GMWM Package - Part II

Automatic and Computationally Efficient Method For Model Selection In Inertial Sensor Calibration

Roberto Molinari

Research Center for Statistics
University of Geneva, Switzerland

joint work with
James Balamuta (UIUC)
Stéphane Guerrier (UIUC)
Jan Skaloud (EPFL)
Xinyu Zhang (CAS)

September 16, 2015

The GMWM package in action

Outline

1. **Visualizing the signal(s)**:
   - Is the signal disturbed (contaminated)?
   - Which models could best describe the visualized WV?
2. **Estimating the model(s)**
   - The `gmwm.imu()` function for parameter estimation
   - The options for estimation
3. **Inference**
   - Confidence intervals for the parameters
   - Goodness-of-fit of the model to the signal
4. **Model Selection**
   - The efficient computation of the WVIC
   - Model selection from a set of user-specified models
   - Automatic model selection from a set of all sub-models of a single user-specified model

The GMWM package in action

Getting to know the package

The package comes with an example dataset called `imu6`

- The data comes from the calibration of an XSens MTi-G sensor
- To use the data within the `R` session:
  1. Load the `gmwmdata` package (separate from the `gmwm` package)
  2. Make the dataset available in the session by typing `data(imu)`
- Load external data (e.g. binary data) with `read.imu()`

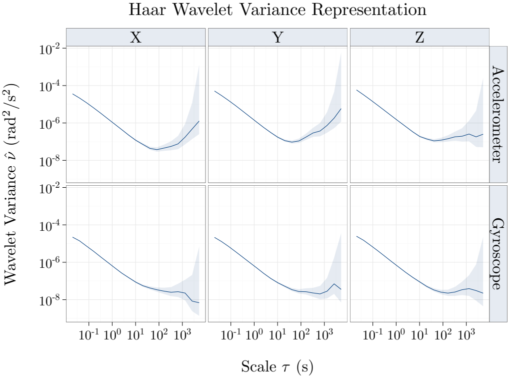
```
install.packages("gmwmdata")
library(gmwmdata)
data(imu6)
head(imu6)
```

	Gyro. X	Gyro. Y	Gyro. Z	Accel. X	Accel. Y	Accel. Z
1	−0.011138	−0.017646	−0.009531	1.083731	0.023897	9.638113
2	−0.006765	−0.013657	−0.021974	1.059858	0.042061	9.646901
3	...	...	...	...	...	...

Visualizing the data

The function `wvar.imu()`

Once the calibration data has been entered, it is possible to plot the observed WV vs τ on a log-log scale



R code:

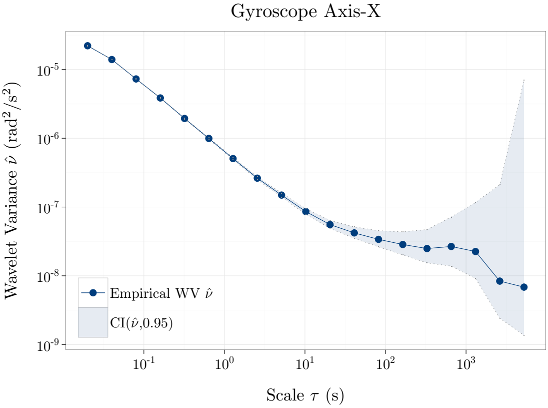
```
imu.obj = imu(imu6,
gyroscope = 1:3,
accelerometer = 4:6)
wv = wvar.imu(imu.obj)
plot(wv)
```

(Run time: 3.70 [sec])

Visualizing the data

The function `wvar()`

It is obviously possible to plot the WV from the axis of an accelerometer or gyroscope individually



R code:

```
WV.gx = wvar(imu6[,1])
plot(WV.gx)
```

(Run time: 0.65 [sec])

