Moving Horizon Estimation for GNSS-IMU sensor fusion

Guido Sánchez
sinc(i)
UNL-CONICET
Santa Fe, Argentina
gsanchez@sinc.unl.edu.ar

Marina Murillo
sinc(i)
UNL-CONICET
Santa Fe, Argentina
mmurillo@sinc.unl.edu.ar

Lucas Genzelis
sinc(i)
UNL-CONICET
Santa Fe, Argentina
lgenzelis@sinc.unl.edu.ar

Nahuel Deniz
sinc(i)
UNL-CONICET
Santa Fe, Argentina
ndeniz@sinc.unl.edu.ar

Leonardo Giovanini
sinc(i)
UNL-CONICET
Santa Fe, Argentina
lgiovanini@sinc.unl.edu.ar

sinc(i) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
G. Sanchez, M. Murillo, L. Genzelis, Nestor N. Deniz & L. Giovanini; "Moving horizon estimation for GNSS-IMU sensor fusion."
IX Jornadas Argentinas de Robótica (JAR), 2017.

***Abstract*—The aim of this work is to develop a Global Navigation Satellite System (GNSS) and Inertial Measurement Unit (IMU) sensor fusion system. To achieve this objective, we introduce a Moving Horizon Estimation (MHE) algorithm to estimate the position, velocity orientation and also the accelerometer and gyroscope bias of a simulated unmanned ground vehicle. The obtained results are compared with the true values of the system and with an Extended Kalman filter (EKF). The use of CasADi and Ipopt provide efficient numerical solvers that can obtain fast solutions. The quality of MHE estimated values enable us to consider MHE as a viable replacement for the popular Kalman Filter, even on real time systems.**

***Index Terms*—State Estimation, Sensor Fusion, Moving Horizon Estimation, GNSS, IMU.**

I. Introduction

Navigation aims to solve the problem of determining the position, velocity and orientation of an object in space using different sources of information. If we want to control efficiently an unmanned vehicle, its position, velocity and orientation should be known as accurately as possible. The integration of Global Navigation Satellite Systems (GNSS) and Inertial Measurement Units (IMU) is the state of the art among navigation systems [1], [2]. It involves non-linear measurement equations combined with rotation matrices, expressed through Euler angles or quaternions, along with the cinematic models for the rigid body's translation and rotation in space. Traditionally, the Extended Kalman Filter (EKF) [3]–[5], Unscented Kalman Filter (UKF) [6], [7] or the Particle Filter (PF) [8], [9] are used to solve the navigation problem.

Recently, the use of non-linear observers have been proposed as an alternative to the different types of Kalman filters and statistical methods. However, there is still little literature on the subject. Grip et al. [10] present an observer for estimating position, velocity, attitude, and gyro bias, by using inertial measurements of accelerations and angular velocities, magnetometer measurements, and satellite-based measurements of position and (optionally) velocity. Vandersteen et al. [2] use a Moving Horizon Estimation (MHE) algorithm in real-time to estimate the orientation and the sensor calibration parameters applied to two space mission scenarios. In the first scenario, the attitude is estimated from three-axis magnetometer and gyroscope measurements. In the second scenario, a star tracker is used to jointly estimate the attitude and gyroscope calibration parameters. In order to solve this constrained optimization problem in real time, an efficient numerical solution method based on the iterative Gauss–Newton scheme has been implemented and specific measures are taken to speed up the calculations by exploiting the sparsity and band structure of matrices to be inverted. In Poloni et al. [1] a nonlinear numerical observer for accurate position, velocity and attitude estimation including the accelerometer bias and gyro bias estimation is presented. A Moving Horizon Observer (MHO) processes the accelerometer, gyroscope and magnetometer measurements from the IMU and the position and velocity measurements from the GNSS. The MHO is tested off-line in the numerical experiment involving the experimental flight data from a light fixed-wing aircraft.

Both EKF and MHE are based on the solution of a least-squares problem. While EKF use recursive updates to obtain the estimates and the error covariance matrix, MHE use a finite horizon window and solve a constrained optimization problem to find the estimates. In this way, the physical limits of the system states and parameters can be modeled through the optimization problem's constraints. The omission of this information can degrade the estimation algorithm performance [11]. Unfortunately, the Kalman based filters do not explicitly incorporate restrictions in the estimates (states and/or parameters) and, because of this, several ad-hoc methods have been developed [12]–[17]. These methods lead to sub-optimal solutions at best and can obtain non-realistic solutions under certain conditions, specially when the statistics of the unknown variables are chosen poorly. On the other hand, MHE solves an optimization problem to find the system estimates on each sample step, providing a theoretical framework for theoretic frame for constrained state and parameter estimation.

