
Nonlinear Data Analytics Midterm Progress Report

John Kath
john.kath@cgu.edu

Contents

1	Project Overview	1
1.1	Description	1
1.2	Requirements	1
1.3	Project goals	2
2	Project Data Analytics	2
2.1	HMOG cell phone data	2
2.1.1	Measurement models	2
2.2	Probabilistic models	2
2.3	Pose estimation	3
2.4	Orientation estimation	3
2.5	Estimating position and orientation	3
2.6	Gauss-Newton optimization	3
2.7	Computing the uncertainty	4
3	Project Implementation	5

1 Project Overview

1.1 Description

Project Type (1)

1. Application of existing algorithm to a new problem and potentially new data.

1.2 Requirements

- **Partner:** Work independently.
- **Dataset:** HMOG cell phone data from at least five users/activities. Will apply data analytics and ML methods using accelerometer, gyroscope and magnetometer data (3-axis each).
- **Format:** \LaTeX NIPS
- **Code Style:** Will use suggested code style guidelines (cookiecutter data science) with MIT open-source license.
- **Programming Tools & Hardware:** Python/Jupyter Notebook, C++, NVIDIA Jetson Nano with Jetpack API.

1.3 Project goals

This project uses probabilistic models and data analytics methods introduced in the paper [1] *Using Inertial Sensors for Position and Orientation Estimation* to estimate the position and orientation (pose) of a users cell phone during an activity. Cell phone 3D accelerometer, gyroscope and magnetometer data is obtained for analysis from the HMOG dataset associated with the article [2] *HMOG: New Behavioral Biometric Features for Continuous Authentication of Smartphone Users*.

The techniques for data analytics used in the project as discussed in [1] *Using Inertial Sensors for Pose Estimation* are:

- **(Ch2) Inertial Sensors:** Coordinate frames, angular velocity, specific force, sensor error.
- **(Ch3) Probabilistic Models:** Parameterizing/probabilistic orientation modeling (Euler angles, Unit quaternions), measurement/probabilistic models for pose estimation.
- **(Ch4) Estimating Position and Orientation:** Smoothing in optimized frame (Gauss-Newton estimation, uncertainty), smoothing estimation of orientation using optimization, filtering estimate of orientation using optimization, filtering estimate in optimization framework, extended Kalman filtering / complementary filtering.

2 Project Data Analytics

NOTE: The following sections restate material from reference [1].

2.1 HMOG cell phone data

3D accelerometer, gyroscope and magnetometer data (3 axis each) [2] HMOG cell phone data for a given user/activity is used as an input to our pose algorithms. The algorithm output is an estimation of orientation (and possibly position). Three algorithms for estimating position and orientation will be introduced in the following sections.

2.1.1 Measurement models

Accelerometer measurement model

The accelerometer measures the specific force f_t^b at each time instance t . Our simplified measurement model is

$$y_{a,t} = R_t^{bn}(a_{nn}^n - g^n) + \delta_{a,t}^b + e_{a,t}^b. \quad (1)$$

Gyroscope measurement model

The gyroscope measures the angular velocity ω_{ib}^b at each time instance t . Our simplified measurement model is

$$y_{\omega,t} = \omega_{nb,t}^b + \delta_{\omega,t}^b + e_{\omega,t}^b. \quad (2)$$

Magnetometer measurement model

Magnetometers measure the local magnetic field, consisting of both the earth magnetic field and the magnetic field due to the presence of magnetic material. Assuming that the magnetometer only measures the local magnetic field, its measurements $y_{m,t}$ can be modeled as

$$y_{m,t} = R_t^{bn}m^n + e_{m,t}, \quad (3)$$

2.2 Probabilistic models

Most complexity in pose estimation lies in the nonlinear nature of the orientation and the fact that orientation can be parametrized in different ways. When all the measurement data is used in our model this is know as *smoothing*. This is computation intensive and assumes all measurement data is available for our estimate. The other class of estimation we will model is know as *filtering*. In filtering we estimate x_t using all measurements up to and including time t .

