GPoM : VI Approach sensitivity

Sylvain Mangiarotti & Mireille Huc

2018-01-18

Approach sensitivity

One important interest of the GPoM package comes from its potential to tackle chaotic dynamics (that is, deterministic behaviors presenting a high sensitivity to initial conditions). This ability can be of first importance when dealing with biophysical and environmental systems, and more generally with most of real world behaviors because most of these dynamics are (or can be assumed to be) governed by deterministic processes, and, at the same time, are found to be poorly predictable at long term. To model such type of dynamics is a difficult task because it requires using an approach which formulation can guaranty the independence to the initial conditions. Furthermore, to model real dynamical behaviors, the approach should also allow working with observational time series, that is, with data contaminated by noise, with short time series, with data which behavior may be perturbed by external forcing, subsampled measurements, etc.

The aim of this vignette is to give some insights of some of these difficulties. To use a theoretical case study can be very useful for this purpose in order to keep a full control on the experiments, and also to enable to consider each source of difficulty, one by one, independently. For the sake of consistency, the same chaotic system (the Rössler system) is used for all the experiments considered in the present vignette. The sensitivity to system algebraic structure will be addressed in another vignette (`VII_Retromodelling`).

Sensitivity to the initial conditions

To illustrate the potential of the GPoM package to deal with time series independently to initial conditions, the global modelling technique is applied to two data sets presenting completely different initial conditions. The two sets of time series have been generated by the same dynamical system (see vignette `I_Generate`). The global modelling technique is then applied (see vignette `III_Modelling`) to check if the original model is retrieved in both cases.

The original system and data

The Rössler system is taken as case study for the present experimentations. For $(a = 0.52, b = 2, c = 4)$, this system can be described by the matrix `K` as follows:

```
# parameters
a = 0.52
b = 2
c = 4
# equations
Eq1 <- c(0,-1, 0,-1, 0, 0, 0, 0, 0, 0)
Eq2 <- c(0, 0, 0, a, 0, 0, 1, 0, 0, 0)
Eq3 <- c(b,-c, 0, 0, 0, 0, 0, 1, 0, 0)
K = cbind(Eq1, Eq2, Eq3)
visuEq(3, 2, K, substit = c("x", "y", "z"))
```

```
## dx/dt = -z  -y
##
## dy/dt = 0.52 y  + x
##
```

```
## dz/dt = 2  -4 z  + x z
```