In this work it will be assumed that the reader is familiar with some of the many coordinate frames used for navigation. If needed, the work of Bekir [18] provides an excellent introduction to these topics. In particular, these coordinate frames will be used:

1) Body reference frame, referred as Body and by the superindex b.
2) Earth-Centered Earth-Fixed, referred as ECEF and by the superindex e.

3) East-North-Up, referred as ENU and by the superindex n.

This work is organized as follows: in Section II the problem formulation is presented. Section III describes the aspects of the Moving Horizon Estimation algorithm and the Extended Kalman Filter implementation. In order to compare the proposed method, a test simulation example is given in Section IV. Finally, in Section V conclusions of this work are stated.

II. Problem formulation

The system equations (for a detailed description, see Poloni et al. [1] and Bekir [18]) that describe the rigid body dynamics in ECEF coordinates are given by:

$$\dot{p}^e = v^e \tag{1}$$
$$\dot{v}^e = -2S(\omega^e_{ie})v^e + a^e + g^e(p^e) \tag{2}$$
$$\dot{q}^e_b = \frac{1}{2}q^e_b \cdot \tilde{\omega}^b_{ib} - \frac{1}{2}\tilde{\omega}^e_{ie} \cdot q^e_b \tag{3}$$
$$\dot{\alpha} = 0 \tag{4}$$
$$\dot{\beta} = 0 \tag{5}$$

where p^e is the position in ECEF coordinates, v^e is the linear velocity in ECEF coordinates, a^e is the linear acceleration in ECEF coordinates and g^e is the gravity vector in ECEF coordinates. The gravity vector is a function of the position and is modeled using the J_2 gravity model [19]. The known Earth's angular velocity around the ECEF z-axis is represented by vector ω^e_{ie} and $\tilde{\omega}^b_{ib} = [0\ \omega^b]^T$ is the quaternion representation of the angular velocities in body frame. The quaternion q^e_b determines the orientation of the rigid body in space and α and β are the gyroscope and accelerometer bias, respectively.

The measurement equations with measurement noise v are given by:

$$\omega^b_m = \omega^b + \alpha + \mathrm{v}_\omega \tag{6}$$
$$a^b_m = R(q^e_b)^T a^e + \beta + \mathrm{v}_a \tag{7}$$
$$m^b_m = R(q^e_b)^T m^e + \mathrm{v}_m \tag{8}$$
$$p^e_m = p^e + \mathrm{v}_p \tag{9}$$
$$v^e_m = v^e + \mathrm{v}_v \tag{10}$$

where m^e is a known vector that contains the values of the magnitude of the terrestrial magnetic field given our current latitude and longitude[1], ω^b and a^e are the angular velocity and linear acceleration vectors in body and ECEF coordinates, respectively. The matrix $R(q^e_b)$ is the rotation matrix associated with the current orientation quaternion.

In order to use GNSS data with Eqs. (1)-(5), we need to convert it to ECEF coordinates. This can be done using the following equations:

$$\begin{aligned} x^e &= (N_e + h)\cos\phi\cos\lambda \\ y^e &= (N_e + h)\cos\phi\sin\lambda \\ z^e &= (b^2 N_e/a^2 + h)\sin\phi \end{aligned} \tag{11}$$

[1]This data is tabulated and can be obtained from https://www.ngdc.noaa.gov/geomag-web

where

$$N_e = \frac{a^2}{\sqrt{a^2\cos^2\phi + b^2\sin^2\phi}}$$

is the Earth's east-west radius of curvature, ϕ is the latitude in radians, λ is the longitude in radians, h is the altitude in meters, $a = 6378137$ m and $b = 6356752.3142$ m are the major and minor axes of the Earth reference ellipsoid, respectively.

