Chapter 5: GMWM online, noise units units conversion & navigation toolbox

Jan Skaloud* & Stéphane Guerrier¶

*École Polytechnique Fédérale de Lausanne; ¶Université de Genève

Material available online: https://gmwm.netlify.app/

EPFL - Spring 2025

Agenda - course

- D1.js: Intro to modelling w. examples (installations)
- D2.sg: Time series & Allan variance (AV exercises)
- D3.sg: General Methods of Wavelet Moments (GMWM exercises)
- D4.js: Impact of stoch. models on trajectory (project definition)
- D5.sg: Extended GMWM (multi series, model selection) & Statistical applications (regression settings with space-time dependence)

Agenda - today

- GMWM via Web
- Unit conversion (from discrete to continuous)
- Navigation toolbox
- Projects

GMWN with Graphical User Interface (GUI)

Principal description

Feature Article:

DOI. No. 10.1109/MAES.2018.170153

Use of a New Online Calibration Platform With Applications to Inertial Sensors

Philipp Clausen, Jan Skaloud, **École Polytechnique Fédérale de Lausanne, Switzerland**
Roberto Molinari, Justin Lee, Stéphane Guerrier, **Pennsylvania State University, PA, USA**

INTRODUCTION

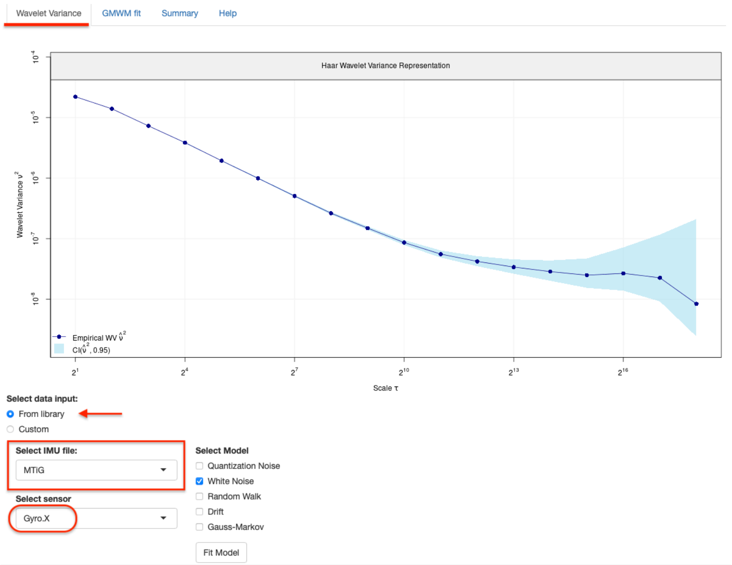
In many fields, going from economics to physics, it is common to deal with measurements that are taken in time. These measurements are often explained by known external factors that describe a large part of their behavior. For example, the evolution of the unemployment rate in time can be explained by the behavior of the gross domestic product (the external factor in this case). However, in many cases the external factors are not enough to explain the entire behavior of the measurements, and it is necessary to use so-called stochastic models (or probabilistic models) that describe

respectively. The integration of this information with other sensors (e.g., odometer, barometer, Global Positioning System (GPS)) allows the increase of the performances, which could never be achieved individually.

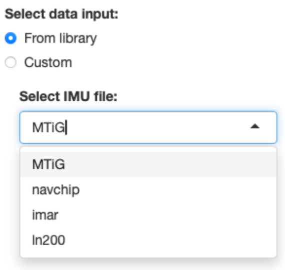
There are different types of IMUs, but one characteristic that distinguishes them is their level of measurement precision. Indeed, the so-called high-end-IMUs (i.e., navigational grade) have excellent properties regarding their errors and measurement-noise, meaning that they measure attitude and position with almost perfect precision. These are, for instance, used in laser-scanning-devices from

- Motivation (after Clausen et al. 2018)
    - – hard-core: R knowledge, installation, programming, etc.
    - + user friendly: OS independent, run in a browser (also locally), GUI.

https://data-analytics-lab.shinyapps.io/gui4gmwm/



Available IMUs

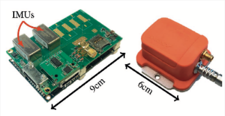


Note: more IMU data are available within "imudata" package (e.g. $6\times$ADIS16405, KVH1750).

