated Data Analysis

Euler Angles Estimation Error [Suh] (1/2)

cf. [Performance Metrics](#):

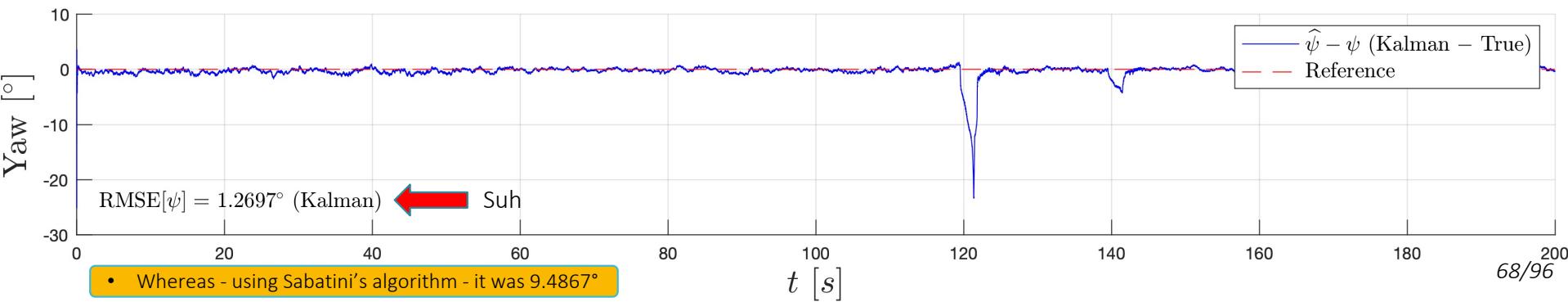
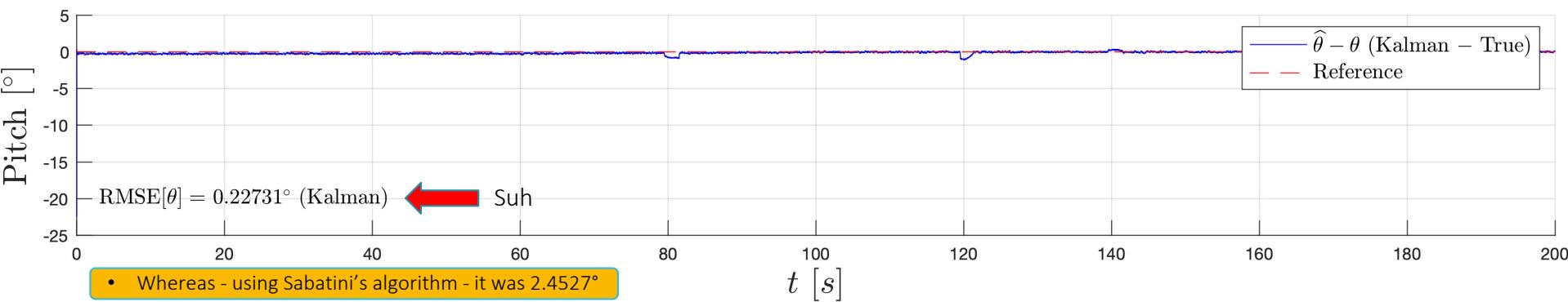
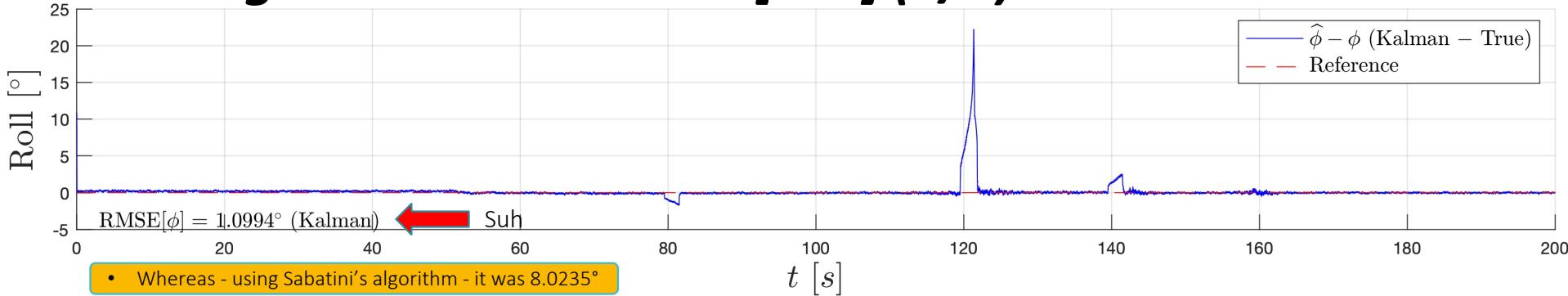
- $\bar{\phi}, \bar{\theta}, \bar{\psi}$ are the estimated-from-measurement Euler angles
- $\hat{\phi}, \hat{\theta}, \hat{\psi}$ are the estimated-from-Kalman-Filter Euler angles
- ϕ, θ, ψ are the true Euler angles



Simulated Data Analysis

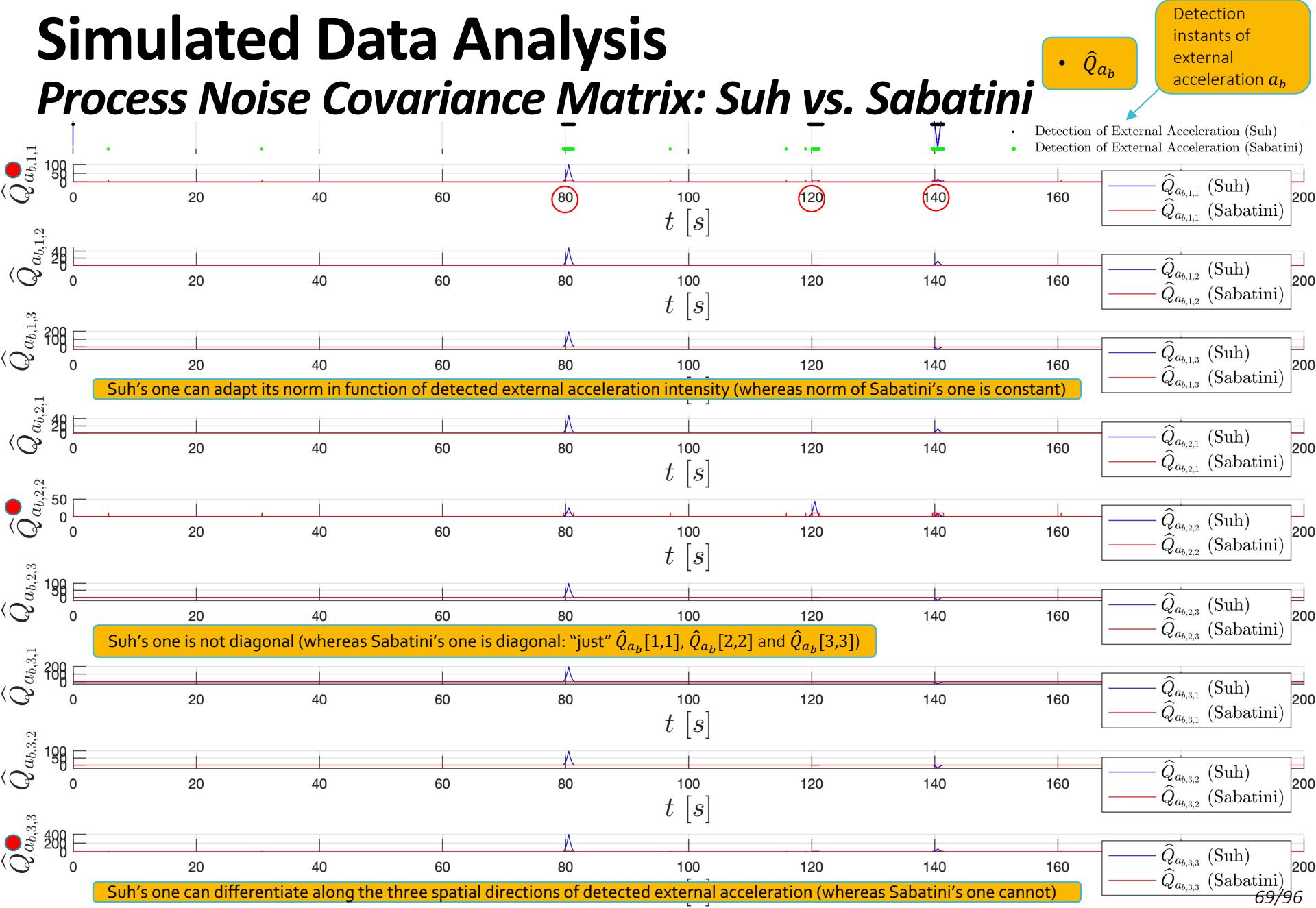
Euler Angles Estimation Error [Suh] (2/2)

Spikes due to external acceleration are smaller than those obtained using Sabatini's algorithm



Simulated Data Analysis

Process Noise Covariance Matrix: Suh vs. Sabatini

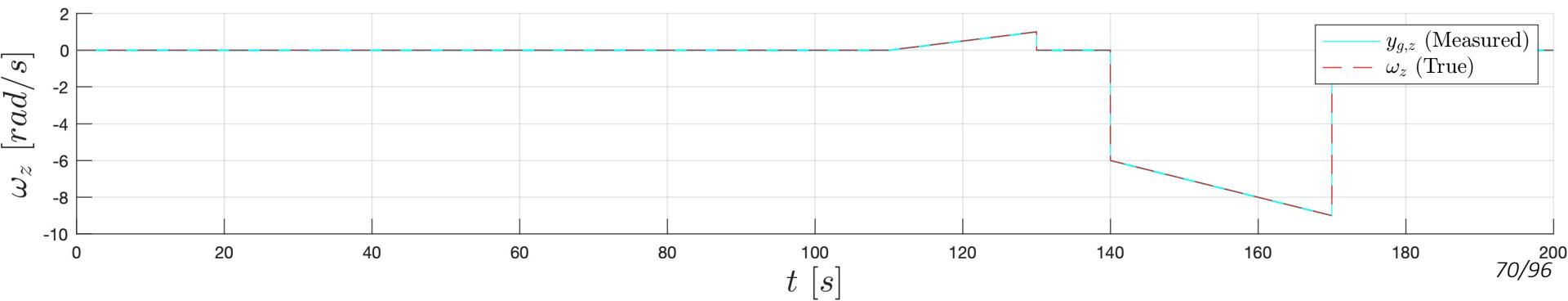
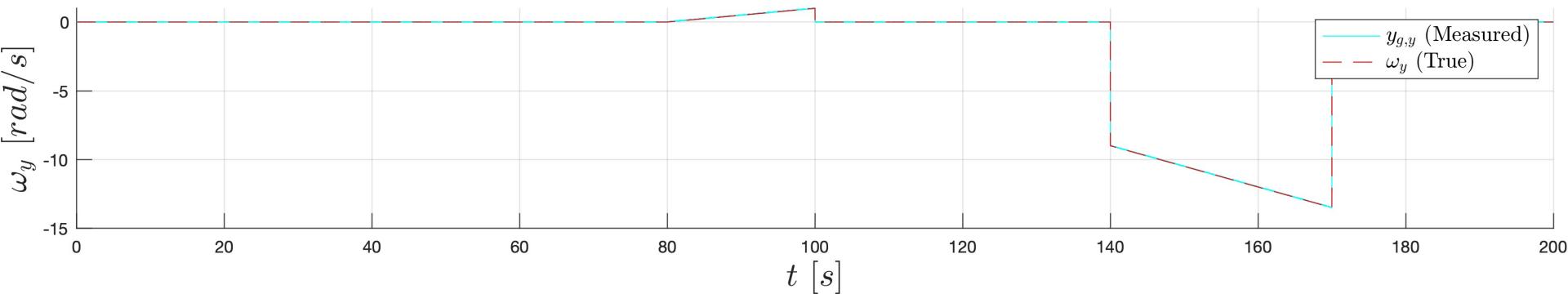
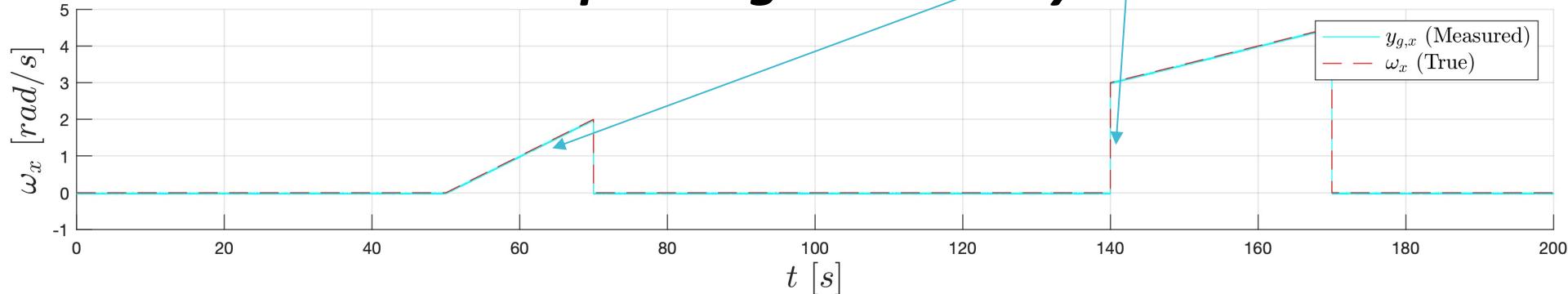


Simulated Data Analysis

A More Extreme Example: Angular Velocity

The only thing that's changed is angular velocity ω :