The set of Eqs. (1)-(5) model the position, velocity and orientation of a vehicle in ECEF coordinates. However, if we wish to travel short distances it is convenient to use ENU coordinates and work in a local reference frame. The steps to convert from ECEF to ENU are the following:

1) Determine the latitude, longitude and altitude of the initial reference position (ϕ_0, λ_0, h_0) and calculate its ECEF coordinates using equation (11) to obtain vector $p^e_0 = [x^e_0, y^e_0, z^e_0]^T$. This position will be the origin of the ENU coordinate system.
2) Transform the incoming GNSS measurements to ECEF coordinates using equation (11) to obtain p^e and compute the relative displacements in ENU coordinates using the following:

$$p^n = R^e_n(\phi_0, \lambda_0)^T(p^e - p^e_0) \tag{12}$$

where $R^e_n(\phi_0, \lambda_0)$ is the ENU to ECEF rotation matrix and depends on the initial latitude and longitude (ϕ_0, λ_0).

The ENU to ECEF rotation matrix is given by two rotations [20]:

1) A clockwise rotation over east-axis by an angle $90 - \phi$ to align the up-axis with the z-axis. That is $R_1(-(\pi/2 - \phi))$.
2) A clockwise rotation over the z-axis by an angle $90 + \lambda$ to align the east-axis with the x-axis. That is $R_3(-(\phi/2 + \lambda))$.

Where rotation matrices are defined as follows:

$$R_1(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \tag{13}$$
$$R_2(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{14}$$
$$R_3(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{15}$$

in matrix form, we obtain

$$\begin{bmatrix} x^e \\ y^e \\ z^e \end{bmatrix} = R_3(-(\phi/2 + \lambda))\ R_1(-(\pi/2 - \phi)) \begin{bmatrix} x^n \\ y^n \\ z^n \end{bmatrix} \tag{16}$$

where we assume that the x-axis points to the East when using ENU coordinates. Taking into account the properties of rotation matrices, the ECEF to ENU transformation is obtained through the transpose of the matrix given by the previous

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
G. Sanchez, M. Murillo, L. Genzelis, Nestor N. Deniz & L. Giovanini; "Moving horizon estimation for GNSS-IMU sensor fusion."
IX Jornadas Argentinas de Robótica (JAR), 2017.

equation. In this way, equation (16) gives a formula to convert coordinates from ENU to ECEF and from ECEF to ENU.

By using ENU, we establish a local coordinate system relative to the reference position p_0^e. We must replace our orientation quaternion from q_b^e to q_b^n. Besides, we must be very careful and know exactly in which frame of reference each of the parameters, constants and sensor measurements are given in order to apply the corresponding rotations to them.

III. Implementation details

A. MHE implementation

The MHE implementation follows the algorithm presented in the work of Rao et al. [21], [22]. In our case, the vector of differential and algebraic states are defined as

$$x = [p^n \ v^n \ q_b^n \ \alpha \ \beta]^T \tag{17}$$

$$z = [\omega^b \ a^n]^T \tag{18}$$

and the measurement vector is defined as

$$y = [\omega_m^b \ a_m^b \ m_m^b \ p_m^e \ v_m^e]^T \tag{19}$$

The cost function Ψ that will be minimized with respect to $\hat{x}_{k-N|k}$, $\hat{z}_{k-N|k}$ and $\hat{\mathrm{w}}_k$ is defined as

$$\Psi_k^N = ||1 - ||\hat{q}_b^n(k-N)||_2^2||_{P_0}^2 + ||\hat{x}_{k-N|k} - \bar{x}_{k-N|k}||_{P_1}^2 + ||\hat{z}_{k-N|k} - \bar{z}_{k-N|k}||_{P_2}^2 + \sum_{j=k-N}^{k} ||\hat{\mathrm{w}}_{j|k}||_Q^2 + ||\hat{\mathrm{v}}_{j|k}||_R^2 \tag{20}$$

Given that the quaternion $\hat{q}_b^n$ must have unit norm, the constraint $||\hat{q}_b^n||_2^2 = 1$ could be included. However, to avoid the computational cost of this restriction, the first term of Ψ, which penalizes its violation at $k-N$, and the following set of constraints

$$\begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \leq \hat{q}_b^n \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{21}$$

are added to the problem.

The horizon length N and the values of the weights P_0, P_1, P_2, Q and R were chosen by a trial and error procedure as $N = 5$, $P_0 = 0.1$, $P_1 = I$, $P_2 = I$, $Q = 0.001\,I$ and $R = diag([10, 10, 10, 10, 10, 10, 5, 5, 5, 1, 1, 1, 1, 1, 1])$. The resulting MHE constrained non-linear optimization problem is solved with CasADi [23] and Ipopt [24].