Two sets of initial conditions will be used from which two simulations are obtained.

```
# equations
v1 <- c(0.6, -0.6, 0.4)
v2 <- c(-4, 2, 0.)
nVar = 3
dMax = 2
outNumi1 <- numicano(nVar, dMax, Istep = 2000, onestep = 1/50, KL = K, v0 = v1)
outNumi2 <- numicano(nVar, dMax, Istep = 2000, onestep = 1/50, KL = K, v0 = v2)
plot(outNumi2$reconstr[,1], outNumi2$reconstr[,2], type = 'l', xlab = 't', ylab = 'x(t)')
lines(outNumi1$reconstr[,1], outNumi1$reconstr[,2], type = 'l', col = 'red')
legend(3,-2.5, c("i.c. v1", "i.c. v2"), col=c('red', 'black'), lty=1, cex = 0.8)
```
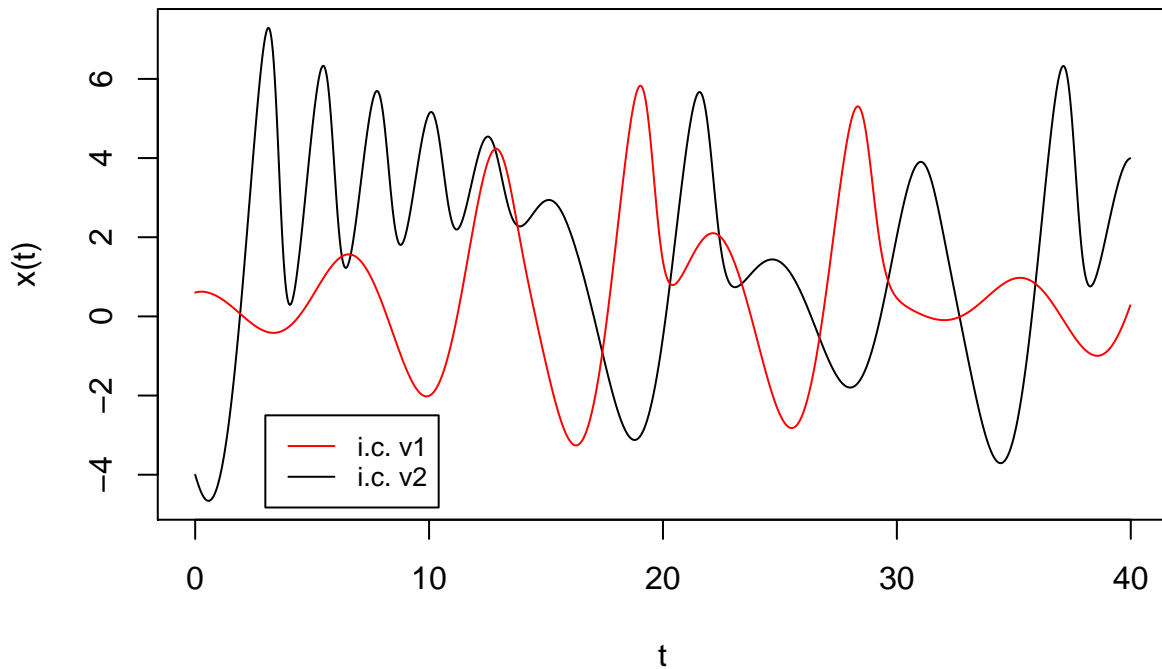

The phase portraits of the two data sets are presented hereafter:

```
plot(outNumi2$reconstr[,2], outNumi2$reconstr[,3], type = 'l', xlab = 'x(t)', ylab = 'y(t)')
lines(outNumi1$reconstr[,2], outNumi1$reconstr[,3], type = 'l', col = 'red')
lines(outNumi2$reconstr[1:400,2], outNumi2$reconstr[1:400,3], type = 'p', col = 'orange', cex = 0.6)
# initial condition
lines(outNumi2$reconstr[1,2], outNumi2$reconstr[1,3], type = 'p', lwd = 3)
lines(outNumi1$reconstr[1,2], outNumi1$reconstr[1,3], type = 'p', col = 'red', lwd = 3)
legend(4,2, c("i.c. v1", "i.c. v2", "transient v2"), col=c('red', 'black', 'orange'),
       lty=c(1,1, NA), pch = c(NA,NA,1), cex = 0.6)
```
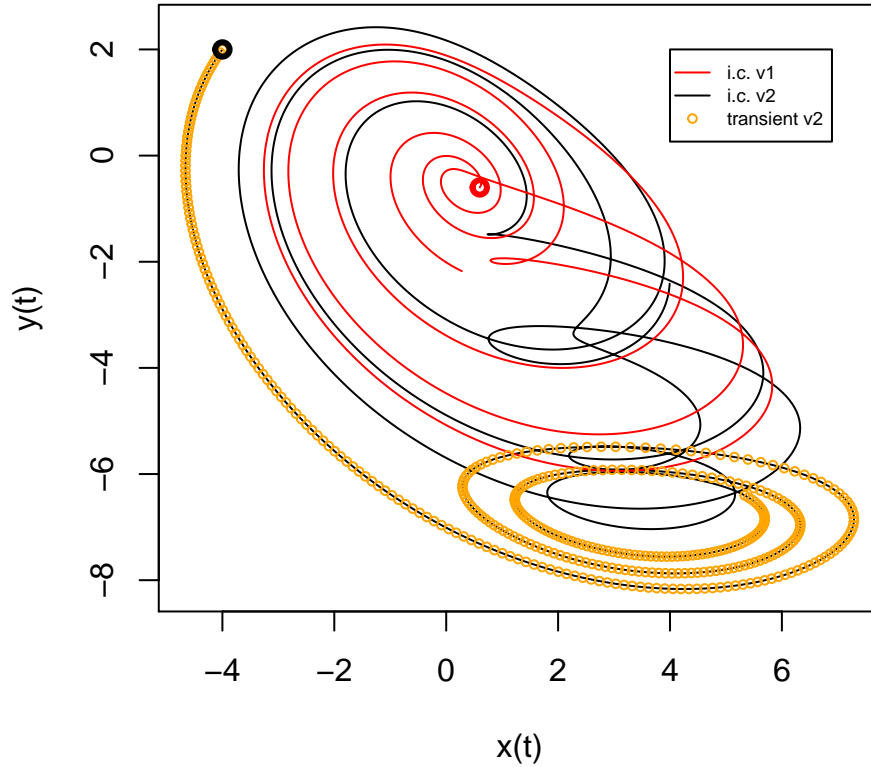
The trajectory in black is issued from the initial condition `v2` (black circle), it starts with a transient (in orange) before reaching the attractor. The trajectory in red is issued from the other initial condition `v1` (red circle) and has already converged to the attractor.