2.3 Pose estimation

For pose estimation, we model the accelerometer and gyroscope measurements as inputs to the dynamics. Hence, the state vector consists of the position p_t^n , the velocity v_t^n and a parametrization of the orientation. In summary, this leads to the following state space model for pose estimation

$$\begin{pmatrix} p_{t+1}^n \\ v_{t+1}^n \\ q_{t+1}^{nb} \end{pmatrix} = \begin{pmatrix} p_t^n + T v_t^n + \frac{T^2}{2} (R_t^{nb} (y_{a,t} - \delta_{a,t}) + g^n + e_{p,a,t}) \\ v_t^n + T (R_t^{nb} (y_{a,t} - \delta_{a,t}) + g^n + e_{v,a,t}) \\ q_t^{nb} \odot \exp_q \left(\frac{T}{2} (y_{\omega,t} - \delta_{\omega,t} - e_{\omega,t}) \right) \end{pmatrix}, \quad (4a)$$

$$y_{p,t} = p_t^n + e_{p,t}, \quad (4b)$$

where

$$e_{p,a,t} \sim \mathcal{N}(0, \Sigma_a), \quad e_{v,a,t} \sim \mathcal{N}(0, \Sigma_a), \quad (4c)$$

$$e_{p,t} \sim \mathcal{N}(0, \Sigma_p), \quad e_{\omega,t} \sim \mathcal{N}(0, \Sigma_\omega), \quad (4d)$$

with $\Sigma_a = \sigma_a^2 \mathcal{I}_3$ and $\Sigma_\omega = \sigma_\omega^2 \mathcal{I}_3$.

$$(4e)$$

2.4 Orientation estimation

For orientation estimation, the state vector only consists of a parametrization of the orientation. We use the inertial sensors in combination with the magnetometer measurements to estimate the orientation. This leads to the following state space model for orientation estimation,

$$q_{t+1}^{nb} = q_t^{nb} \odot \exp_q \left(\frac{T}{2} (y_{\omega,t} - \delta_\omega - e_{\omega,t}) \right), \quad (5a)$$

$$y_{a,t} = -R_t^{bn} g^n + e_{a,t}, \quad (5b)$$

$$y_{m,t} = R_t^{bn} m^n + e_{m,t}, \quad (5c)$$

where (5a) describes the dynamics while (5b) and (5c) describe the measurement models and

$$e_{\omega,t} \sim \mathcal{N}(0, \Sigma_\omega), \quad e_{a,t} \sim \mathcal{N}(0, \Sigma_a), \quad e_{m,t} \sim \mathcal{N}(0, \Sigma_m), \quad (5d)$$

with $\Sigma_\omega = \sigma_\omega^2 \mathcal{I}_3$ and $\Sigma_a = \sigma_a^2 \mathcal{I}_3$. The initial orientation is given by the QUEST algorithm.

2.5 Estimating position and orientation

We will focus on position and orientation estimation using the probabilistic models developed. Algorithm 1 is a orientation estimate based on a smoothing. Algorithms 2 and 3 use filtering to estimate orientation. Our focus will be orientation estimation problems since they illustrate the most important parts of the pose estimation. Most complexities lie in the parametrization of the orientation and in the nonlinear nature of the orientation.

2.6 Gauss-Newton optimization

To obtain a smoothing estimate of the position and orientation using optimization, we first recognize that for our models (4) and (5), all probability distributions are Gaussian. Let us therefore consider a slightly more general problem where the objective function consists of the product of Gaussian probability functions $p(e_i(x_{1:N}))$, $i = 1, \dots, M$. Hence, the optimization problem can be written as

$$\hat{x}_{1:N} = \arg \min_{x_{1:N}} - \sum_{i=1}^M \log p(e_i(x_{1:N})). \quad (6)$$

The probability distribution of $e_i(x)$ is given by

$$p(e_i(x_{1:N})) = \frac{1}{\sqrt{(2\pi)^{n_e} \det \Sigma_i}} \exp \left(-\frac{1}{2} e_i^\top(x_{1:N}) \Sigma_i^{-1} e_i(x_{1:N}) \right). \quad (7)$$

Omitting the terms independent of $x_{1:N}$, the optimization problem (6) reduces to

$$\hat{x}_{1:N} = \arg \min_{x_{1:N}} \frac{1}{2} \sum_{i=1}^M \|e_i(x_{1:N})\|_{\Sigma_i^{-1}}^2, \quad (8)$$

with $\|e_i(x_{1:N})\|_{\Sigma_i^{-1}}^2 = e_i^\top(x_{1:N})\Sigma_i^{-1}e_i(x_{1:N})$. The function that is being minimized in optimization problems, is often referred to as the *objective function*.