B. EKF implementation

The implementation of the Extended Kalman Filter follows the standard procedure; however, there are a couple of subtleties. Firstly, gyroscope and accelerometer readings are treated as control inputs instead of as measurements. To that end, ω^b and a^n, which were previously regarded as algebraic states, are expressed as functions of the inputs ω_m^b and a_m^b and subsequently eliminated from the problem formulation. The differential states remain the same as in the MHE formulation, while the measurement vector is comprised of the remaining data readings, namely,

$$y = [m_m^b \ p_m^e \ v_m^e]^T. \tag{22}$$

And secondly, the quaternion $\hat{q}_b^n$ must be renormalized at each time step, given that there is no way to take this constraint into account in the EKF, as was done in the MHE implementation.

The covariance matrices Q and R of the EKF are chosen as the inverse of the weighting matrices employed in the MHE formulation, given that a smaller covariance in the former must correspond to a bigger weight, i.e., "trust", in the latter. Therefore, the covariance matrices are defined as $Q = 1000\,I$ and $R = diag([0.2, 0.2, 0.2, 1, 1, 1, 1, 1, 1])$.

IV. Example

In the following example, we will perform a manoeuvre using Gazebo and ROS to run a simulation of the Husky[2] unmanned ground vehicle moving to the following set of *way-points*: $w_1 = [5; 0; 0]^T$, $w_2 = [15; 10; 0]^T$, $w_3 = [20; 10; 0]^T$, $w_4 = [30; 0; 0]^T$ and $w_5 = [35; 0; 0]^T$. As stated before, the vehicle is equipped with GNSS and IMU sensors, which will be used to estimate the position and orientation in ENU coordinates using MHE and EKF. Both of these estimated values will be compared to the true values.

Figures 1a, 1c and 1e show the true position p^n and its estimates $\hat{p}^n$ in ENU coordinates. It can be seen that both the MHE and EKF provide good estimates. Figures 1a and 1c show that both estimators are able to follow the changes on the x and y axis. Since the vehicle is moving on flat terrain, the z coordinate is only affected by noise (see Fig. 1e). Figures 1b, 1d and 1f show the difference $\hat{p}^n - p^n$. It can be seen that MHE error is slightly smaller on the x and y axis, while EKF filters slightly better the noise on the z axis.

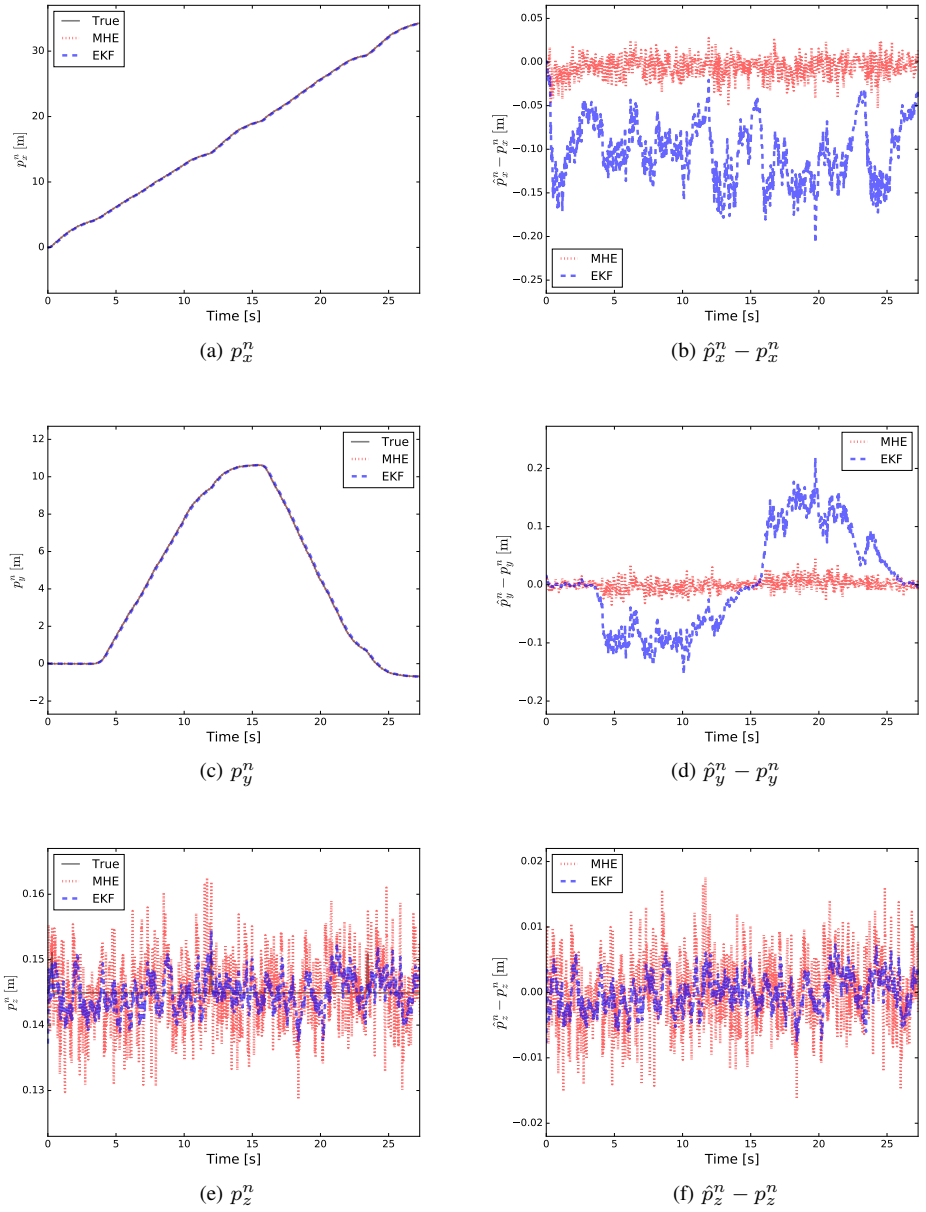
The orientation quaternion q^n and its estimates $\hat{q}^n$ are shown in Figures 2 and 3, where it can be seen a similar behaviour than the one obtained from the position. MHE performs a better estimation of the states that change through time –q_0^n and q_3^n–, as it can be seen in Figures 2a, 2b, 3c and 3d, while EKF is able to do a slightly better job at filtering the noise on q_1^n and q_2^n, as it can be seen in Figures 2c, 2d, 3a and 3b.