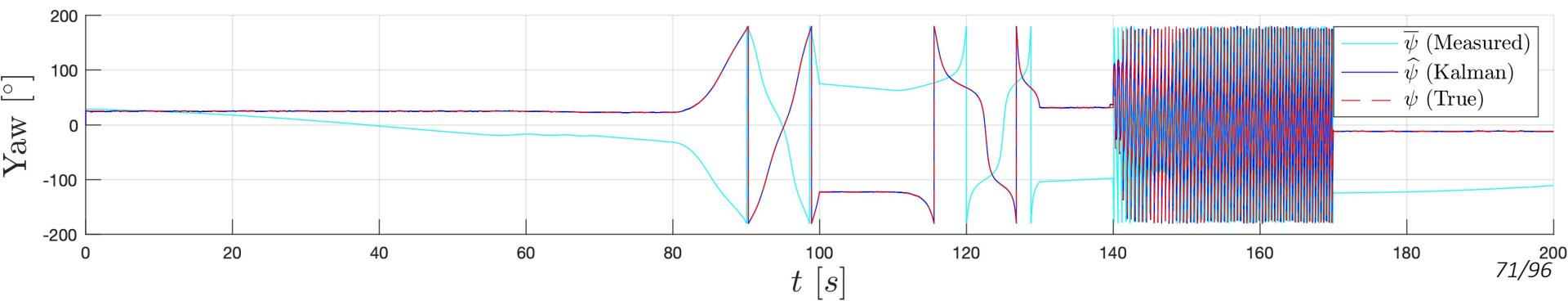
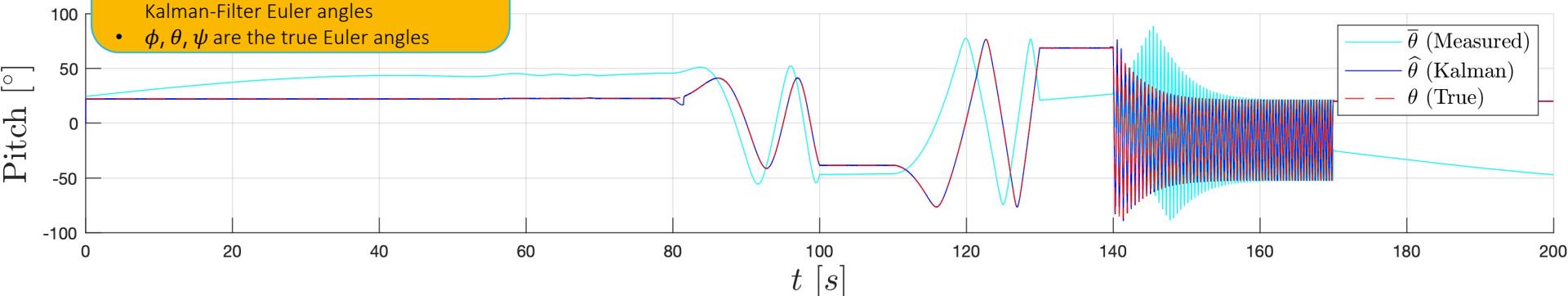
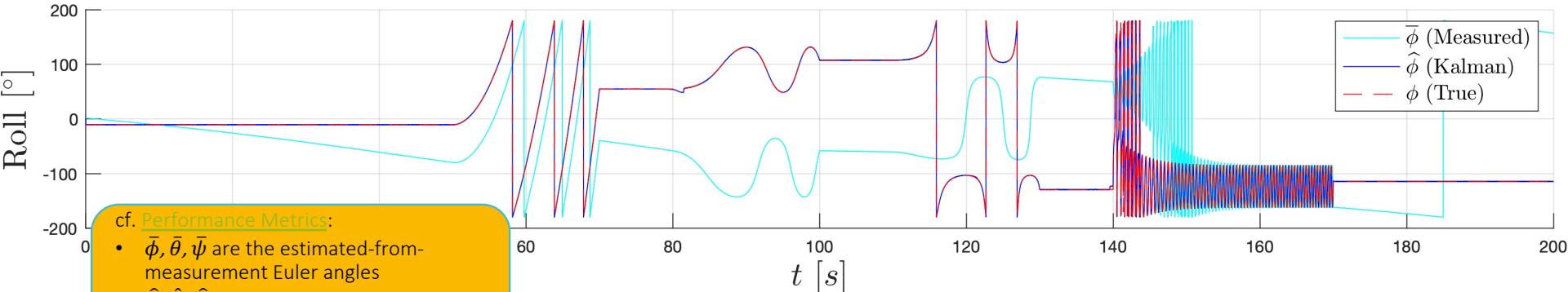
- now it is ramp-like
- and it has larger discontinuities



Simulated Data Analysis

A More Extreme Example: Euler Angles Estimation [Sabatini]

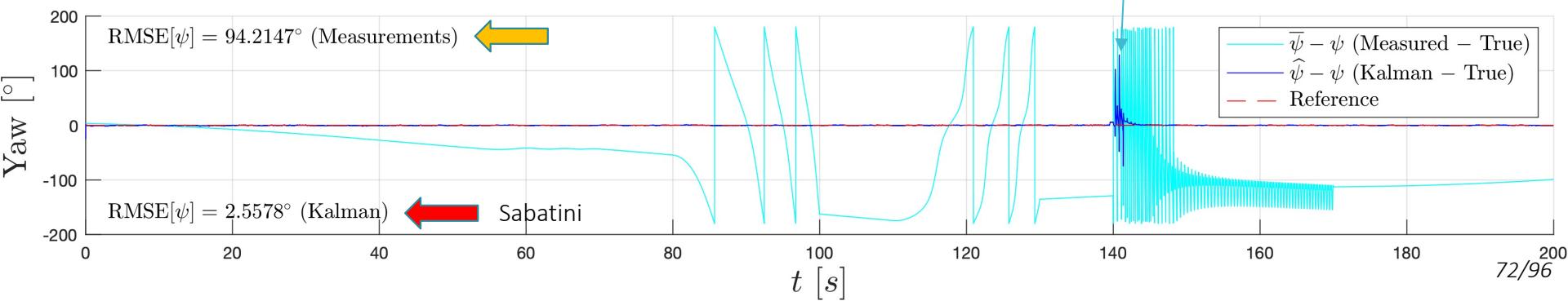
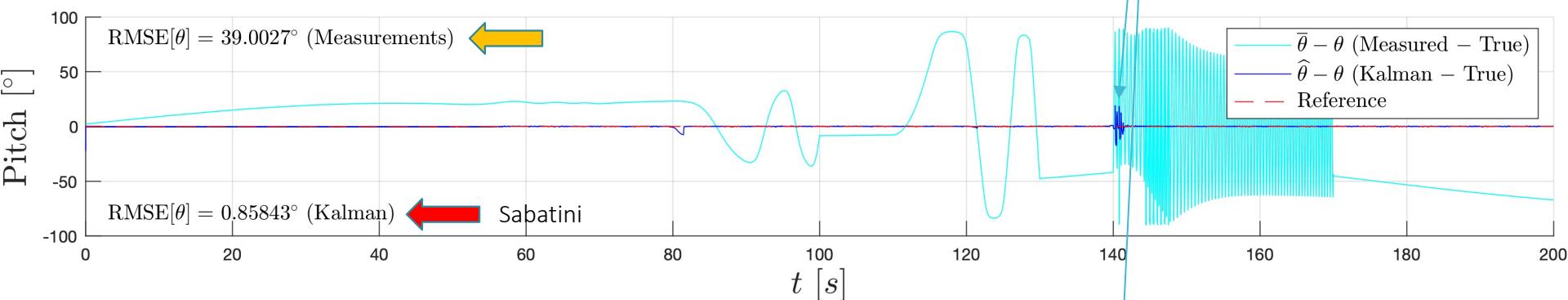
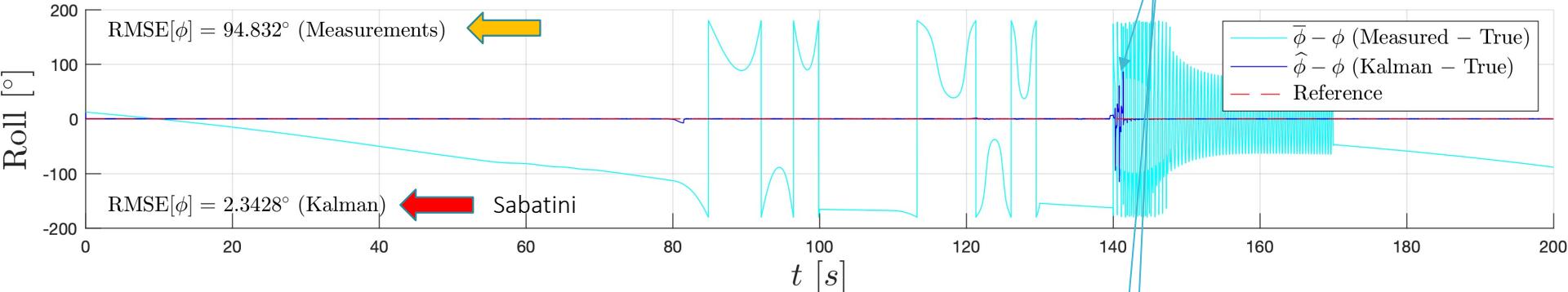
The implemented method — along with Sabatini's algorithm — still works fine



Simulated Data Analysis

A More Extreme Example: Euler Angles Est. Error [Sabatini] (1/2)

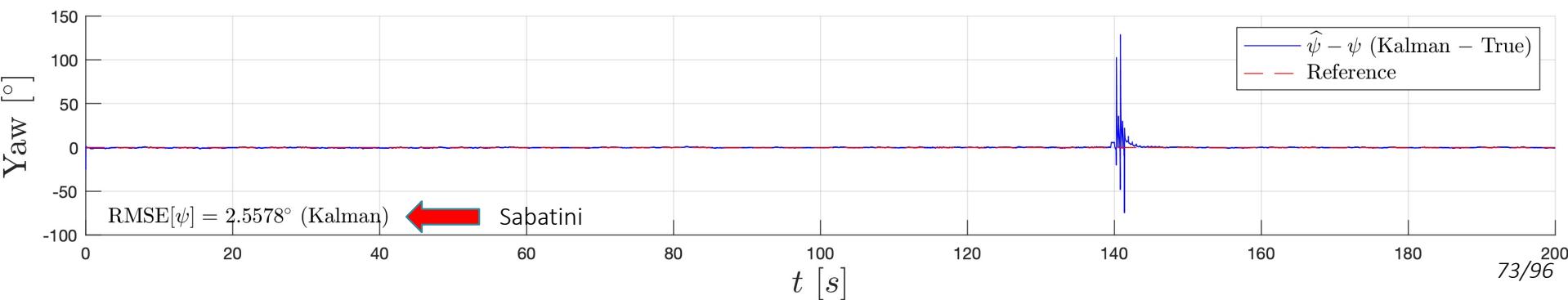
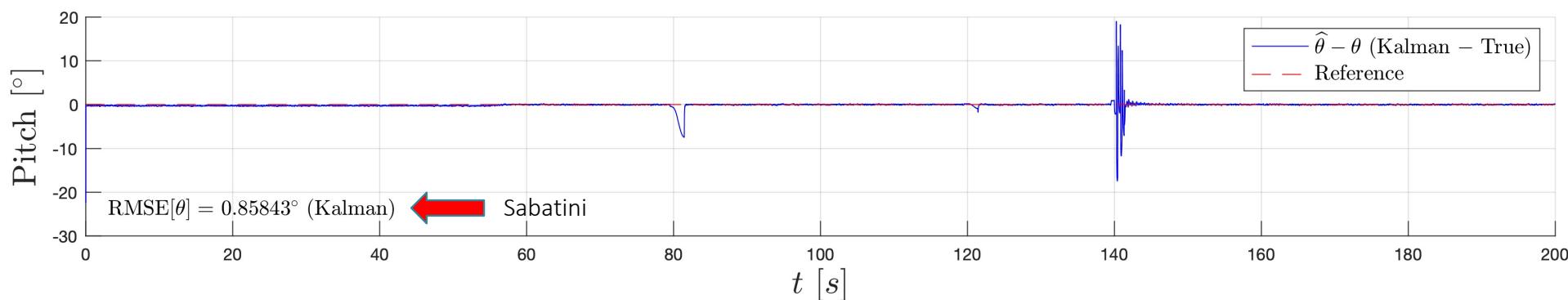
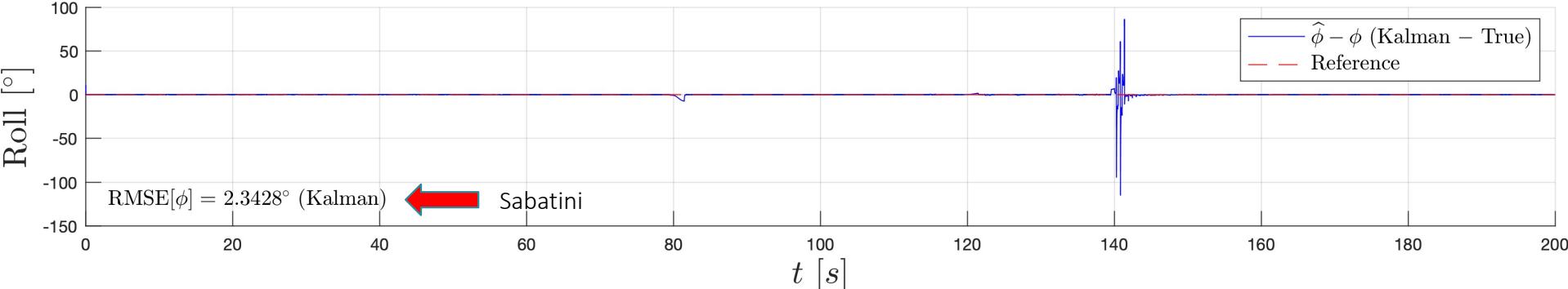
Errors are high just at about 140 s (when an acceleration spike along with a discontinuity in angular velocity occurs)



Simulated Data Analysis

A More Extreme Example: Euler Angles Est. Error [Sabatini] (2/2)

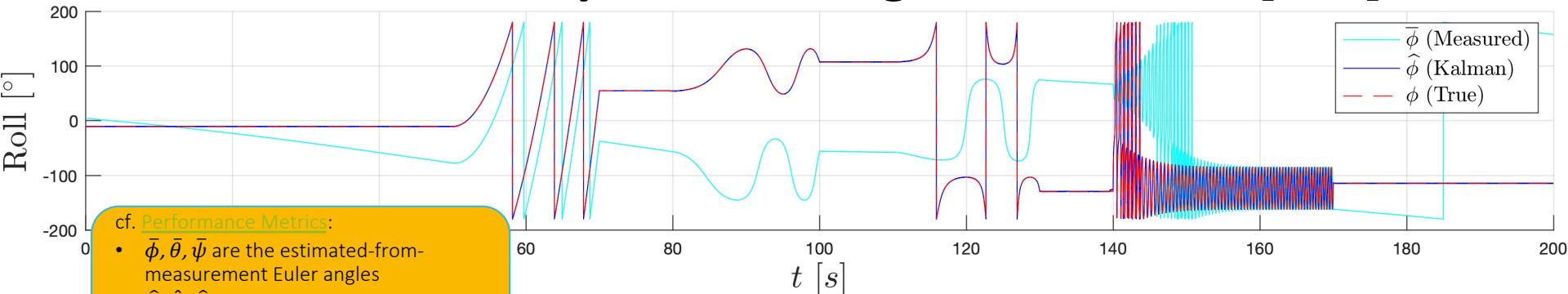
Major errors occur during external acceleration spikes (80 s, 120 s, and 140 s) and angular velocity discontinuity (140 s)