The solution to (8) can be found by studying the shape of the objective function as a function of $x_{1:N}$. This can be characterized in terms of the *gradient* $\mathcal{G}(x_{1:N})$ and *Hessian* $\mathcal{H}(x_{1:N})$, which provide information about the slope and curvature of the function, respectively. Defining

$$e_i^\top(x_{1:N})\Sigma_i^{-1}e_i(x_{1:N}) = \varepsilon_i^\top \varepsilon_i, \quad \varepsilon_i = \Sigma_i^{-1/2}e_i(x_{1:N}),$$

and the stacked variables

$$\varepsilon = (\varepsilon_1^\top \quad \cdots \quad \varepsilon_M^\top)^\top,$$

the gradient and the Hessian are given by

$$\mathcal{G}(x_{1:N}) = \sum_{i=1}^M \left(\frac{\partial \varepsilon_i}{\partial x_{1:N}} \right)^\top \varepsilon_i = \mathcal{J}^\top(x_{1:N})\varepsilon, \quad (9a)$$

$$\begin{aligned} \mathcal{H}(x_{1:N}) &= \sum_{i=1}^M \left(\left(\frac{\partial \varepsilon_i}{\partial x_{1:N}} \right)^\top \frac{\partial \varepsilon_i}{\partial x_{1:N}} + \varepsilon_i^\top \frac{\partial^2 \varepsilon_i}{\partial x_{1:N}^2} \right) \\ &= \mathcal{J}^\top(x_{1:N})\mathcal{J}(x_{1:N}) + \sum_{i=1}^M \varepsilon_i^\top \frac{\partial^2 \varepsilon_i}{\partial x_{1:N}^2}. \end{aligned} \quad (9b)$$

Note that for notational convenience, we have omitted the explicit dependence of ε on $x_{1:N}$. In (9), we introduced the notation $\mathcal{J}(x_{1:N})$, which is the *Jacobian* of the vector ε with respect to $x_{1:N}$ as

$$\mathcal{J}(x_{1:N}) = \begin{pmatrix} \frac{\partial \varepsilon_1}{\partial x_1} & \cdots & \frac{\partial \varepsilon_1}{\partial x_N} \\ \vdots & & \vdots \\ \frac{\partial \varepsilon_{Mn_\varepsilon}}{\partial x_1} & \cdots & \frac{\partial \varepsilon_{Mn_\varepsilon}}{\partial x_N} \end{pmatrix}, \quad (10)$$

where n_ε is the length of the vector ε_i . Instead of computing the true Hessian (9b), we compute an approximation of it given by

$$\hat{\mathcal{H}}(x_{1:N}) = \mathcal{J}^\top(x_{1:N})\mathcal{J}(x_{1:N}). \quad (11)$$

This has the benefit of not having to compute second derivatives, at the same time as it guarantees that the Hessian is positive semidefinite. The downside of using (11) is that it introduces an approximation.

The gradient and the (approximate) Hessian can be used to find the minimum of the objective function. For our models (4) and (5), in which the functions $e_i(x_{1:N})$ are nonlinear, an estimate $\hat{x}_{1:N}$ can iteratively be computed as

$$\hat{x}_{1:N}^{(k+1)} = \hat{x}_{1:N}^{(k)} - \beta^{(k)} \left(\hat{\mathcal{H}}(\hat{x}_{1:N}^{(k)}) \right)^{-1} \mathcal{G}(\hat{x}_{1:N}^{(k)}), \quad (12)$$

where k denotes the iteration number. The *step length* $\beta^{(k)}$ is computed for instance using a backtracking line search. The *search direction* is computed as $\left(\hat{\mathcal{H}}(\hat{x}_{1:N}^{(k)}) \right)^{-1} \mathcal{G}(\hat{x}_{1:N}^{(k)})$. Note that an initial point $\hat{x}_{1:N}^{(0)}$ needs to be chosen close enough to the desired minimum to ensure convergence to this minimum.

At each Gauss-Newton iteration (12), we estimate the state vector $\eta_{1:N}^n$. Before starting the next iteration, the linearization points $\tilde{q}_{1:N}^{\text{nb}}$ are updated and the state vector $\eta_{1:N}^n$ is reset to zero.