The results obtained by MHE can be attributed to the fact that: *i)* MHE uses more measurements to obtain the current estimate; *ii)* MHE does not assume Gaussian distribution for the process and measurement noises within the estimation horizon, such as the EKF.

Finally, Table I shows the mean and the standard deviation of the squared error between the real state and the estimated state over 50 realizations of the same experiment. The average execution time for each sample was 5.293 milliseconds for each MHE iteration and 0.211 milliseconds for each EKF iteration.

[2]https://www.clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/

sinc(i) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
G. Sanchez, M. Murillo, L. Genzelis, Nestor N. Deniz & L. Giovanini; "Moving horizon estimation for GNSS-IMU sensor fusion."
IX Jornadas Argentinas de Robótica (JAR), 2017.



(a) p_x^n (b) $\hat{p}_x^n - p_x^n$

(c) p_y^n (d) $\hat{p}_y^n - p_y^n$

(e) p_z^n (f) $\hat{p}_z^n - p_z^n$

Fig. 1. Estimation of the position p^n in ENU coordinates.

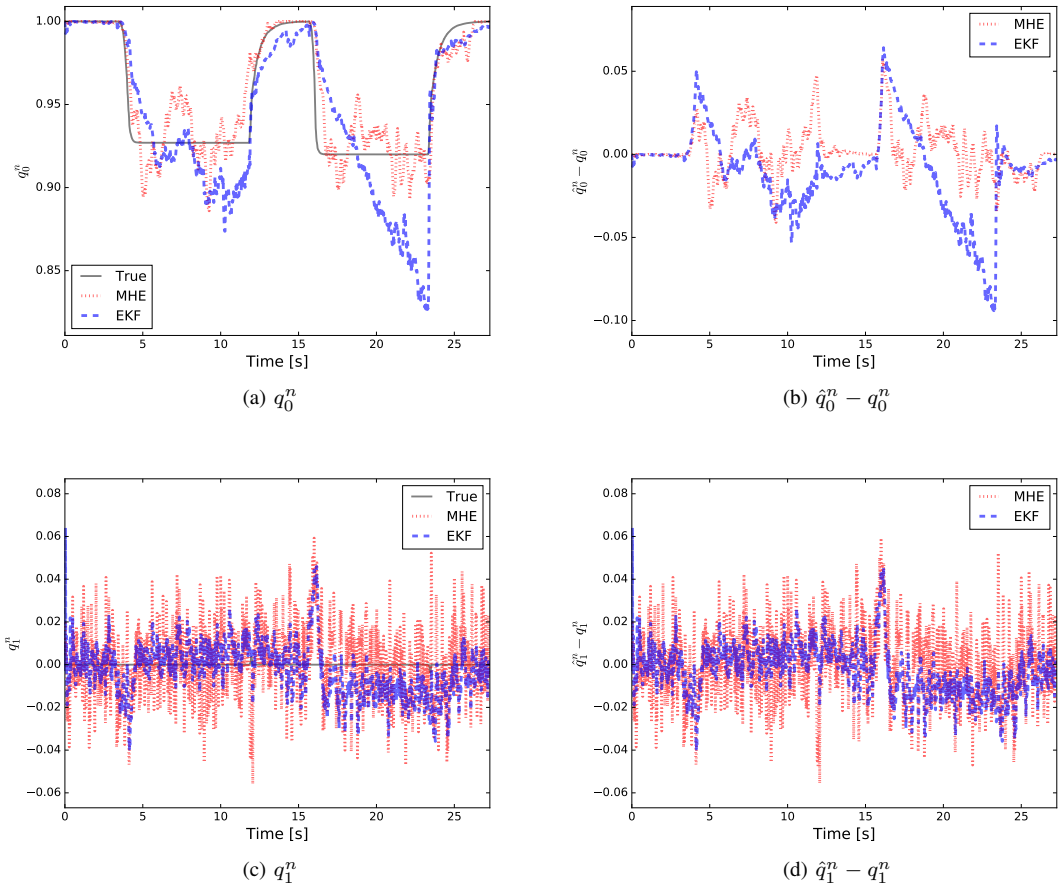
	Mean ($\times 10e{-}3$)		Std. Dev. ($\times 10e{-}3$)	
	MHE	EKF	MHE	EKF
p_x	0.30643	0.28912	0.36298	0.33779
p_y	0.18676	0.17014	0.22421	0.20285
p_z	0.02815	0.00870	0.03828	0.01201
v_x	0.93519	1.70043	1.28402	2.32408
v_y	0.92798	1.67139	1.26584	2.26883
v_z	1.09800	2.12559	1.50401	2.90566
q_0	0.19255	1.08827	0.28763	1.17203
q_1	0.22829	0.06821	0.31084	0.09292
q_2	0.22064	0.07187	0.30205	0.09909
q_3	1.73780	5.35359	2.38613	5.35354

TABLE I
MEAN AND STANDARD DEVIATION OF THE SQUARED ERROR.

V. CONCLUSION