The global modelling technique is applied to these two sets of time series, from which it is expected to retrieve the original Rössler system.

```
# time vector
tin <- outNumi1$reconstr[,1]
# modelling from time series starting from initial condition v1
out1 <- gPoMo(data = outNumi1$reconstr[,2:4], tin = tin, dMax = 2, nS = c(1,1,1), show = 0,
              IstepMin = 10, IstepMax = 1500, nPmin = 7, nPmax = 7, method = 'rk4')
```

```
## ### For Istep = 10 (max: 1500), models to test: 15 / 15
## ### For Istep = 20 (max: 1500), models to test: 15 / 15. Runtime: ~ 0h 0min 0.25s
## ### For Istep = 40 (max: 1500), models to test: 15 / 15. Runtime: ~ 0h 0min 0.47s
## ### For Istep = 80 (max: 1500), models to test: 15 / 15. Runtime: ~ 0h 0min 0.91s
## ### For Istep = 160 (max: 1500), models to test: 15 / 15. Runtime: ~ 0h 0min 1.8s
## ### For Istep = 320 (max: 1500), models to test: 15 / 15. Runtime: ~ 0h 0min 3.5s
## ### For Istep = 640 (max: 1500), models to test: 6 / 15. Runtime: ~ 0h 0min 2.82s
## ### For Istep = 1280 (max: 1500), models to test: 6 / 15. Runtime: ~ 0h 0min 5.66s
## ### Number of unclassified models: 6 / 15
```

```
# visualize all the models obtained from i.c. v1
visuOutGP(out1)
# visualize all the models obtained from i.c. v2
visuOutGP(out2)
```

3

Model selection

When applying the model search, models can be categorized as (1) diverging trajectories (these will be directly rejected), (2) fixed points (these will be kept aside once categorized), (3) period-1 cycles or (4) not identified. In the present version of the package, the obtained models may belong either to (3) or to (4), this latter being the most interesting category for us because it can include all the non-trivial solutions (categories (2) or (3) may also present some interest in the case that converging solution or period cycles should be expected). Depending on the quality of the measured time series, but also on the observability of the system, none, few or many models may be obtained using GPoM. A selection criterion will have to be used if several models are obtained. In practice, the models characterized by a phase portrait similar to the original phase portrait should be preferred. To do so, a selection based on a visual checking among the obtained models is found to be very efficient and generally quite obvious. The more terms being used in the model, the better the model should be. However, a high number of terms may also foster numerical instabilities. As a consequence, models with less algebraic terms should be preferred as far as they enable a better consistency with the original data set. For this reason, our selection will systematically foster the simplest model formulations among the obviously 'good' models.