The Allan variance can also be computed through the function `avar()`

GMWM estimation

The `gmwm.imu()` function

The function tailor-made for IMU error modeling is `gmwm.imu()`

Main arguments of the `gmwm.imu()` function

- `model`: The structure of the model is specified through this argument
- `data`: The signal to be modelled
- `compute.v`: The method for computing the weighting matrix $\boldsymbol{\Omega}$
  - `fast`: Estimated diagonal matrix
  - `diag`: Use a diagonal matrix with the asymptotic variance of $\hat{\boldsymbol{\nu}}$
  - `bootstrap`: Estimate $\boldsymbol{\Omega}$ through parametric bootsrap

GMWM estimation

The `model` argument

The GMWM package can estimate each specific model or any latent model made by a combination of all or a subset of the following models

- `AR1()`: a first-order autoregressive process (reparametrization of Gauss-Markov process)
- `WN()`: white noise process
- `QN()`: quantization noise (rounding error)
- `RW()`: random walk process
- `DR()`: drift
- `AR()`: p-order autoregressive process
- `MA()`: q-order moving average process
- `ARMA()`: autoregressive-moving average processes

GMWM estimation

Latent model syntax

To specify a latent model we use the sign "+" between the available models while the `AR1()` model can be included more than once (say k times) and is can be specified as `k*AR1()`, for example

- `ARMA()+WN()+RW()`
- `3*AR1()+DR()`

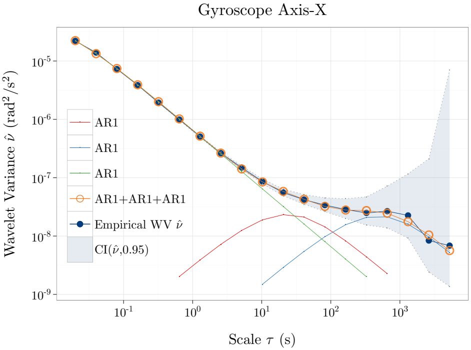
Convergence

If the `gmwm.imu()` function has problems of convergence, one can specify **starting values** for the parameters using the brackets in the syntax of each model (e.g. `WN(sigma2=0.5)`)

GMWM estimation

Visually assessing the fit

The function `plot()` applied to the object of a GMWM estimation allows to see how well the WV implied by the estimated model fits the observed WV



R code:

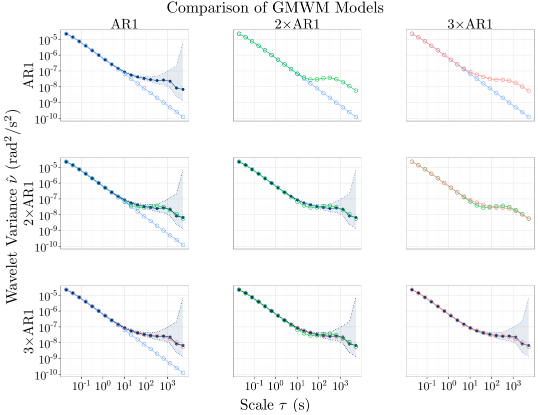
```
mod = gmwm.imu(3*AR1(),
imu6[,1]) plot(mod,
process.decomp = T)
```

(Run time: 0.56 [sec])

Model Comparison

Suppose we want to compare three models...

AR1(), 2*AR1() and 3*AR1()



R code:

```
Xt = imu[,1]
m1 = gmwm.imu(AR1(),Xt)
m2 = gmwm.imu(2*AR1(),Xt)
m3 = gmwm.imu(3*AR1(),Xt)
compare.models(m1, m2, m3)
```

(Run time: 5.20 [sec])

Inference

Output of estimation

Aside from the value of the estimated parameters, the output of the function `gmwm.imu` provides other information for inference which include **confidence intervals** and **goodness of fit** test (GoF)

```
mod = gmwm.imu(2*AR1(), imu6[,1])
summary(mod, inference = T)
```

```
Model Information:
          Estimates        CI Low        CI High            SE
AR1    9.998700e-01 9.998336e-01 9.999064e-01 2.211547e-05
SIGMA2 5.223319e-11 4.343878e-11 6.102759e-11 5.346620e-12
AR1    1.265324e-01 1.265324e-01 1.265324e-01 1.765050e-08
SIGMA2 5.031185e-05 5.021566e-05 5.040805e-05 5.848209e-08

* The initial values of the parameters used in the minimization
of the GMWM objective function were  generated by the program
underneath seed: 1337.

Objective Function: 20.9297

Asymptotic Goodness of Fit:
Test Statistic: 903.8 on 15 degrees of freedom
The resulting p-value is: 0
```