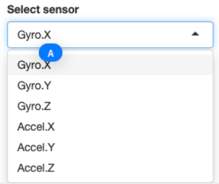
IMU types

- Tactical grade (FOG) $<$ 35'000 USD (2000): IMAR **FSAS**, Northrop Grumann (Litton) **LN200**


- MEMS $<$ 3'500 USD (2010): Thales **Navchip**-v1, XSens **MTiG**.



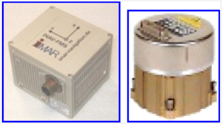
Available IMU channels



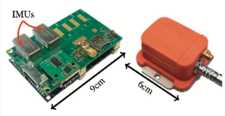
- library("imu"): 1st col=time (s)
- Gyro units: rad/s
- Accel units: m/s^2
- FSAS & LN200 at 400 Hz
- Navchip at 250 Hz
- MTiG at 100 Hz

IMU types

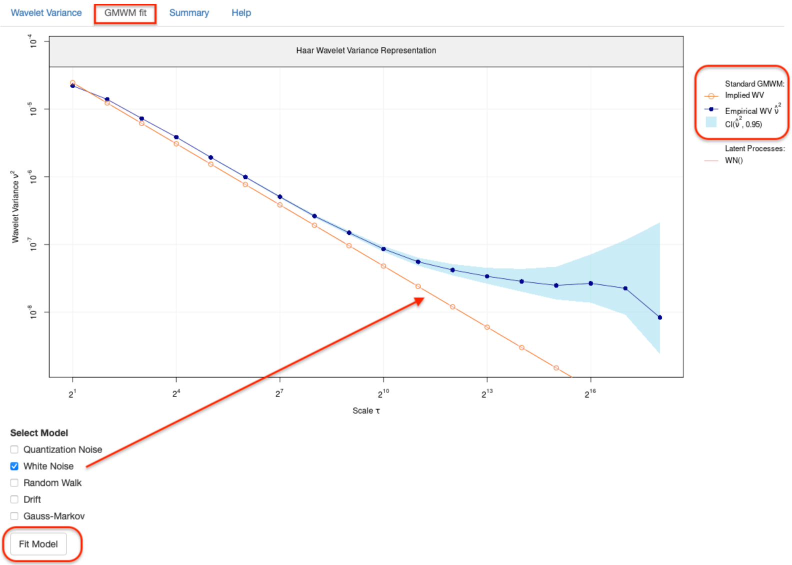
- Tactical grade (FOG) $<$ 35'000 USD (2000): IMAR **FSAS**, Northrop Grumann (Litton) **LN200**



- MEMS $<$ 3'500 USD (2010): Thales **Navchip**-v1, XSens **MTiG**.

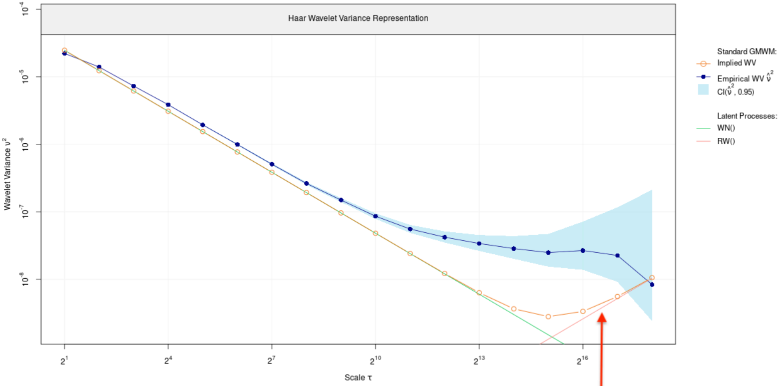


Model fit: White Noise (WN)