In this work we employed MHE to estimate the position, velocity and orientation of an unmanned ground vehicle by fusing data from GNSS and IMU sensors. These estimates are compared with the classic benchmark algorithm, the EKF and the true values. MHE is able to perform better than EKF at a fast rate (around 100 Hz), which indicates that it can easily be used for real time estimation. Since MHE solves a non linear optimization problem on each iteration, the addition of constraints and bounds such as the ones described by Eq. (21), is straight forward.

Since the solution of the navigation problem requires a very specific set of knowledge and the use of different coordinate



(a) q_0^n (b) $\hat{q}_0^n - q_0^n$

(c) q_1^n (d) $\hat{q}_1^n - q_1^n$

Fig. 2. Estimation of the orientation quaternion q^n in ENU coordinates.

systems often leads to confusion, we also showed all the necessary steps to perform position, velocity and orientation estimation either in ECEF or ENU coordinates. One of the issues that remains open is how to tune both MHE and EKF weight matrices in order to provide better results.

If we want to run these algorithms with real sensors, special care must be taken in order to account for different sampling rates, especially when typical GNSS receivers sampling rate is around 1 Hz to 10 Hz and commercial IMUs sampling rate is around 500 Hz.

Acknowledgment

The authors wish to thank the *Universidad Nacional de Litoral* (with CAI+D Jóven 500 201501 00050 LI and CAI+D 504 201501 00098 LI), the *Agencia Nacional de Promoción Científica y Tecnológica* (with PICT 2016-0651) and the *Consejo Nacional de Investigaciones Científicas y Técnicas* (CONICET) from Argentina, for their support.

We also would like to thank the groups behind the development of CasADi, Ipopt and the HSL solvers [25].