Simulated Data Analysis

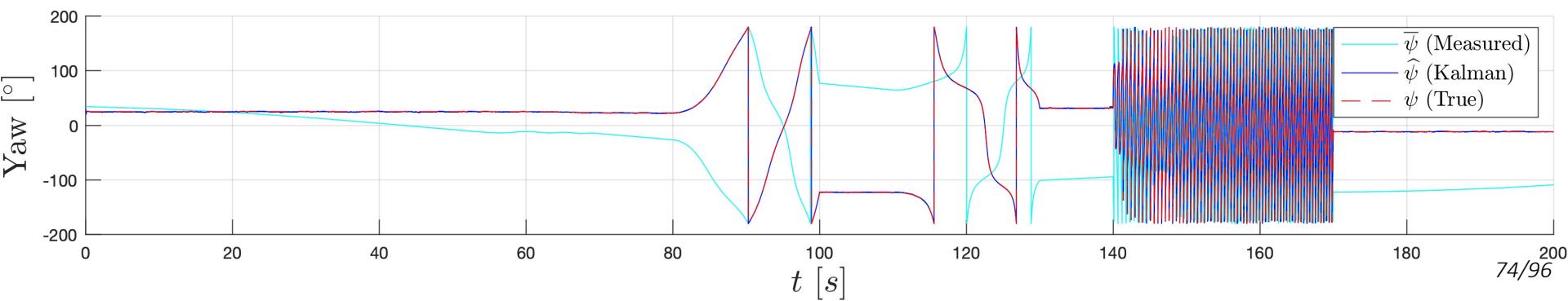
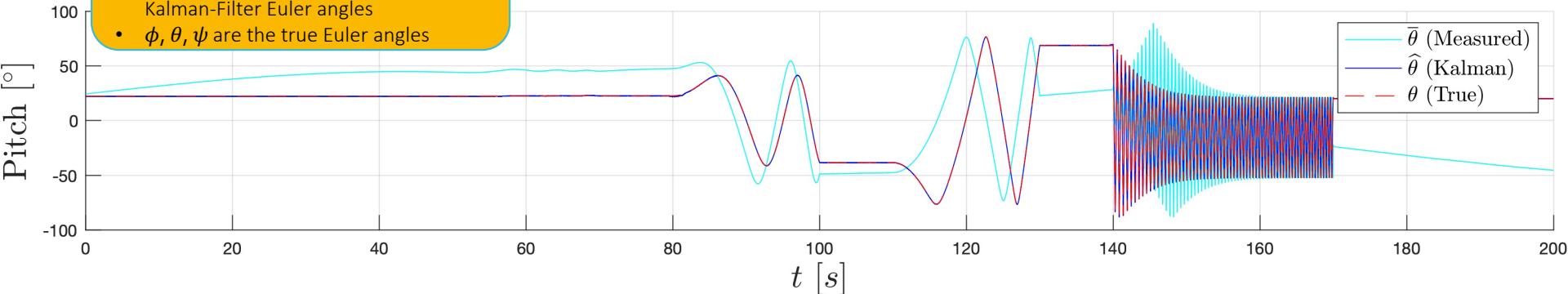
A More Extreme Example: Euler Angles Estimation [Suh]

The implemented method — along with Suh's algorithm — still works fine



cf. [Performance Metrics](#):

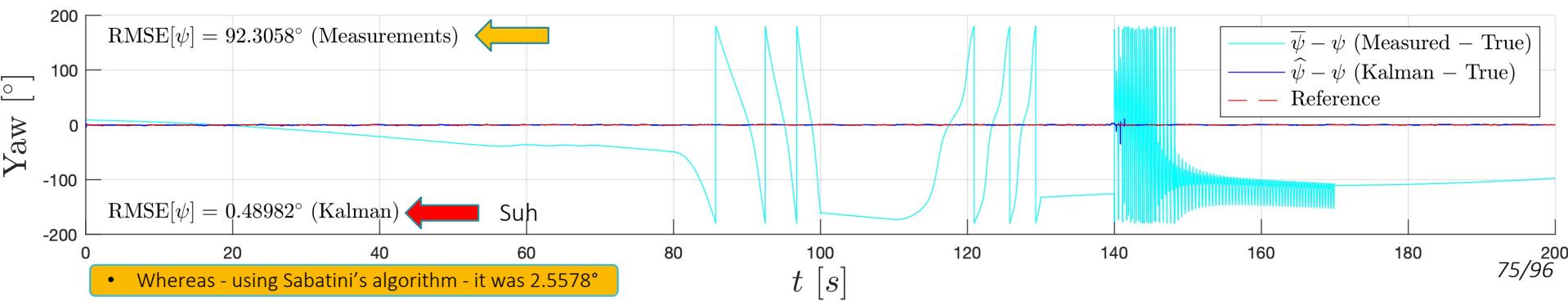
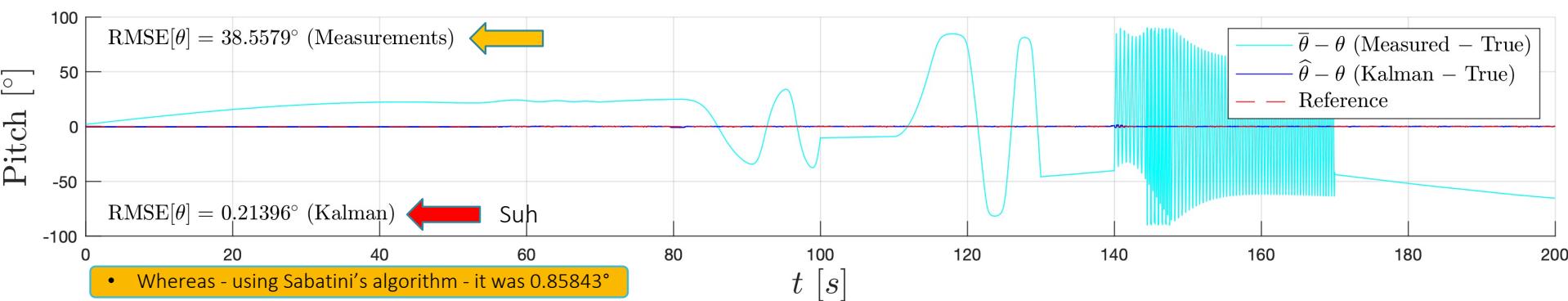
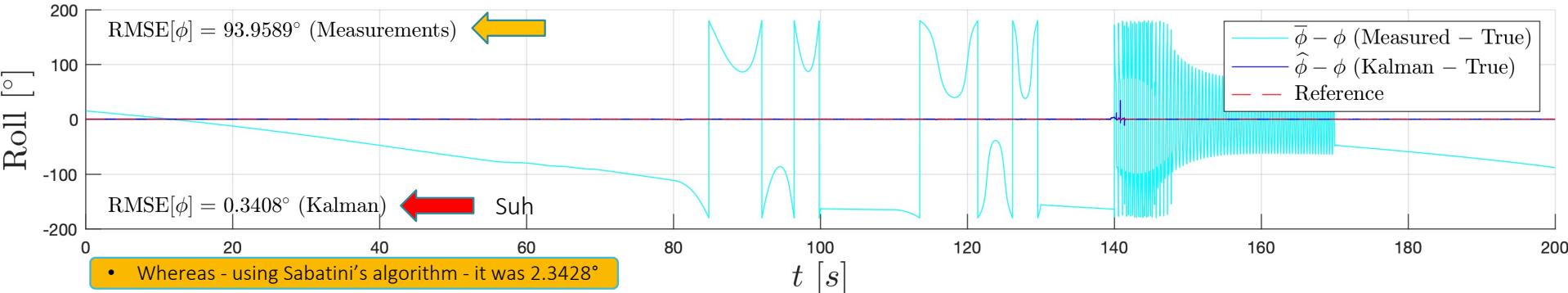
- $\bar{\phi}, \bar{\theta}, \bar{\psi}$ are the estimated-from-measurement Euler angles
- $\hat{\phi}, \hat{\theta}, \hat{\psi}$ are the estimated-from-Kalman-Filter Euler angles
- ϕ, θ, ψ are the true Euler angles



Simulated Data Analysis

A More Extreme Example: Euler Angles Estim. Error [Suh] (1/2)

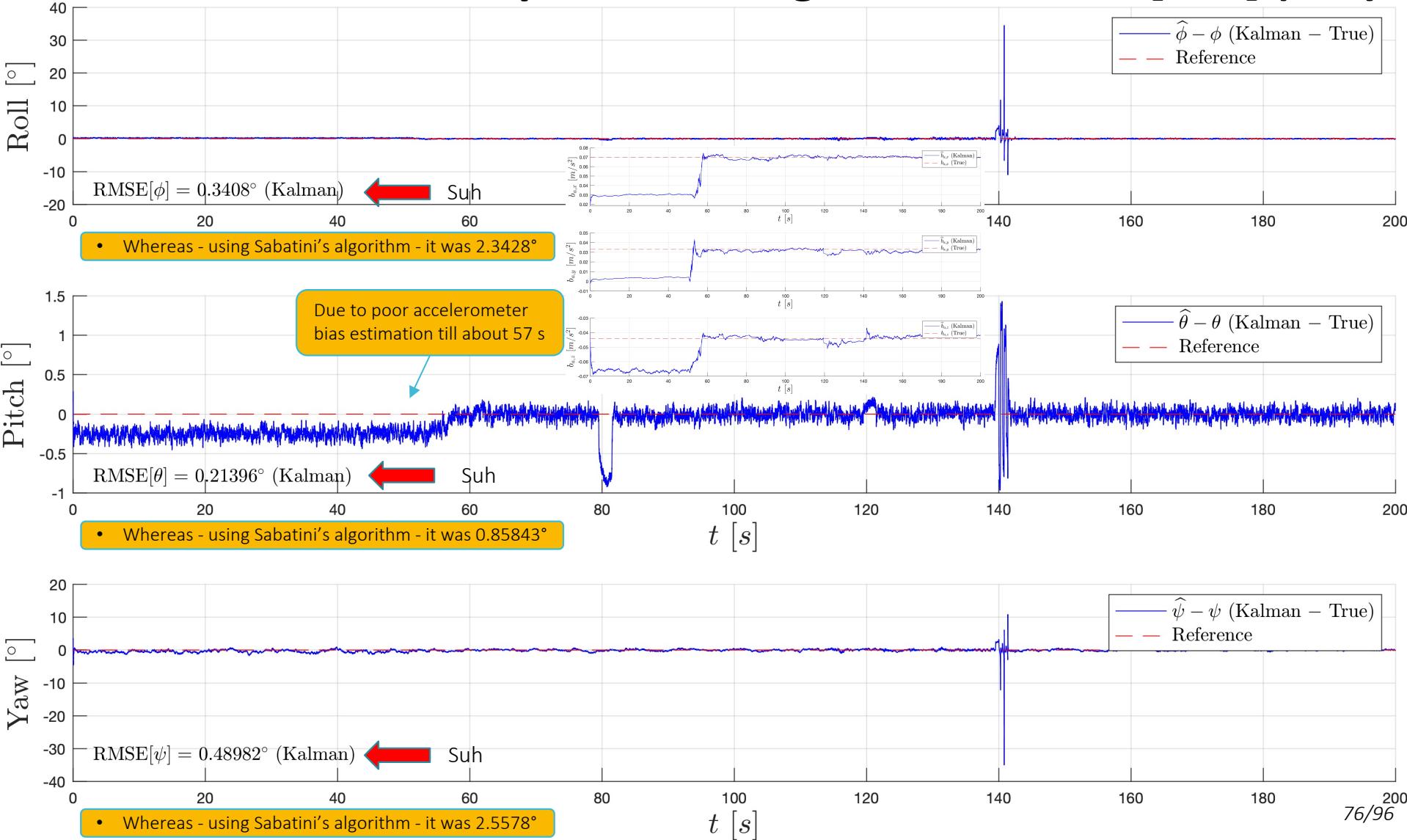
Suh's algorithm is still better than Sabatini's algorithm, as lower RMSEs can prove



Simulated Data Analysis

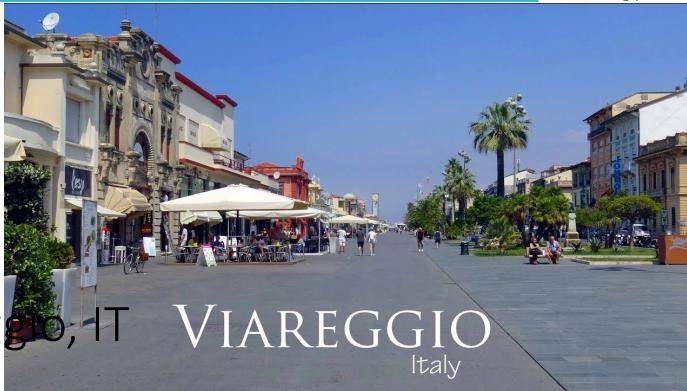
A More Extreme Example: Euler Angles Estim. Error [Suh] (2/2)

However, major errors still occur during external acceleration spikes (at 80 s, 120 s, and 140 s) and angular velocity discontinuity (at 140 s)



Real Data

Time Params & Sensor Characteristics



- Flight date: February 13th, 2020
- Flight coordinates: 43°52'52"N, 10°14'6"E, Viareggio, IT
- Flight duration: $t_f = 1625.640015$ s (about 27 minutes)
- Sampling frequency: $f_s = 50$ Hz → Sampling time: $T_s = 1/f_s = 0.02$ s
- Number of measurements: $N_{measurements} = 80041$
- N. of samples after interpolation: $N = \left\lfloor t_f / T_s \right\rfloor = 81242$
- Gravitational field parameter: $g = 9.805185$ m/s²
- Magnetic field parameters: $\alpha = 60.10803^\circ$, $m = 47.1179$ µT
- Values of biases and standard deviations:

- Sampling frequency and sampling time are assumed to be constant, and the same for all the sensors.
- Missing samples are interpolated (using Matlab® function *interp1* with *nearest* interpolation method).