Model fit: White Noise (WN) + Random Walk (RW)

Wavelet Variance | GMWM fit | Summary | Help

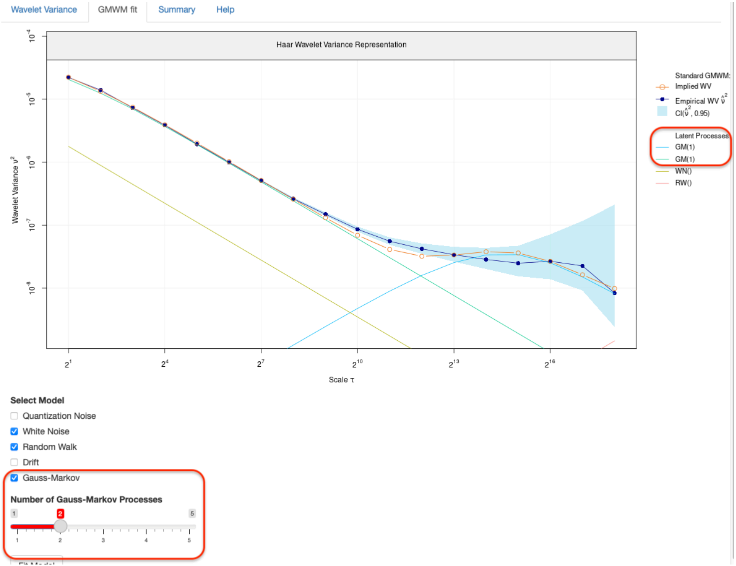


Select Model

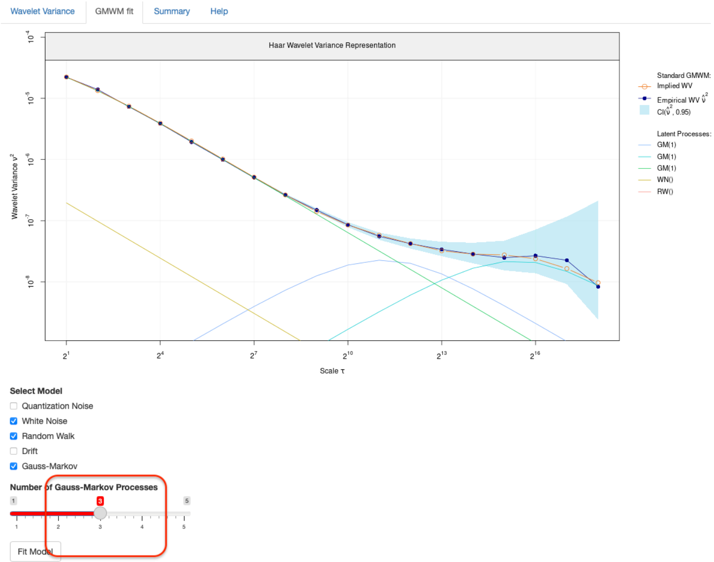
- [ ] Quantization Noise
- [x] White Noise
- [x] Random Walk
- [ ] Drift
- [ ] Gauss-Markov

Fit Model

Model fit: WN, RW, 2× 1st order Gauss-Markov (GM1)



Model fit: WN, RW, 3× 1st order Gauss-Markov (GM1)



Model fit: Objective function (WN, RW, $3\times$ GM1)

```
## Model Information:
##            Estimates
## WN      0.9851106066
## AR1     0.9009999310
## SIGMA2  0.0959828526
## RW      0.0001174386
```

- Relevant questions
    - What are the units of each parameter?
    - How to use them in estimation?