References

[1] T. Polóni, B. Rohal-Ilkiv, and T. Johansen, "Moving Horizon Estimation for Integrated Navigation Filtering," *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 519–526, 2015. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S2405896315026154

[2] J. Vandersteen, M. Diehl, C. Aerts, and J. Swevers, "Spacecraft Attitude Estimation and Sensor Calibration Using Moving Horizon Estimation," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 3, pp. 734–742, may 2013. [Online]. Available: http://arc.aiaa.org/doi/10.2514/1.58805

[3] E. Lefferts, F. Markley, and M. Shuster, "Kalman filtering for spacecraft attitude estimation," pp. 417–429, 1982. [Online]. Available: http://arc.aiaa.org/doi/pdf/10.2514/3.56190

[4] F. Markley and J. Sedlak, "Kalman Filter for Spinning Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 6, pp. 1750–1760, nov 2008. [Online]. Available: http://dx.doi.org/10.2514/1.35221

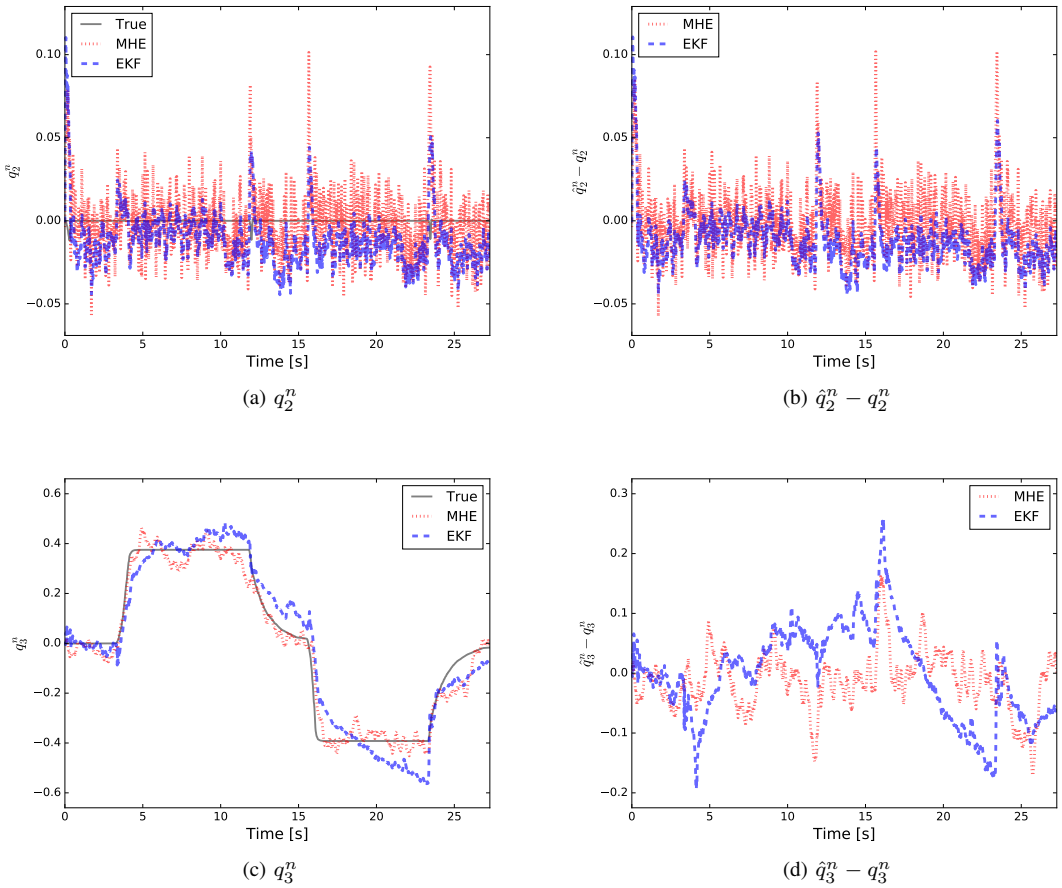
[5] S. Roumeliotis and G. Bekey, "3-D Localization for a Mars Rover Prototype," in *Artificial Intelligence, Robotics and Automation in Space, Proceedings of the Fifth International Symposium, ISAIRAS '99*, 1999, pp. 441–448. [Online]. Available: http://adsabs.harvard.edu/abs/1999ESASP.440..441R

[6] J. Crassidis and F. Markley, "Unscented Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 4, pp. 536–542, jul 2003. [Online]. Available: http://arc.aiaa.org/doi/10.2514/2.5102