Viareggio,
February 13th, 2020

	Bias	Standard deviation
Gyroscope [rad/s]	$b_g = [-0.0640 \quad -0.0240 \quad 0.0205]'$	$\sigma_g = 0.001$
Accelerometer [m/s ²]	$b_a = [0.5875 \quad -0.0000036 \quad 0.0312]'$	$\sigma_a = 0.039$
Magnetometer [µT]	$b_m = [0 \quad 0 \quad 0]'$	$\sigma_m = 2$

Real Data

Initial State Vector and State Covariance Matrix

- Initial state vector:

$$x_0 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]'$$

- Initial state covariance matrix:

$$P_0 = \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \\ 0_{3 \times 3} & 0.0005 & 0 \\ 0 & 0 & 0.0005 \\ 0 & 0 & 0.0005 \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0.05 \end{bmatrix}$$

Real Data

Norm-Based and Adaptive Alg. Params & Process Noise Cov. Matrix

- Norm-based algorithm parameters:

$$\varepsilon_a = 0.25 \text{ m/s}^2$$

$$s = 10$$

- Adaptive estimation of external acceleration algorithm parameters:

$$M_1 = 3$$

$$M_2 = 2$$

$$\gamma = 0.1$$

- Continuous-time process noise covariance matrix:

Q_{bg} is the covariance noise matrix of the gyroscope

$$Q_{bg} = \begin{bmatrix} 10^{-7} & 0 & 0 \\ 0 & 10^{-7} & 0 \\ 0 & 0 & 10^{-7} \end{bmatrix}, \quad Q_{ba} = \begin{bmatrix} 10^{-5} & 0 & 0 \\ 0 & 10^{-5} & 0 \\ 0 & 0 & 10^{-5} \end{bmatrix}$$

Q_{ba} is the covariance noise matrix of the accelerometer

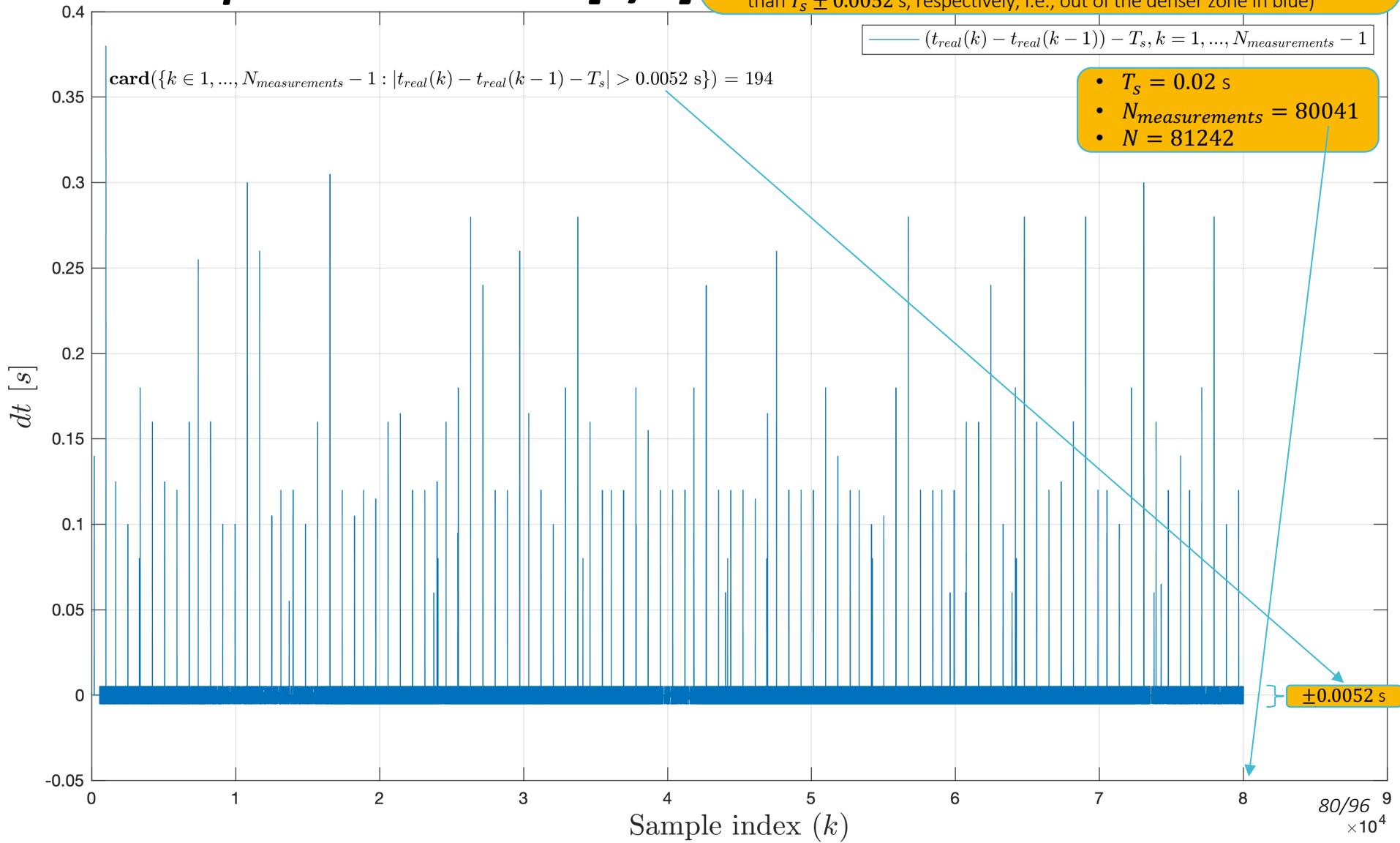
$R_g = \sigma_g^2 \cdot I_{3 \times 3}$ is the covariance measurement matrix of the gyroscope

Recall that: $Q = \begin{bmatrix} 0.25R_g & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & Q_{bg} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & Q_{ba} \end{bmatrix}$

Real Data Analysis

Time Step Discontinuities [1/2]

- For simplicity, sampling time T_s was assumed to be constant
- Unfortunately, there are (usually) small temporal discontinuities between two consecutive samples, i.e., $\exists k \in \{1, \dots, N_{\text{measurements}} - 1\} : t_{\text{real}}(k) - t_{\text{real}}(k - 1) \neq T_s$
- Fortunately, only a minority of them (194) are non-negligible (lower/greater than $T_s \pm 0.0052$ s, respectively, i.e., out of the denser zone in blue)



Real Data Analysis

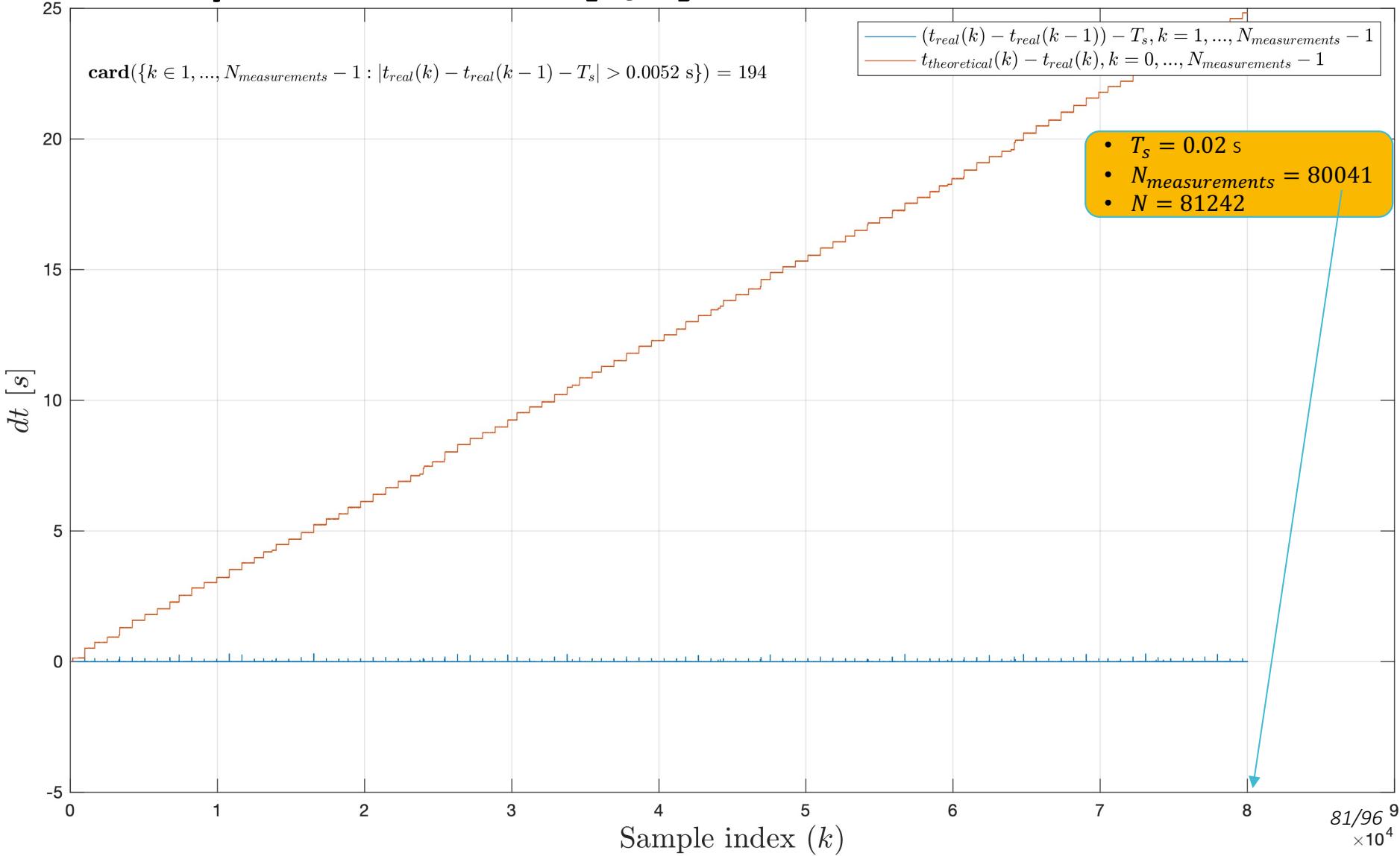
Time Step Discontinuities [2/2]

- Time step cumulative errors are plotted in red
- Periodicity in time step discontinuities is evident

$\text{card}(\{k \in 1, \dots, N_{\text{measurements}} - 1 : |t_{\text{real}}(k) - t_{\text{real}}(k - 1) - T_s| > 0.0052 \text{ s}\}) = 194$

— $(t_{\text{real}}(k) - t_{\text{real}}(k - 1)) - T_s, k = 1, \dots, N_{\text{measurements}} - 1$
— $t_{\text{theoretical}}(k) - t_{\text{real}}(k), k = 0, \dots, N_{\text{measurements}} - 1$