Help & local installation (gui4gmwm2)

Wavelet Variance | GMWM fit | Summary | Help

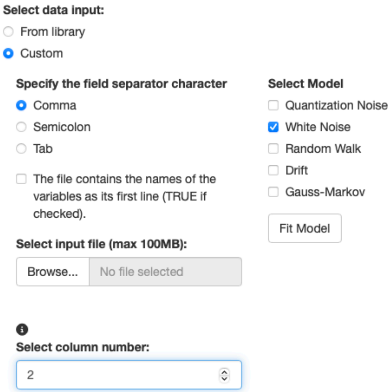
Help Tab

gui4gmwm GitHub repository: https://github.com/SMAC-Group/gui4gmwm2

Application developped by:
Stéphane Guerrier
Lionel Voirol
Philipp Clausen
Justin Lee
Roberto Molinari
Jan Skaloud

- Relevant answers
  - Partially in the `library("navigation")`
  - Partially elsewhere (see the next section).

Custom data - ascii/text file

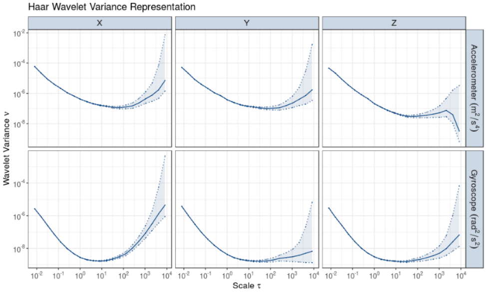


Possibilities

- different cols deliminators
- select column to analyze

Original features (gui4gmwm)

Model Data | Selected Sensor | Summary



Data

Select data input:

- (•) from library
- ( ) custom

Select IMU file:

Navchip

Selected sensor

Gyro. Y

Select estimator:

- (•) Classic WV
- ( ) Robust WV

Plot / Update WV

GMWM Modelling

Select Model

- [ ] Quantization Noise
- [x] White Noise
- [ ] Gauss-Markov
- [ ] Random Walk
- [ ] Drift

Fit Model

Reduce Model Automatically

Options

Plot options:

- [x] Process Decomp.
- [x] Show CI of empirical WV
- [ ] Overlay Datasheet Specifications

Summary options:

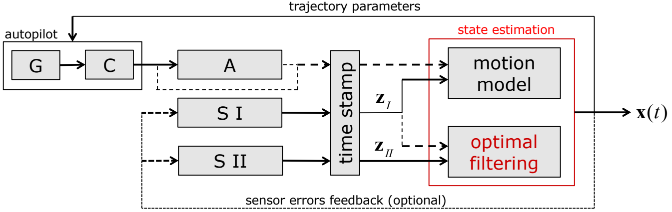
- [ ] Show CI of parameters
- [ ] Edit Optimization Parameters

Agenda

- GMWM via Web
- Unit conversion (from discrete to continuous)
- Navigation toolbox
- Projects

Where is useful stochastic and dynamic modelling?

Autonomous platform - principle



- Legend
  - G – guidance, C - control, A - actuators
  - S I – autonomous sensors (IMU, pressure, etc.)
  - S II – non-autonomous sensors (GNSS, vision, ultrasound, etc.)
- Motion model
  - Kinematic (sensor based, i.e. observing forces, rates, ...)
  - Dynamic (model based, i.e. specifying forces, rates, ...)

Where is useful stochastic and dynamic modelling?

Moving platforms - state estimation

- How?
  - Bayesian (*mostly* Kalman) filters, and/or optimal smoothers
  - Special factor graphs (dynamic networks)
- Noise modeling - via auxiliary states or parameters
  - Deterministic (biases, non-orthogonality, const. scale factors)
  - Stochastic (GMWM analysis output, θ)
- Attention in noise-parameter specification !!!
  - Sensor noise level *varies* with the choice of sensor *frequency* (f).
  - Sensor noise specifications are *usually* frequency **independent**.
  - GMWM is intended to be general w.r.t. data type (stock market, hydrology, medical data) and therefore **device/source agnostic**.