WIC model selection

Two options for model selection

- **Manual**: the function `rank.models()` allows the user to specify the set of models from which to select
- **Automatic**: the function `auto.imu()` allows to specify one single model from which all sub-models are generated to create a set from which to select

R code:

```
rank.models(3*AR1()+WN(),
2*AR1()+QN(), imu6[,1], nested =
F, bootstrap = F, model.type="imu",
robust = F)
```

```
mod.res = auto.imu(imu6, model
= 3*AR1()+WN()+RW()+QN()+DR(),
bootstrap = F, robust = F)
```

WIC model selection

Automatic IMU model selection

Suppose we want to apply the `auto.imu()` function to the X-axis gyroscope in the `imu` dataset and consider all model combinations within the `4*AR1() + WN() + RW()` model

```
Xt = imu6[,1]
mod = 4*AR1()+WN()+RW()
mod.sel = auto.imu(Xt, model = mod)
summary(mod.sel)
```

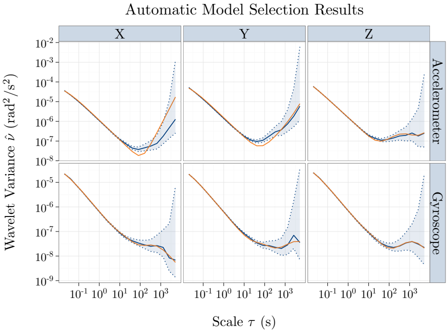
```
The model ranking for data column 1:
                              Obj Fun  Optimism   Criterion  GoF P-Value
1. AR1 AR1 AR1                17.2769    0.2974     17.5743            0
2. AR1 AR1 AR1 RW             17.2679    0.3139     17.5818            0
3. AR1 AR1 AR1 AR1            17.6022    0.4330     18.0351            0
4. AR1 AR1 RW                 20.1044    0.4871     20.5915            0
5. AR1 AR1                    20.9864    0.2070     21.1933            0
...
19. RW                     28420.4972    0.0160  28420.5132            0
```

(Run time: 50.25 [sec] ≈ 14x faster than the bootstrap option 685.4 [sec])

WIC Model Selection

Visualizing the Automatic IMU Model Selection

We can observe the results of the `auto.imu()` function by requesting a plot. The plot will contain the empirical wavelet variance in addition to the best implied wavelet variance that we find. In this particular case, we used the defaults of the `auto.imu()` (63 models $\times$ 6 columns).



R code:

```
imu.obj = imu(imu6, gyroscope = 1:3,
accelerometer = 4:6, axis = c('X',
'Y', 'Z'))
auto.mod = auto.imu(imu.obj)
plot(auto.mod)
```

(Run time: 377.77 [sec])

Upcoming features

Aside from **fine-tuning existing features** of the package (additional options, function optimization, support documentation, etc.), the GMWM package will be updated with the following features

- Sensor calibration under **dynamic conditions**

$$Y_t = \boldsymbol{\rho_t} Y_{t-1} + \epsilon_t$$

- **Multivariate latent models**

$$\begin{aligned} Y_t^{(x)} &= V_t + W_t^{(x)} \\ Y_t^{(y)} &= V_t + W_t^{(y)} \\ Y_t^{(z)} &= V_t + W_t^{(z)} \end{aligned}$$

- Others (e.g. additional moments for the GMWM (AGMWM), other wavelet filters, unit-root and independence testing)

Thank you very much for your attention!

A special thanks to

- Dr. Yannick Stebler (EPFL/u-blox)
- Prof. Maria-Pia Victoria-Feser (U. Geneva)
- Wenchao Yang (UIUC)

Any questions?

More info...

SMAC-group.com
github.com/SMAC-Group
stephane@illinois.edu