- $T_s = 0.02 \text{ s}$
- $N_{\text{measurements}} = 80041$
- $N = 81242$

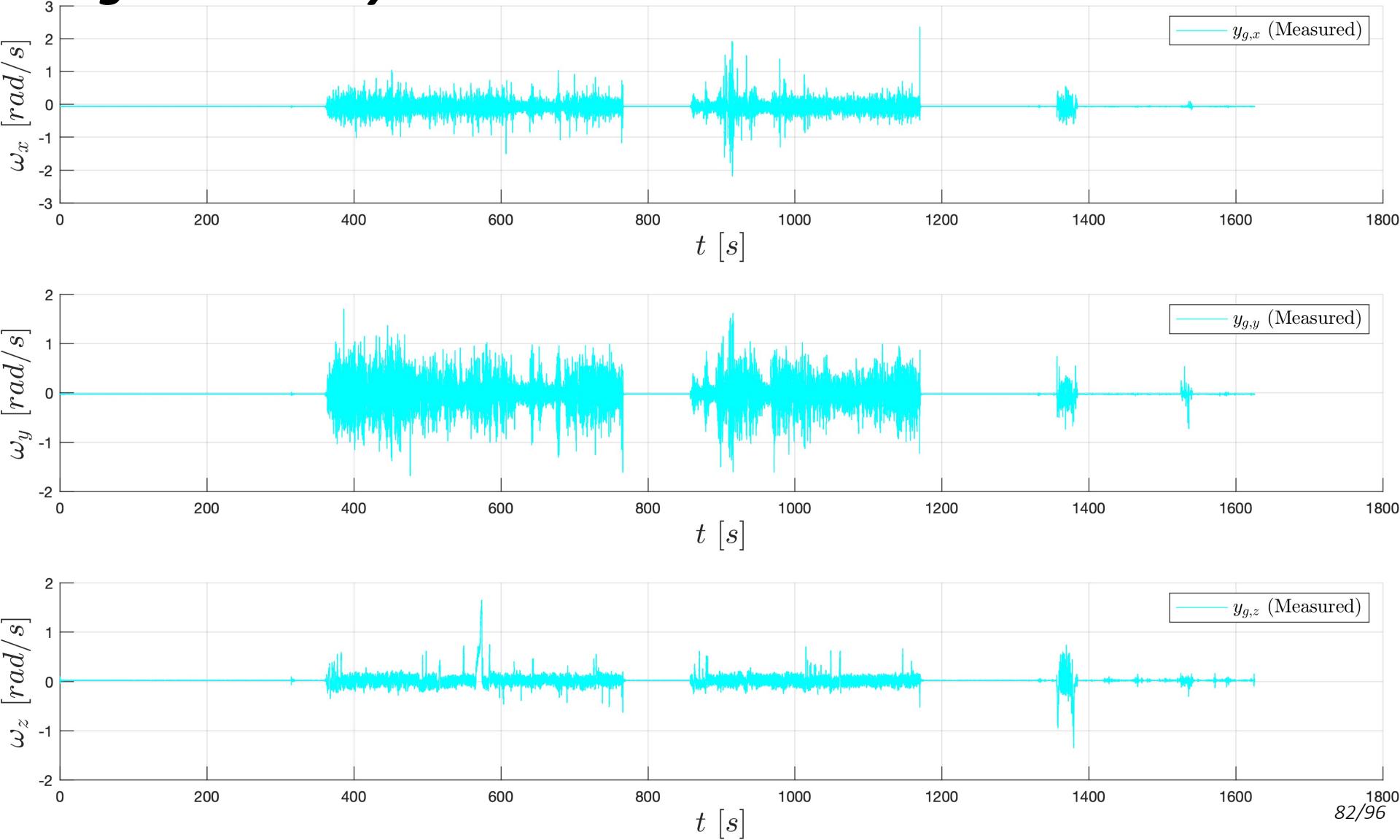


Real Data Analysis

Angular Velocity

$$\cdot \quad y_g = \omega + b_g + n_g$$

- Measured angular velocity y_g is affected by noise n_g and bias b_g

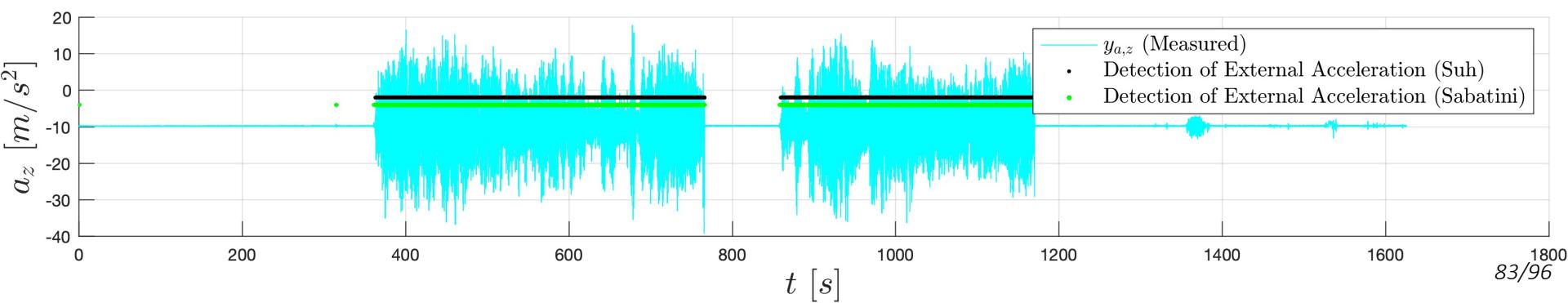
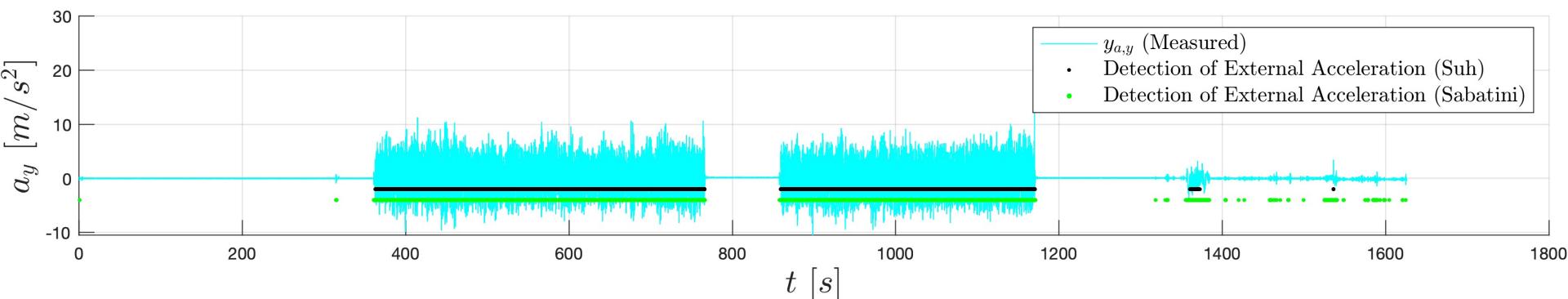
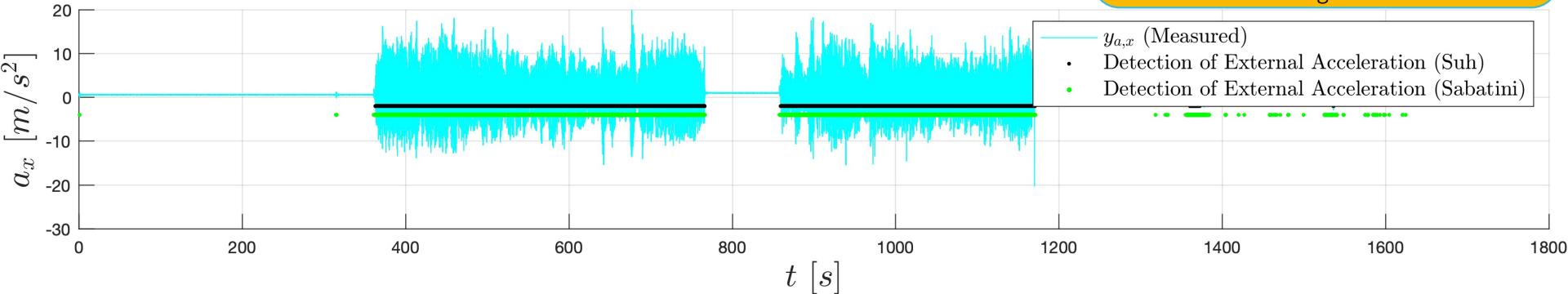


Real Data Analysis

Acceleration

$$\bullet \quad y_a = C(q)\bar{g} + b_a + n_a + a_b$$

- Measured acceleration y_a is affected by noise n_a and bias b_a
- Detection instants of external acceleration seem to be almost the same for both algorithms

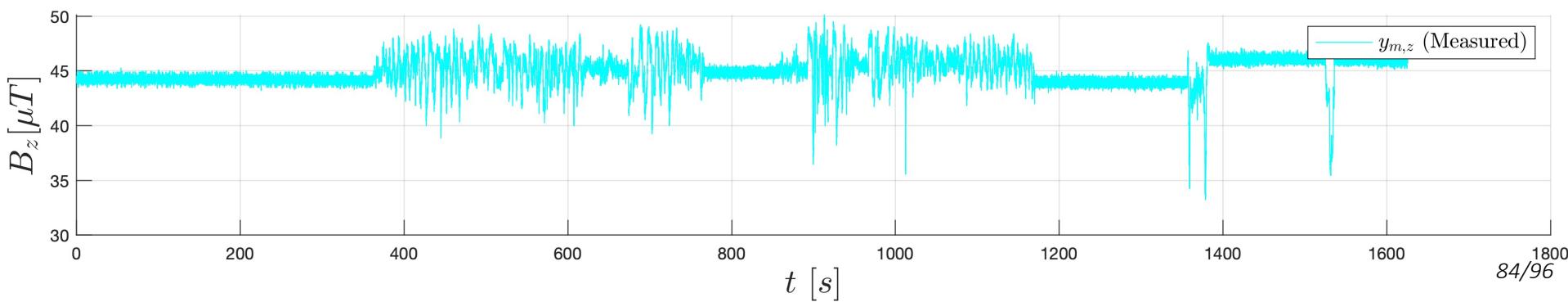
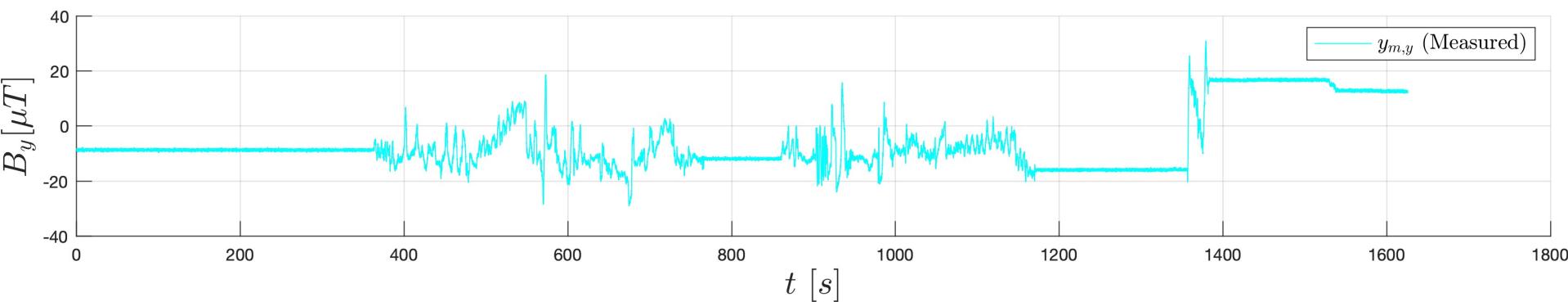
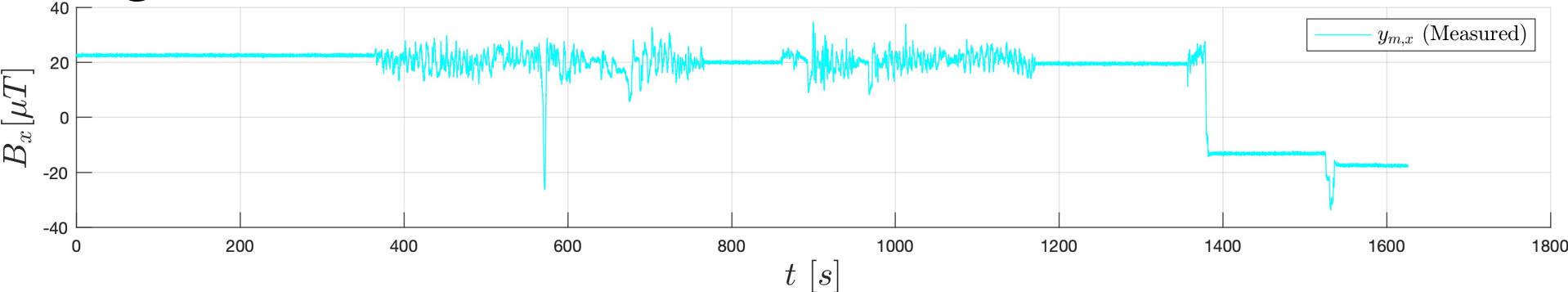


Real Data Analysis

Magnetic Field

- $y_m = C(q)\tilde{m} + n_m$

- Measured magnetic field y_m is highly affected by noise n_m
- Magnetic bias b_m was not considered in the model



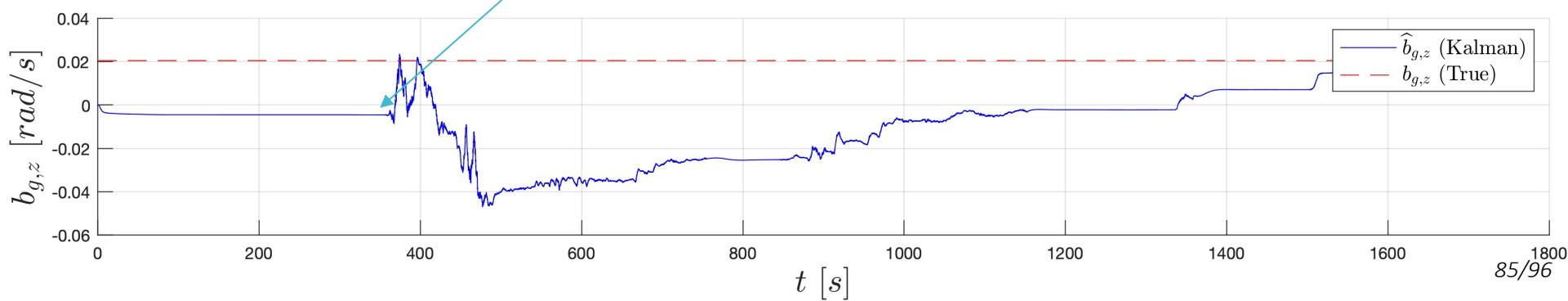
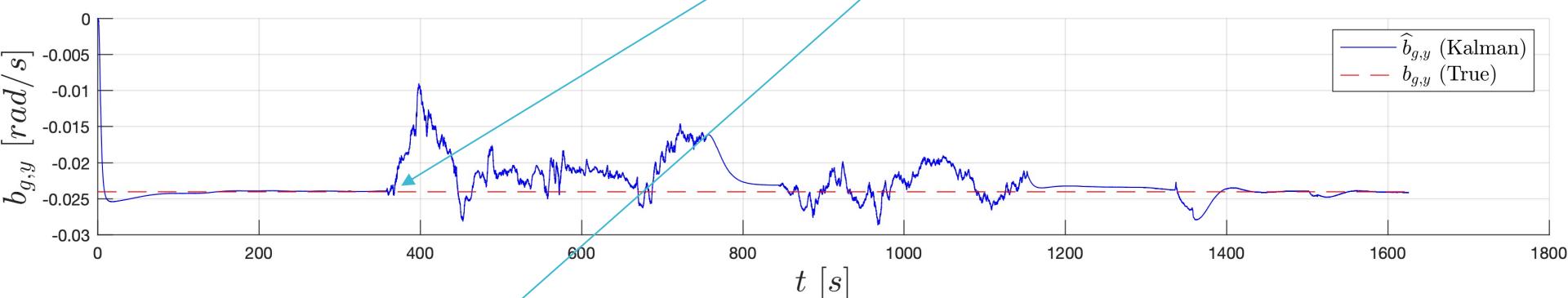
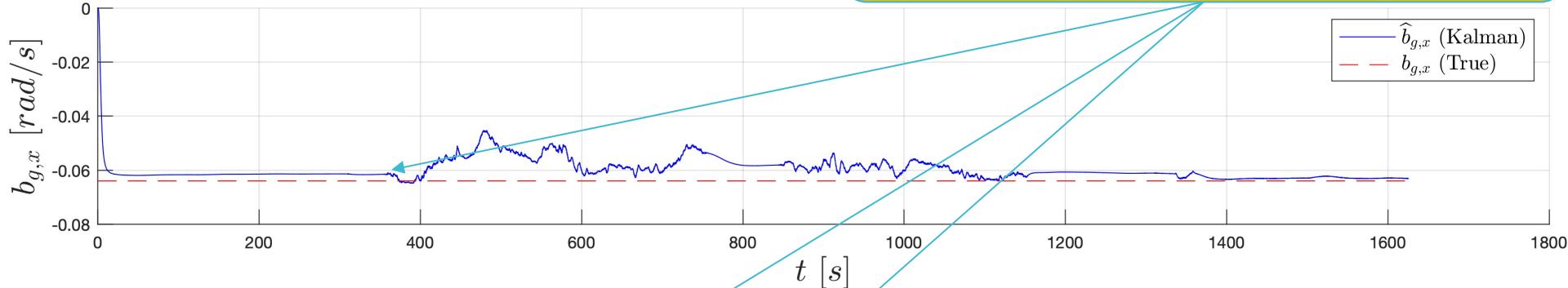
Real Data Analysis

Gyroscope Bias Estimation

$$\cdot \hat{b}_g = x(4:6)$$

$$\cdot b_g$$

- Gyro bias b_g — taken as reference — was obtained averaging angular velocity measurements during first quiet period (about 5 minutes long), when gyroscope output was almost static
- Gyro bias b_g (especially on the x and y axes) is estimated rather well by \hat{b}_g
- There is still not negligible inaccuracy during non-null gyro measurements (starting at about 360 s)