White noise (GMWM $\rightarrow$ PSD)

Convention

- "$\diamond$" denotes a **unit** of a given sensor readings, e.g. m/s^2 for an accelerometer.
- GMWM model: $x_t = \xi_t, \;\; \xi_t \sim \mathcal{N}(0, \sigma^2_{\text{GMWM}})$
- f denotes the sampling frequency in Hz (1/s).

```
## Model Information:
##         Estimates
## WN      0.9851106066
## AR1     0.9009999310
## SIGMA2  0.0959828526
## RW      0.0001174386
```

Conversion WN to continuous time (KF)

	GMWM	unit	$\rightarrow$	KF	unit (PSD)
STD	σ_{GMWM}	$\diamond$	$\sqrt{\text{WN}}/\sqrt{f}$	σ_{KF}	$\diamond/\sqrt{\text{Hz}}$
VAR	σ^2_{GMWM}	$\diamond^2$	WN/f	q_{KF}	$\diamond^2/\text{Hz}$

Random walk (GMWM $\rightarrow$ PSD)

Models

- GMWM : $x_t = x_{t-1} + \xi_t, \;\; \xi_t \sim \mathcal{N}(0, \gamma^2_{\text{GMWM}})$
- KF : $\;\; \dot{x}(t) = \xi(t)$
- if $x(t)$ is expressed in $\diamond$, thus $\dot{x}(t)$ is in $\diamond$/s and the PSD is in $\diamond^2/\text{s}^2/\text{Hz} = \diamond^2\text{Hz}$

```
## Model Information:
##             Estimates
## WN       0.9851106066
## AR1      0.9009999310
## SIGMA2   0.0959828526
## RW       0.0001174386
```

Conversion RW to continuous time (KF)

GMWM	unit	$\rightarrow$	KF	unit (PSD)
γ_{GMWM}	$\diamond$	$\sqrt{\text{RW}} \cdot \sqrt{f}$	σ_{KF}	$\diamond\sqrt{Hz} = \diamond/s/\sqrt{\text{Hz}}$
γ^2_{GMWM}	$\diamond^2$	$\text{RW} \cdot f$	q_{KF}	$\diamond^2\,\text{Hz} = \diamond^2/s^2/\text{Hz}$

1st order Gauss Markov (GMWM AR1 $\rightarrow$ KF GM1)

Models

- GMWM: $x_t = \phi_{\text{GMWM}} x_{t-1} + \xi_t, \quad \xi_t \sim \mathcal{N}(0, \sigma^2_{\text{GMWM}})$
- discrete KF: $x_{t+1} = \exp(-\beta_{\text{DISC}}/f) x_t + \eta_t, \quad \eta_t \sim \mathcal{N}(0, \sigma^2_{\text{DISC}})$

Step 1: conversion (AR1)$\rightarrow$ discrete (GM1)

$$\begin{aligned}
Var(x_{t+1}) &= Var(x_t) && (1.1)\\
&= \phi^2_{\text{GMWM}} Var(x_t) + Var(\xi_t) && (1.2)\\
&= \exp(-\beta_{\text{DISC}}/f)^2 Var(x_t) + Var(\eta_t). && (1.3)\\
\frac{\sigma^2_{\text{GMWM}}}{1-\phi^2_{\text{GMWM}}} &= \frac{\sigma^2_{\text{DISC}}}{1-\exp(-\beta_{\text{DISC}}/f)^2} && (1.4)\\
\sigma^2_{\text{GMWM}} &= \sigma^2_{\text{DISC}} && (1.5)\\
\beta_{\text{DISC}} &= -\ln(\phi_{\text{GMWM}}) \cdot f && (1.6)
\end{aligned}$$

1st order Gauss Markov (GMWM AR1 $\to$ KF GM1)

Models

- discrete KF: $x_{t+1} = \exp\left(-\beta_{\text{DISC}}/f\right)x_t + \eta_t$
- continuous KF: $\dot{x}(t) = -\beta_{\text{KF}}x(t) + \xi(t)$

Consider continuous time dynamic system of the form $\dot{x}(t) = A(t)x(t) + D(t)\xi(t)$, where $\xi(t)$ is a continuous time white noise with power spectral density q_{KF}. If we set $A(t) = -\beta_{\text{DISC}} = -\beta_{\text{KF}} = -\beta$ and $D = 1$ we find GM1 continuous model.