2.7 Computing the uncertainty

We are not only interested in obtaining point estimates of the position and orientation, but also in estimating their uncertainty. We can therefore approximate the covariance of our estimates as

$$\text{cov}(\hat{x}_{1:N}) = \left(\mathcal{J}^\top(\hat{x}_{1:N})\mathcal{J}(\hat{x}_{1:N}) \right)^{-1}. \quad (13)$$

Algorithm 1 Smoothing estimates of the orientation using optimization

INPUTS: An initial estimate of the orientation $\tilde{q}_{1:N}^{nb,(0)}$, inertial data $\{y_{a,t}, y_{\omega,t}\}_{t=1}^N$, magnetometer data $\{y_{m,t}\}_{t=1}^N$ and covariance matrices Σ_{ω} , Σ_a and Σ_m .

OUTPUTS: An estimate of the orientation $\hat{q}_{1:N}^{nb}$ and optionally its covariance $\text{cov}(\hat{\eta}_{1:N}^n)$.

1. Set $\hat{\eta}_t^{n,(0)} = 0_{3 \times 1}$ for $t = 1, \dots, N$, set $k = 0$ and compute \tilde{q}_1^{nb} and $\Sigma_{\eta,i}$.
2. **while** *termination condition is not satisfied* **do**
 - (a) Compute the gradient (9a) and the approximate Hessian (11) of the orientation smoothing problem using the expressions for the different parts of the cost function and their Jacobians.
 - (b) Apply the update (12) to obtain $\hat{\eta}_{1:N}^{n,(k+1)}$.
 - (c) Update the linearization point as

$$\tilde{q}_t^{nb,(k+1)} = \exp_q \left(\frac{\hat{\eta}_t^{n,(k+1)}}{2} \right) \odot \tilde{q}_t^{nb,(k)}, \quad (14)$$

and set $\hat{\eta}_t^{n,(k+1)} = 0_{3 \times 1}$ for $t = 1, \dots, N$.

- (d) Set $k = k + 1$.

end while

3. Set $\hat{q}_{1:N}^{nb} = \tilde{q}_{1:N}^{nb,(k)}$.
4. Optionally compute

$$\text{cov}(\hat{\eta}_{1:N}^n) = \left(\mathcal{J}^T(\hat{\eta}_{1:N}^n) \mathcal{J}(\hat{\eta}_{1:N}^n) \right)^{-1}. \quad (15)$$

3 Project Implementation

Dataset - Algorithm 1:

Based on a thorough perusal of article [1] and related reference material I am currently developing the python code (Jupyter Notebook) implementation for Algorithm 1. Datasets from reference [2] HMOG cell phone data for a given user(s) activities will be read into the program.

Analytics - Algorithm 1 smoothing orientation estimation:

Inertial data (accelerometer, gyroscope) and magnetometer data (3-axis each) from dataset are *inputs* to Algorithm 1 producing an *output* estimate of the orientation and optionally its covariance.

The following functions are being written to support Algorithm 1:

Gauss-Newton optimization: gradient, Hessian approximation and Jacobian functions.

Algorithm 1 also requires developing orientation estimation based on the appropriate probabilistic model and representing the state vector. A covariance matrix function will be written to represent computational uncertainty.

Next steps (as time permits)- Algorithms 2 & 3 filtering orientation estimation:

Upon completing Algorithm 1, the optimization and Extended Kalman Filter filtering orientation estimation algorithms will be developed (Algorithms 2 and 3).

References

- [1] Manon Kok, Jeroen D. Hol and Thomas B. Schon (2017), Using Inertial Sensors for Position and Orientation Estimation, *Foundations and Trends in Signal Processing: Vol. 11: No. 1-2*, pp 1-153.
<http://dx.doi.org/10.1561/20000000094>
- [2] SITOVIĆ, Zdeňka, Jaroslav ŠEDĚNKA, Qing YANG, Ge PENG, Gang ZHOU, Paolo GASTI and Kiran BALAGANI. HMOG: New Behavioral Biometric Features for Continuous Authentication of Smartphone Users. *IEEE Transactions on Information Forensics and Security*, 2016, Vol. 11, No. 5, p. 877 - 892. ISSN 1556-6013.
<http://dx.doi.org/10.1109/TIFS.2015.2506542>