[7] M. Rhudy, Y. Gu, J. Gross, S. Gururajan, and M. R. Napolitano, "Sensitivity Analysis of Extended and Unscented Kalman Filters for Attitude Estimation," *Journal of Aerospace Information Systems*, vol. 10, no. 3, pp. 131–143, mar 2013. [Online]. Available: http://arc.aiaa.org/doi/10.2514/1.54899

[8] A. Carmi and Y. Oshman, "Adaptive Particle Filtering for Spacecraft Attitude Estimation from Vector Observations," *Journal of Guidance,*

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
G. Sanchez, M. Murillo, L. Genzelis, Nestor N. Deniz & L. Giovanini, "Moving horizon estimation for GNSS-IMU sensor fusion."
IX Jornadas Argentinas de Robótica (JAR), 2017.



(a) q_2^n (b) $\hat{q}_2^n - q_2^n$

(c) q_3^n (d) $\hat{q}_3^n - q_3^n$

Fig. 3. Estimation of the orientation quaternion q^n in ENU coordinates.

Control, and Dynamics, vol. 32, no. 1, pp. 232–241, jan 2009. [Online]. Available: http://arc.aiaa.org/doi/10.2514/1.35878

[9] Y. Cheng and J. Crassidis, "Particle Filtering for Sequential Spacecraft Attitude Estimation," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Reston, Virigina: American Institute of Aeronautics and Astronautics, aug 2004. [Online]. Available: http://arc.aiaa.org/doi/10.2514/6.2004-5337

[10] H. F. Grip, T. I. Fossero, T. A. Johansent, and A. Saberi, "A nonlinear observer for integration of gnss and imu measurements with gyro bias estimation," in *American Control Conference (ACC), 2012*. IEEE, 2012, pp. 4607–4612.

[11] E. Haseltine and J. Rawlings, "Critical evaluation of extended kalman filtering and moving-horizon estimation," *Industrial & engineering chemistry research*, vol. 44, no. 8, pp. 2451–2460, 2005.

[12] D. Simon, "Kalman filtering with state constraints: a survey of linear and nonlinear algorithms," *Control Theory & Applications, IET*, vol. 4, no. 8, pp. 1303–1318, 2010.

[13] D. Simon and T. Chia, "Kalman filtering with state equality constraints," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 38, no. 1, pp. 128–136, 2002.

[14] D. Hall and S. McMullen, *Mathematical techniques in multisensor data fusion*. Artech House, 2004.

[15] B. Teixeira, J. Chandrasekar, H. Palanthandalam-Madapusi, L. Tôrres, L. Aguirre, and D. Bernstein, "Gain-constrained kalman filtering for linear and nonlinear systems," *Signal Processing, IEEE Transactions on*, vol. 56, no. 9, pp. 4113–4123, 2008.

[16] D. Simon and D. Simon, "Constrained kalman filtering via density function truncation for turbofan engine health estimation," *International Journal of Systems Science*, vol. 41, no. 2, pp. 159–171, 2010.

[17] S. Ko and R. Bitmead, "State estimation for linear systems with state equality constraints," *Automatica*, vol. 43, no. 8, pp. 1363–1368, 2007.

[18] E. Bekir, *Introduction to modern navigation systems*. World Scientific, 2007.

[19] D. Hsu, "Comparison of four gravity models," in *Proceedings of Position, Location and Navigation Symposium - PLANS '96*. IEEE, 1996, pp. 631–635. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=509138

[20] J. Z. J. Sanz-Subirana and M. Hernández-Pajares, "Transformations between ECEF and ENU coordinates," http://www.navipedia.net/index.php/Transformations_between_ECEF_and_ENU_coordinates, 2011, [Online; accessed 30-07-2017].

[21] C. V. Rao, J. B. Rawlings, and J. H. Lee, "Constrained linear state estimation—a moving horizon approach," *Automatica*, vol. 37, no. 10, pp. 1619–1628, oct 2001. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0005109801001157

[22] C. V. Rao, J. B. Rawlings, and D. Q. Mayne, "Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations," *IEEE transactions on automatic control*, vol. 48, no. 2, pp. 246–258, 2003.

[23] J. Andersson, "A General-Purpose Software Framework for Dynamic Optimization," PhD thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium, October 2013.

[24] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[25] The HSL Mathematical Software Library, "HSL. A collection of Fortran codes for large scale scientific computation," http://www.hsl.rl.ac.uk, 2016, [Online; accessed 30-07-2017].