Step 2: conversion discrete (GM1)$\to$ continuous (GM1)

From (Bar-Shalom, Li, and Kirubarajan 2004) [Eq. 4.3.1-8] it follows that

$$
\begin{aligned}
\sigma^2_{\text{DISC}} &= q_{\text{KF}} \int_0^{\Delta t} \exp[-2\beta(\Delta t - \tau)]d\tau \\
&= q_{\text{KF}} \frac{1 - \exp(-2\beta/f)}{2\beta}. \qquad (1.7)
\end{aligned}
$$

1st order Gauss Markov (GMWM AR1 $\rightarrow$ KF GM1)

Models

- GMWM $x_t = \phi_{\text{GMWM}} x_{t-1} + \xi_t, \;\; \xi_t \sim \mathcal{N}(0, \sigma^2_{\text{GMWM}})$
- KF: $\dot{x}(t) = -\beta_{\text{KF}} x(t) + \xi(t)$
- Steps: GMWM 1st order auto-regressive (AR1) $\rightarrow$ discrete GM1 $\rightarrow$ continuous GM1

```
## Model Information:
##               Estimates
## WN         0.9851106066
## AR1        0.9009999310
## SIGMA2     0.0959828526
## RW         0.0001174386
```

Conversion AR1 to continuous time (KF)

GMWM	unit	$\rightarrow$	KF	unit
ϕ_{GMWM}	–	$-\ln(\text{AR1}) \cdot f$	$\beta_{\text{KF}} = 1/\text{T}_{\text{KF}}$[a]	$\text{Hz} = 1/s$
σ^2_{GMWM}	$\diamond^2$	$\frac{2\,\beta_{\text{KF}} \cdot \text{SIGMA2}}{1-\exp(-2\beta_{\text{KF}}/f}$)	q_{KF}	$\diamond^2/s^2/\text{Hz}$
σ_{GMWM}	$\diamond$	$\sqrt{q_{\text{KF}}}$	σ_{KF}	$\diamond/s/\sqrt{\text{Hz}}$

Recap - GMWM Unit Conversion

Take home message

- GMWM output depends on input.
- Conversion is needed for estimation.
- Use the above mentioned formulas together with the knowledge of (i) sampling frequency f, (ii) units used for GMWM analysis.
- What about quantization noise?

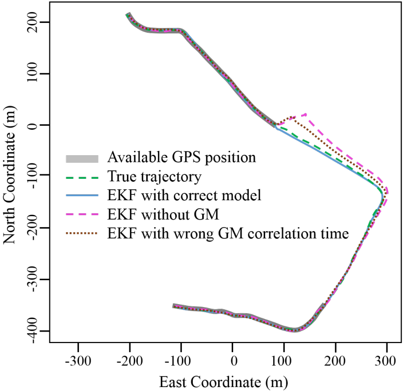
Agenda

- GMWM via Web
- Unit conversion (from discrete to continuous)
- Navigation toolbox
- Projects

Former real-case: Impact of sensor model on trajectory

Auto-motive example

- Different stoch. models are used to describe sensor (random) errors in *accelerometers* and *gyroscopes* within an inertial system (INS) that is integrated with satellite positioning.
- The realization of *time correlated random errors* in the sensors is estimated by a navigation filter and subtracted via a feedback.
- In the absence of satellite signals the trajectory is entirely based on INS, which performance depends partly on sensor models.



After Clausen et al. 2018.

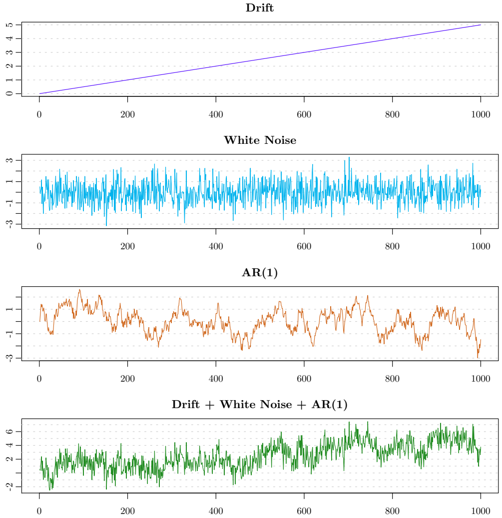
UAV experiment - impact

Interactive demo after Khaghani et al. 2019 - later in course (Day 4).