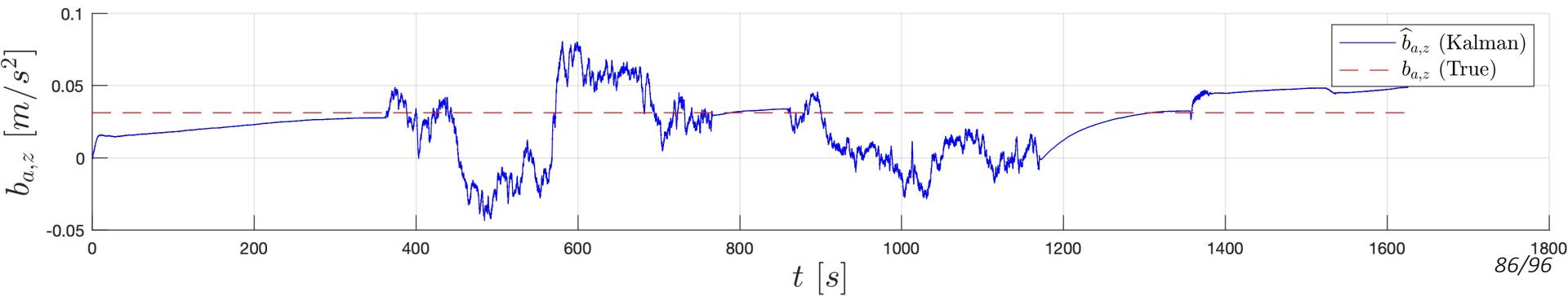
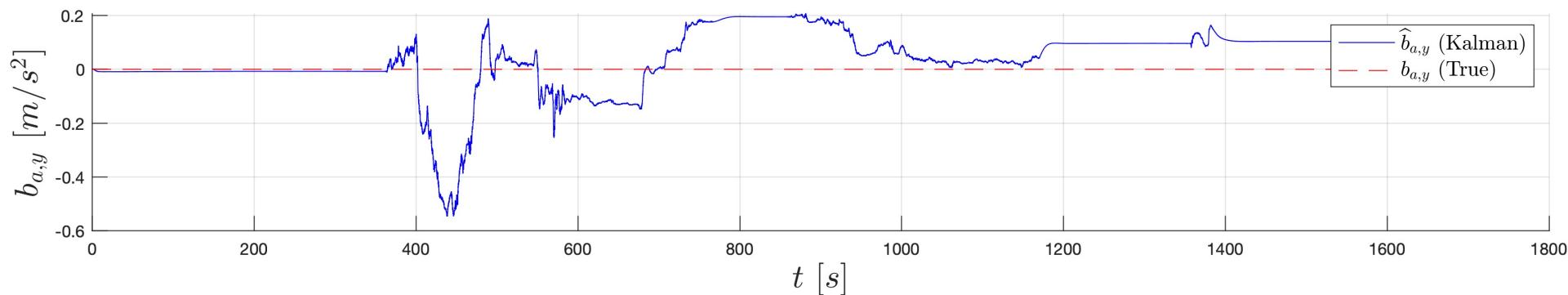
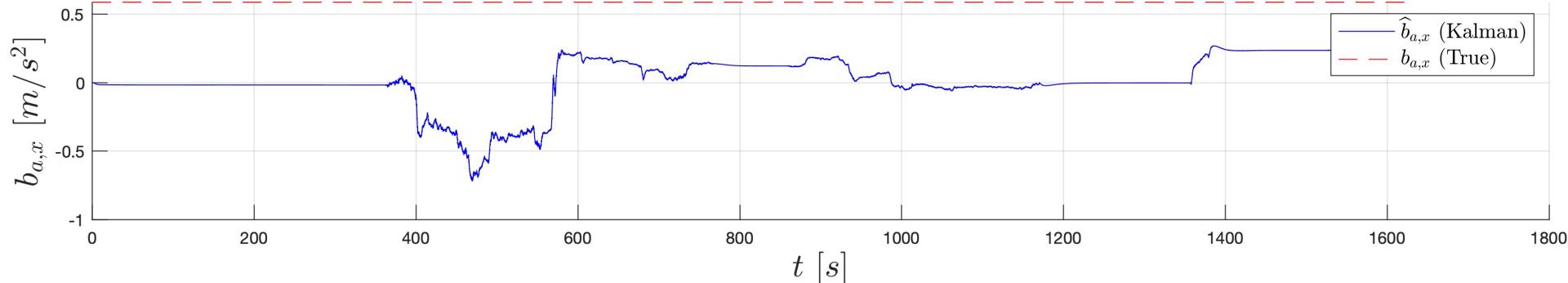


Real Data Analysis

Accelerometer Bias Estimation

$$\begin{aligned} \cdot \hat{b}_a &= x(7:9) \\ \cdot b_a \end{aligned}$$

- Accelerometer bias b_a — taken as reference — was obtained averaging acceleration measurements during first quiet period (about 5 minutes long), when accelerometer output was almost static
- Accelerometer bias b_a is not estimated too well by \hat{b}_a : it cannot be excluded that real bias differs a lot from the bias taken as reference; also, it may not be constant

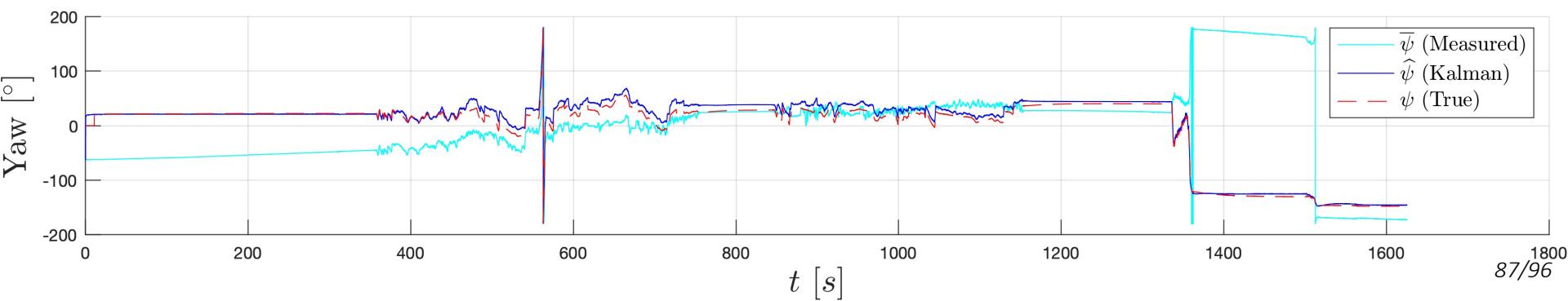
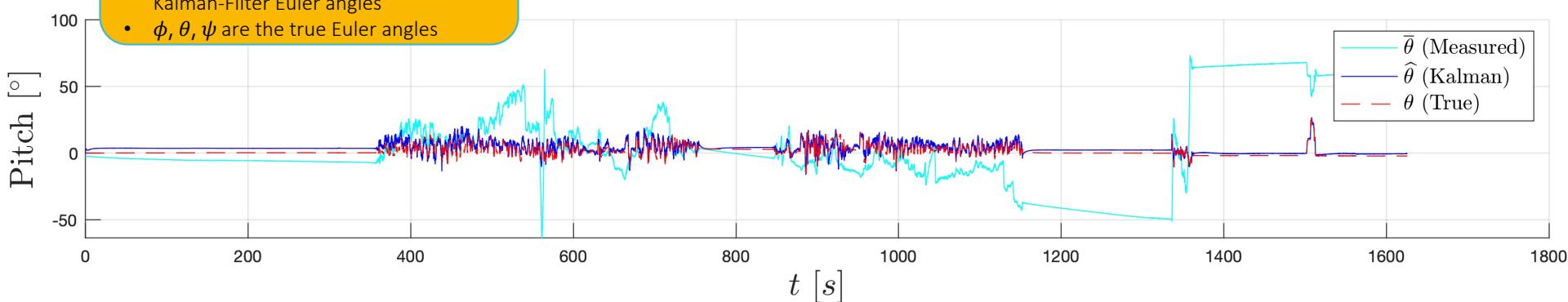
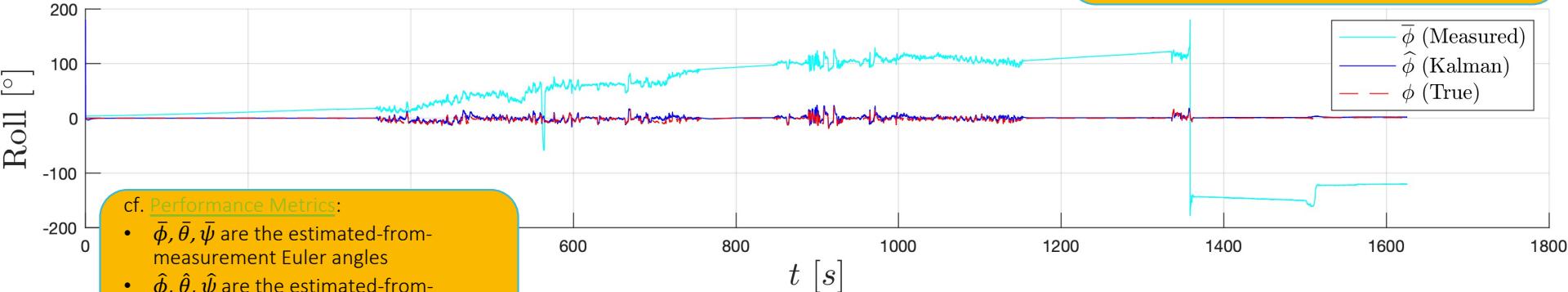


Real Data Analysis

Euler Angles Estimation [Sabatini] [1/2]

Note: gyroscope bias b_a was removed from gyro measurements y_g before integrating the latter and obtaining estimated-from-measurement Euler angles $\bar{\phi}, \bar{\theta}, \bar{\psi}$

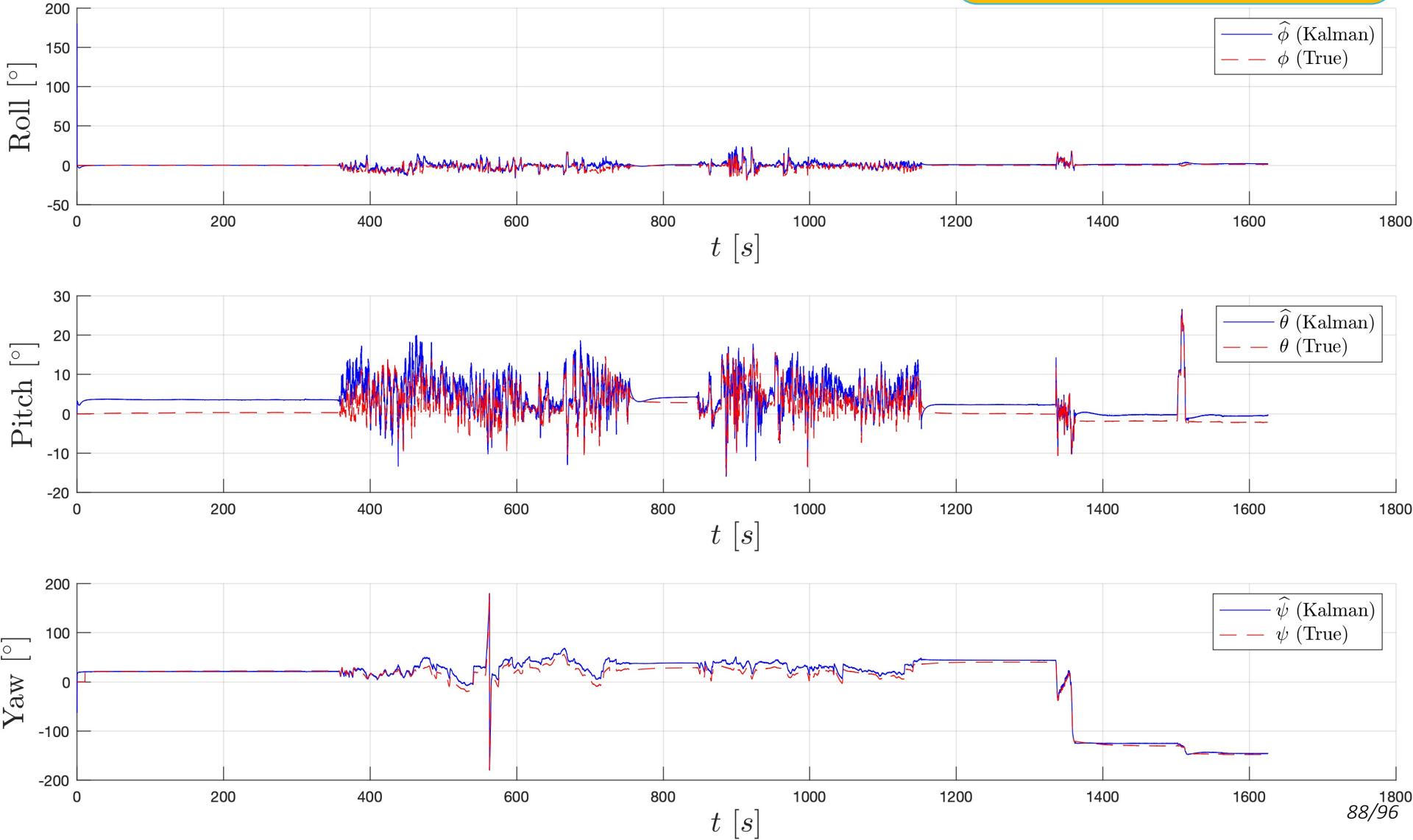
- Euler angles estimated using the proposed method along with Sabatini's algorithm are way more accurate than Euler angles estimated using just angular velocity measurements ...