Results

Based on these criteria, the smallest obtained model able to reproduce the original phase portrait is found to be the model #5 in both cases. Obtained equations respectively read

```
# Edit the equations of model obtained from time series generated from v1
visuEq(nVar = 3, dMax = 2, out1$models$model5, substit = c("x", "y", "z"), approx = 2)
```

```
## dx/dt = -0.998 z  -0.998 y
##
## dy/dt = 0.519 y  + 0.999 x
##
## dz/dt = 1.986  -3.971 z  + 0.993 x z
```

for the first data set, and for the other one

```
# Edit the equations of model obtained from time series generated from v2
visuEq(nVar = 3, dMax = 2, out2$models$model5, substit = c("x", "y", "z"), approx = 2)
```

```
## dx/dt = -0.998 z  -0.998 y
##
## dy/dt = 0.519 y  + 0.999 x
##
## dz/dt = 1.986  -3.971 z  + 0.993 x z
```

The original equations of the Rössler system can thus be retrieved in both cases with data sets of different initial conditions (and even with very different trajectories, the first one being based on the transient only).

The error on the model coefficients varies from 0.1 % to 0.7 % for the observational time series located close to the attractor. The level of error doubles more or less when using the transient as input.

```
# min values
min(abs((out1$models$model5 - K) / K) * 100, na.rm = TRUE)
min(abs((out2$models$model5 - K) / K) * 100, na.rm = TRUE)
min(abs((out3$models$model5 - K) / K) * 100, na.rm = TRUE)
# max values
max(abs((out1$models$model5 - K) / K) * 100, na.rm = TRUE)
max(abs((out2$models$model5 - K) / K) * 100, na.rm = TRUE)
max(abs((out3$models$model5 - K) / K) * 100, na.rm = TRUE)
```

The approach is thus able to retrieve the exact equations form, independently to the initial conditions. Despite a rather high level of precision (the error is lower than 1%), parameter identification remains slightly dependent to the initial conditions.