Employs R package navigation:

1. Uses real trajectory.
2. Emulates noise on sensors along this trajectory.
3. Evaluate an impact, possibly compares models.

Impact of sensor noise on navigation application



Approach

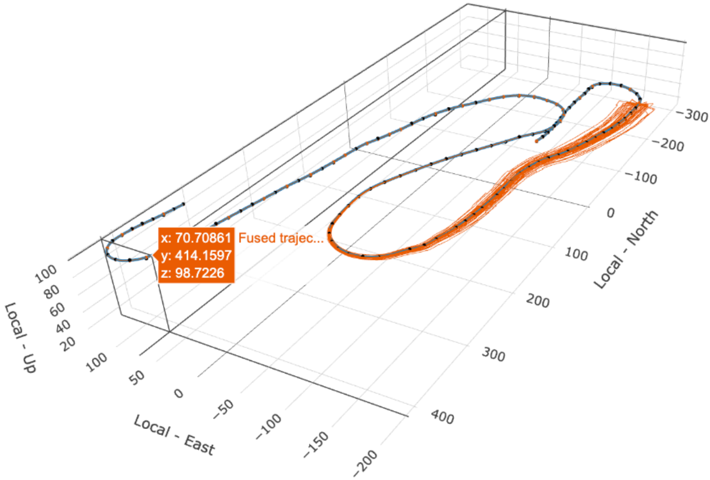
- GMWM can estimate the noise structure & parameter values.
- The noise can be *emulated* on a real sensors.
- R "navigation" package analyses the *effect of emulated inertial noise* w.r.t sensor fusion *models* on a chosen trajectory.

UAV experiment - example_navigation.R

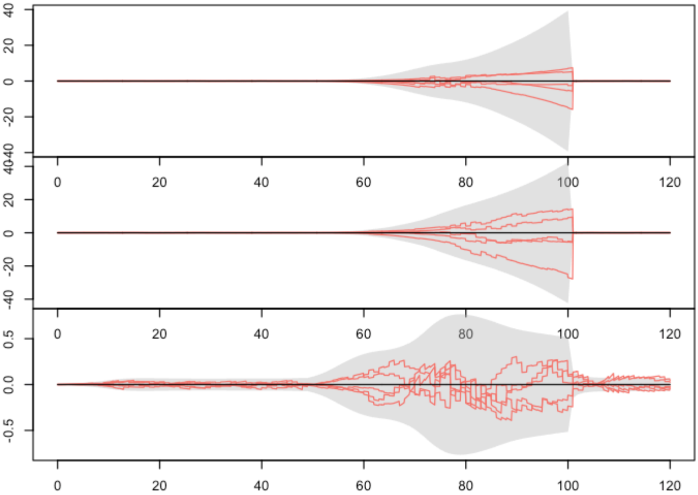
Preamble

1. install.packages("navigation")
2. obtain example_navigation.R from github/examples.
3. add library(devtool)
4. add library(simts)
5. execute the R script (Ctrl + Shift + Enter)

Impact of noise (WN + GM1) - traj. realisation

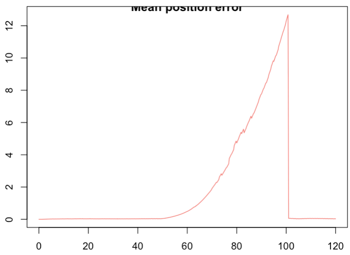


Impact of noise (WN + GM1) - East-North-Up Errors



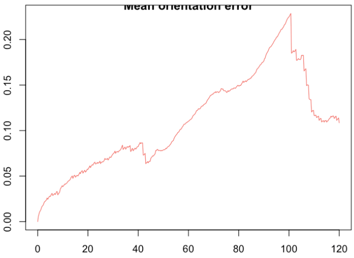
Impact of noise (WN + GM1) - Mean Errors