Real Data Analysis

Euler Angles Estimation [Sabatini] [2/2]

- ... there are still evident estimation errors, though



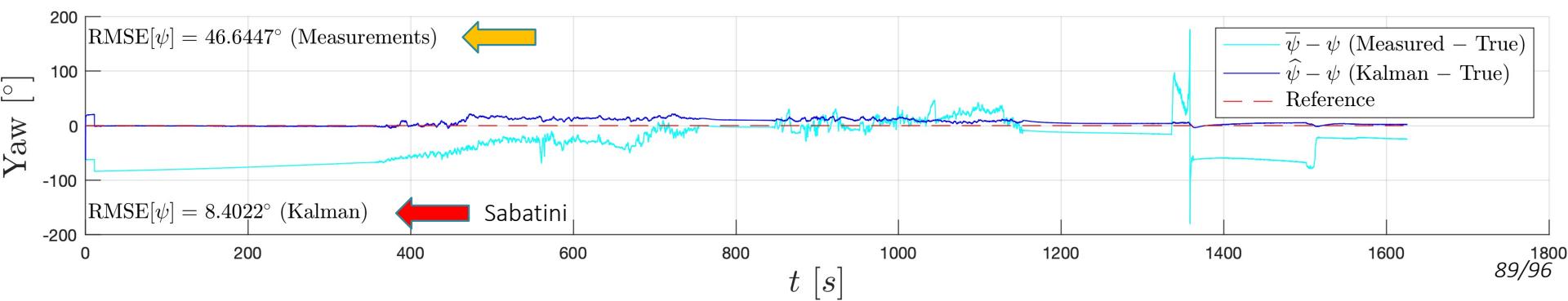
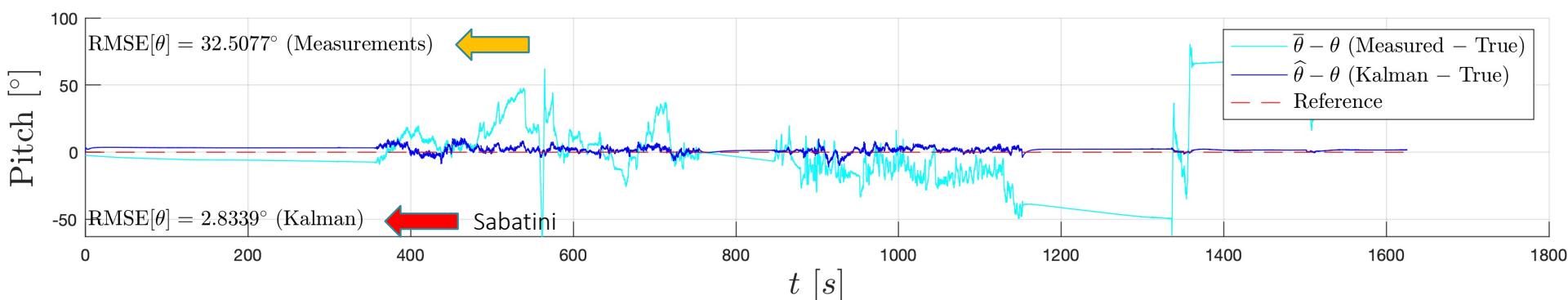
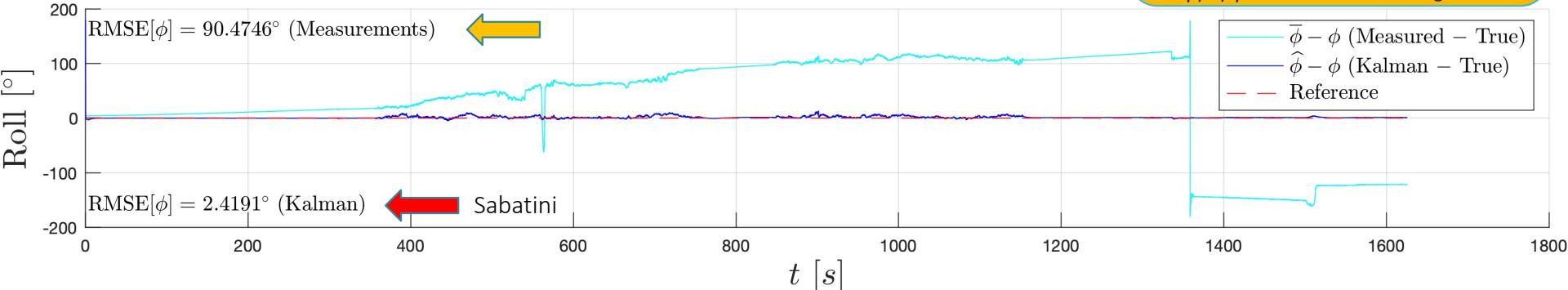
Real Data Analysis

Euler Angles Estimation Error [Sabatini] [1/2]

Note: gyroscope bias b_a was removed from gyro measurements y_g before integrating the latter and obtaining estimated-from-measurement Euler angles $\bar{\phi}, \bar{\theta}, \bar{\psi}$

cf. [Performance Metrics](#):

- $\bar{\phi}, \bar{\theta}, \bar{\psi}$ are the estimated-from-measurement Euler angles
- $\hat{\phi}, \hat{\theta}, \hat{\psi}$ are the estimated-from-Kalman-Filter Euler angles
- ϕ, θ, ψ are the true Euler angles

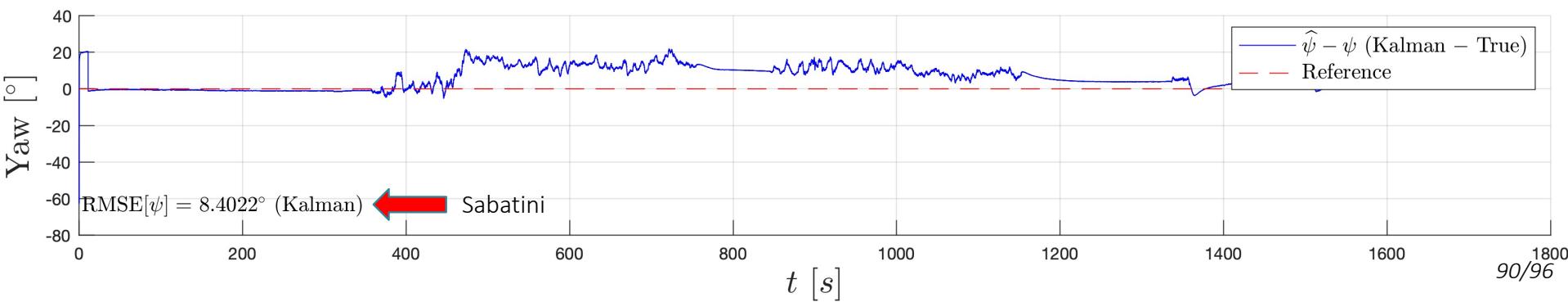
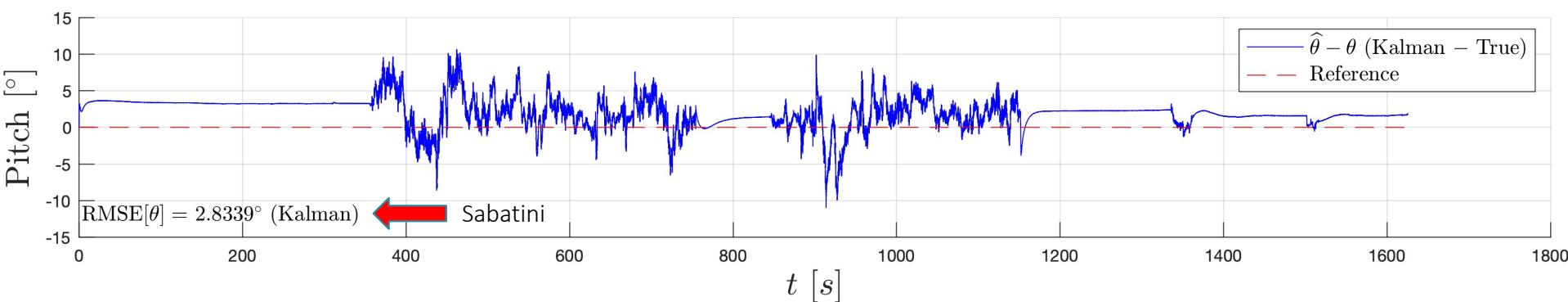
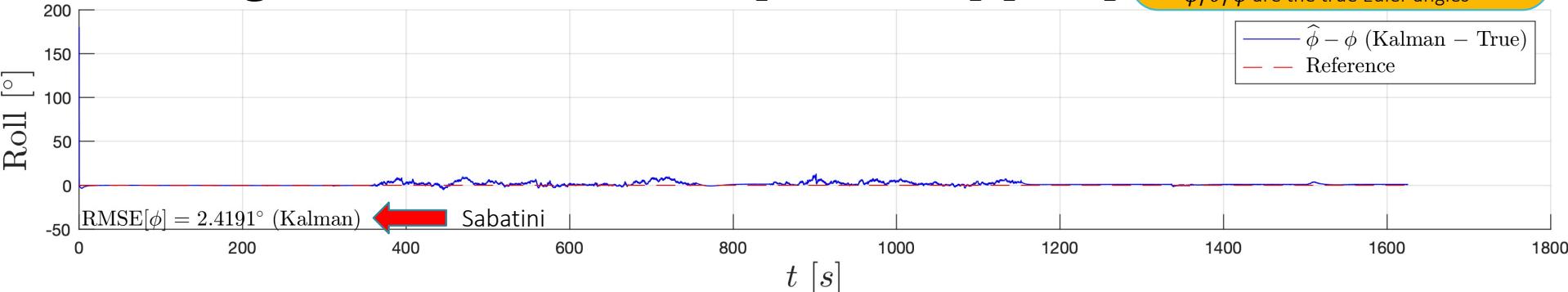


Real Data Analysis

Euler Angles Estimation Error [Sabatini] [2/2]

cf. [Performance Metrics](#):

- $\bar{\phi}, \bar{\theta}, \bar{\psi}$ are the estimated-from-measurement Euler angles
- $\hat{\phi}, \hat{\theta}, \hat{\psi}$ are the estimated-from-Kalman-Filter Euler angles
- ϕ, θ, ψ are the true Euler angles

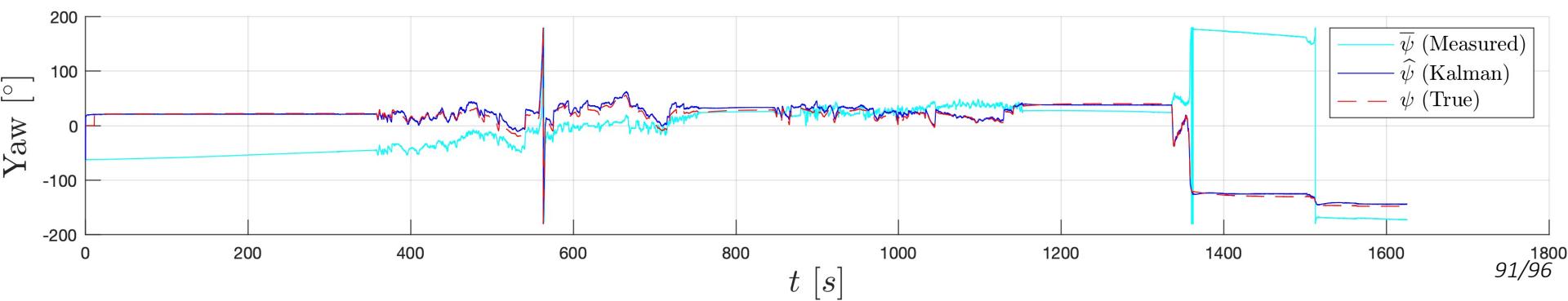
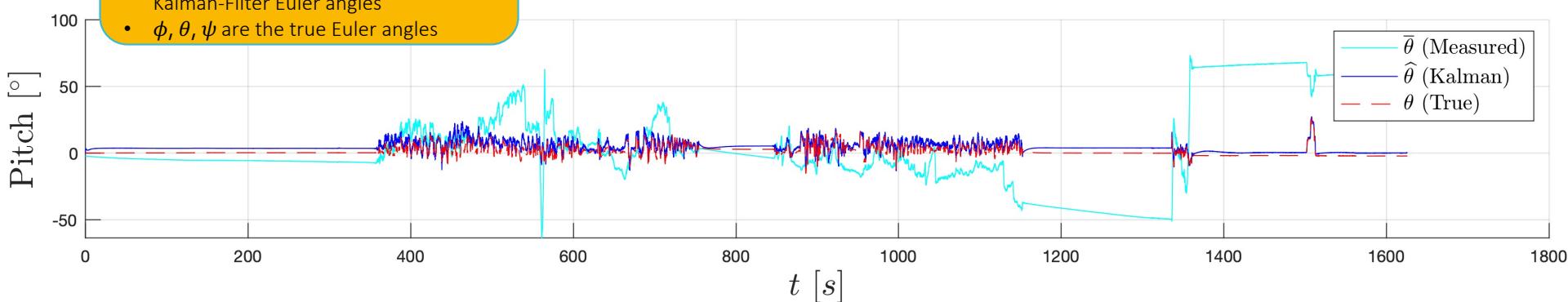
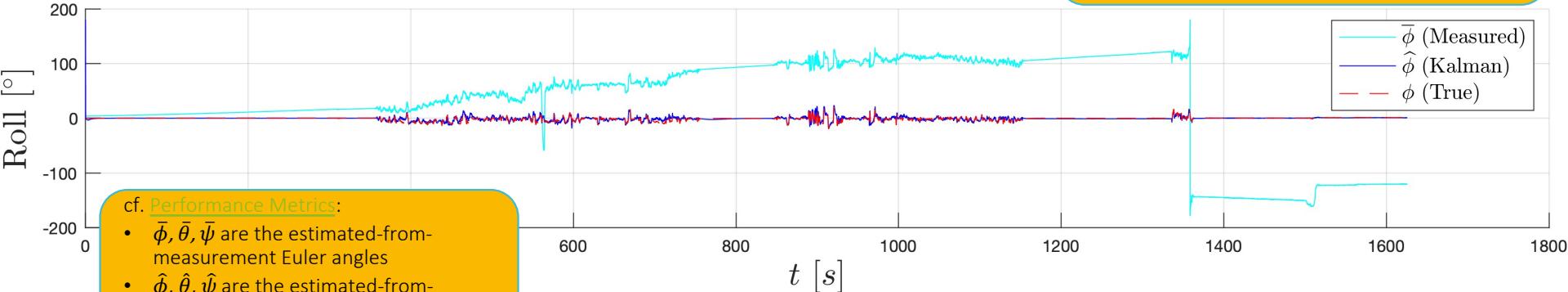


Real Data Analysis

Euler Angles Estimation [Suh] [1/2]

Note: gyroscope bias b_a was removed from gyro measurements y_g before integrating the latter and obtaining estimated-from-measurement Euler angles $\bar{\phi}, \bar{\theta}, \bar{\psi}$