Mean position error [m]



time (s)

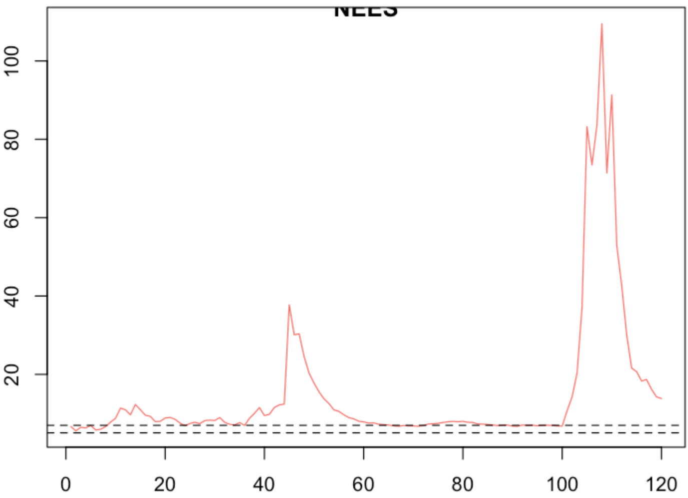
Mean orientation error [deg]



time (s)

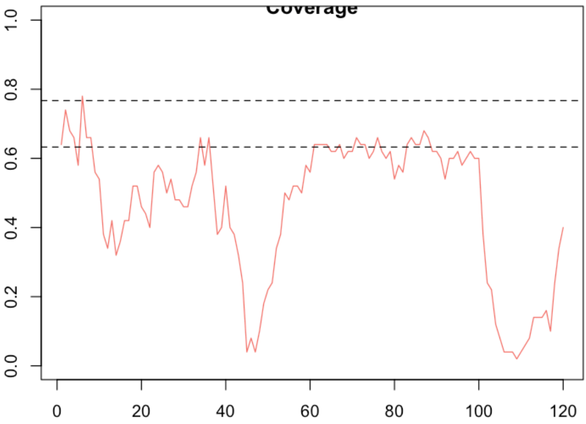
Normalized Estimation Error Squared (NEES) [m]

NNES - Consistency of error vs. predicted confidence levels.



Impact of noise - Coverage (1.0=100%)

Measures how often the confidence intervals cover the true trajectory.



An approximate 95% confidence interval for the averaged (i)ct over N Monte Carlo simulation is given by: $(1-\alpha) \pm 1.96\sqrt{\frac{\alpha(1-\alpha)}{N}} \approx [66\%, 74\%]$

Other example - `example_compare_navigation.R`

Impact on sensor fusion

- Compares sensor fusion (traj. estimation) with correct vs. incorect (IMU) sensor noise models.
- Synthetic trajectory (lemniscate).
- Details in publication.

Current limits of "navigation" package

- Initialization errors are not considered, so are time-constant random biases or scale.
- Other simplifications are absences of (non-orthogonality) and influence of gravitational anomalous field.
- For certain sensors and situations the above mentioned errors can mask or be higher than the effects of time-correlated stochastic errors.

References I

Bar-Shalom, Yaakov, X Rong Li, and Thiagalingam Kirubarajan (2004). *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons.

Clausen, P. et al. (2018). "Use of a new online calibration platform with applications to inertial sensors". In: *IEEE AEROSPACE AND ELECTRONIC SYSTEMS MAGAZINE* August, pp. 30–36. DOI: http://dx.doi.org/10.1109/MAES.2018.170153.

Khaghani, M. et al. (2019). "Optimal stochastic sensor error modeling based on actual impact on quality of GNSS-INS integrated navigation". In: *ION GNSS+*. URL: https://www.ion.org/publications/abstract.cfm?articleID=17057.