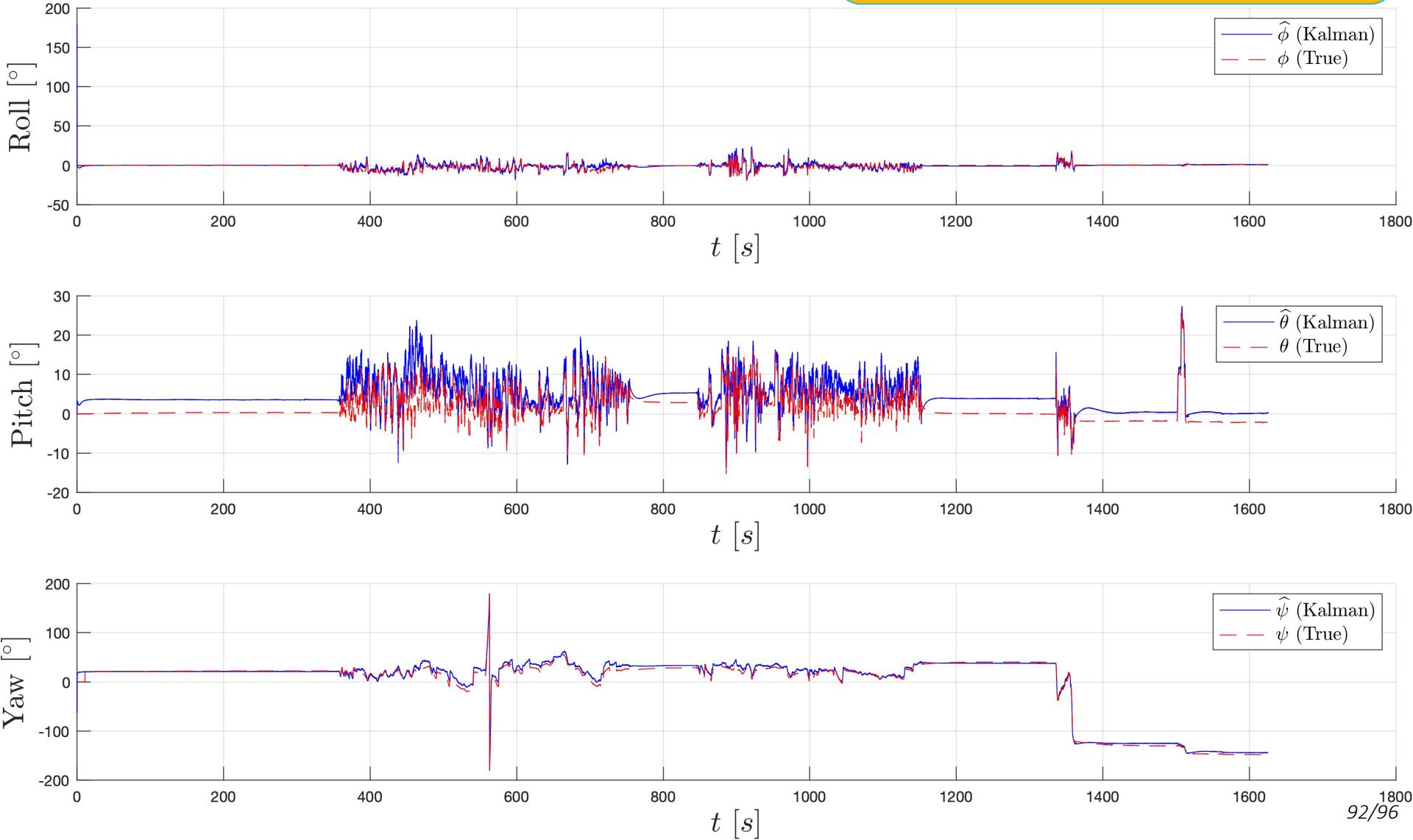
- The proposed method, along with Suh's algorithm, seems to behave slightly better than Sabatini's algorithm...



Real Data Analysis

Euler Angles Estimation [Suh] [2/2]

- Unfortunately, there's a bias of about 4° on $\hat{\theta}$
- It was present also when using Sabatini's algorithm

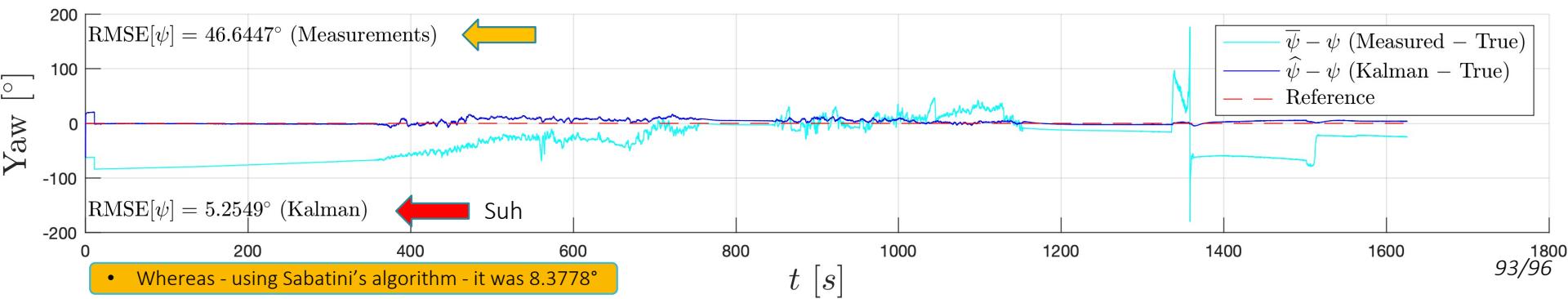
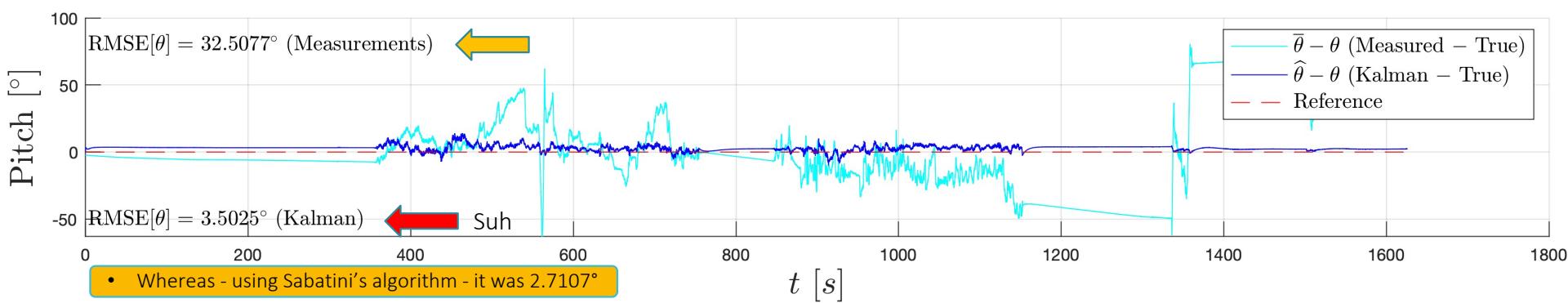
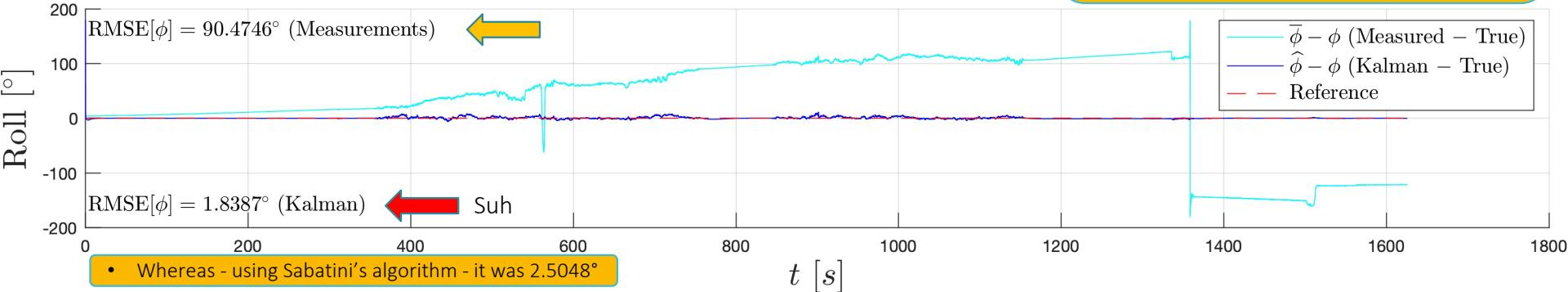


Real Data Analysis

Euler Angles Estimation Error [Suh] [1/2]

Note: gyroscope bias b_a was removed from gyro measurements y_g before integrating the latter and obtaining estimated-from-measurement Euler angles $\bar{\phi}, \bar{\theta}, \bar{\psi}$

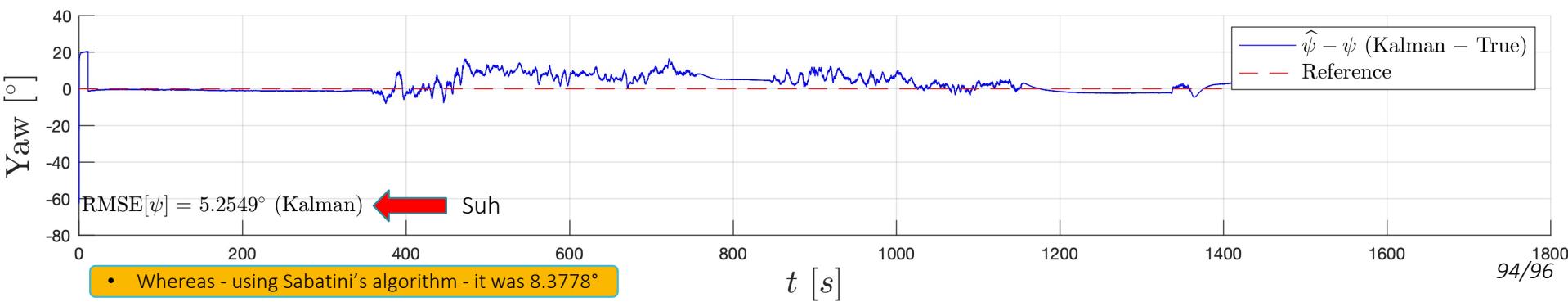
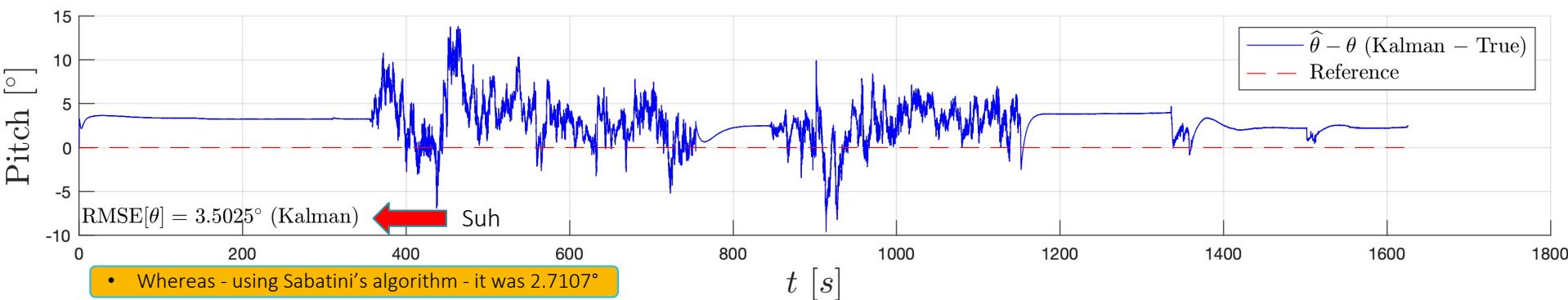
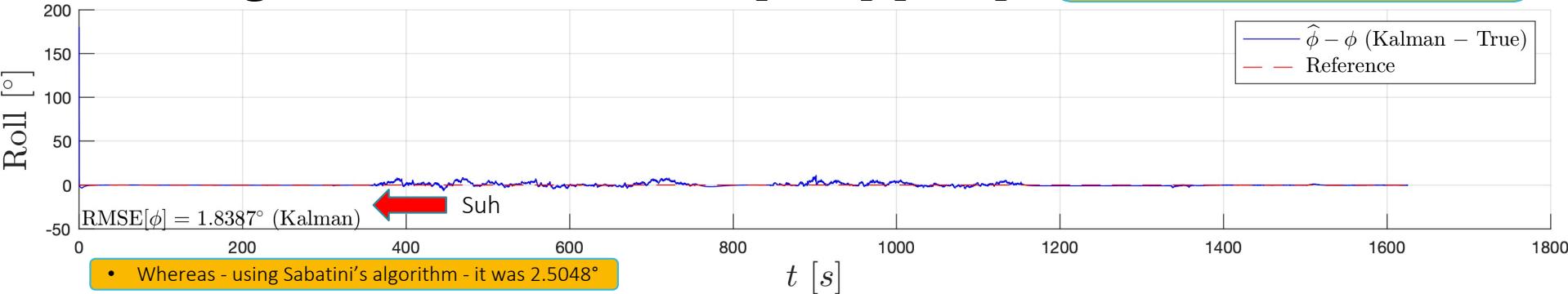
- The proposed method, along with Suh's algorithm, seems to behave slightly better than Sabatini's algorithm...
- ... as lower RMSEs can prove



Real Data Analysis

Euler Angles Estimation Error [Suh] [2/2]

- It is noteworthy that improvements using Suh's algorithm — compared to using Sabatini's algorithm — are lower with real data than with simulated data



Conclusions

1. The implemented method limits the effect of the correction made during the magnetic sensor measurement update only to the yaw component: this prevents magnetic disturbance — which sometimes could be very large — from affecting also pitch and roll components.
2. Nevertheless, this method does not take into account magnetometer bias, which should be considered in future works.
3. Step-by-step estimate of the external acceleration covariance matrix $\hat{Q}_{a_b,k}$ using Sabatini's norm-based algorithm seems to react promptly to external accelerations disturbances.
4. Nevertheless, even better results can be achieved using Suh's approach. In fact, it considers differences along the three spatial directions of the external accelerations — giving different weights to different axes of the accelerometer output — which ensures even more precise results.
5. Gyroscope and accelerometer biases in simulated data are identified sufficiently fast and with a high degree of attainability, proving the method efficacy.

Conclusions

6. Worse bias prediction behavior with real data can be explained mainly by not knowing precisely real (true) bias values (they are estimated by averaging first 5 minutes of measurements, when sensor outputs are almost static), by unlikely being constant over time, and by highly noisy disturbances affecting measurements.
7. Overall, the implemented filter can precisely estimate the vehicle attitude with simulated data, as very low RMSEs — even under extreme situations — can prove.
8. With real data the filter seems less precise, but can still achieve useful results, estimating orientation much better than without the filter.
9. Moreover, Suh's algorithm can achieve lower RMSEs than Sabatini's algorithm, even though the difference is less pronounced compared to using simulated data.
10. Ultimately, it is worthwhile to mention that Euler angles estimated by the implemented Kalman Filter may even be more accurate than those taken as reference (and previously estimated using another Kalman Filter).

A large, vibrant green banana leaf with a prominent central vein is floating on the surface of clear, turquoise-blue ocean water. The water is filled with fine, radiating ripples and light reflections, creating a sense of depth and tranquility.

- Thanks a bunch for your attention