## Sensitivity to signal length

Data

To test the sensitivity to the signal length, eight time series of decreasing length are considered in this experiment (small shifts have been applied to the plots along the vertical abscissa to help their visusalization).

```
plot(outNumi1$reconstr[,1], outNumi1$reconstr[,2], type = 'l', xlab = 't', ylab = 'x(t)', lwd = 3)
lines(outNumi1$reconstr[1:1000,1], outNumi1$reconstr[1:1000,2]+0.25, col = 2, lwd = 3)
lines(outNumi1$reconstr[1:900,1], outNumi1$reconstr[1:900,2]+0.5, col = 3, lwd = 3)
lines(outNumi1$reconstr[1:800,1], outNumi1$reconstr[1:800,2]+1., col = 4, lwd = 3)
lines(outNumi1$reconstr[1:700,1], outNumi1$reconstr[1:700,2]+1.25, col = 5, lwd = 3)
```
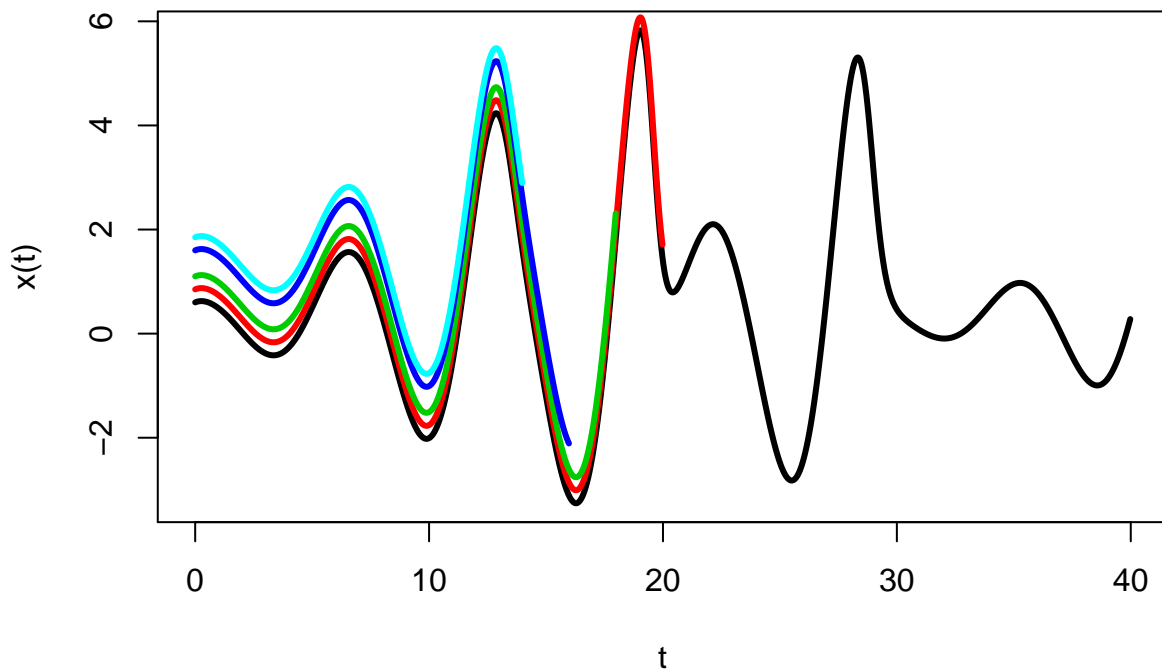

Corresponding phase portaits:

```
plot(outNumi1$reconstr[,2], outNumi1$reconstr[,3], type = 'l',
     xlab = 'x(t)', ylab = 'y(t)', lwd = 3)
lines(outNumi1$reconstr[1:1000,2], outNumi1$reconstr[1:1000,3], col = 2, lwd = 3)
lines(outNumi1$reconstr[1:900,2], outNumi1$reconstr[1:900,3], col = 3, lwd = 3)
lines(outNumi1$reconstr[1:800,2], outNumi1$reconstr[1:800,3], col = 4, lwd = 3)
lines(outNumi1$reconstr[1:700,2], outNumi1$reconstr[1:700,3], col = 5, lwd = 3)
```
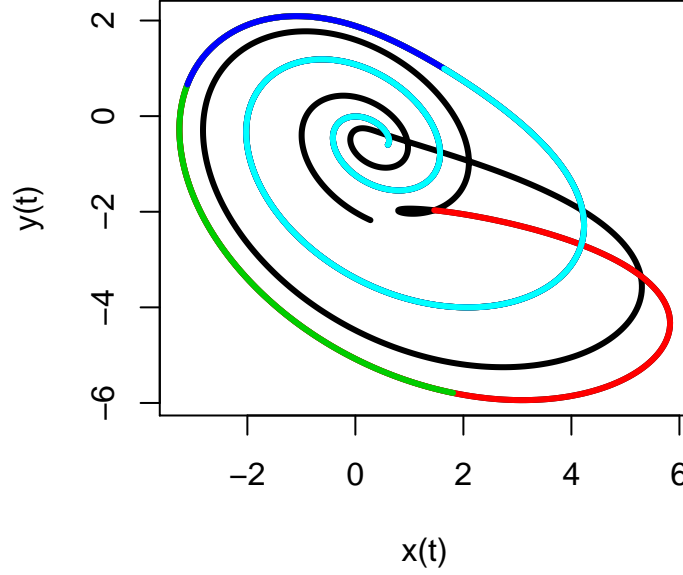
Global modelling

The global modelling is applied to the corresponding time series of varying length, each time considering three time series corresponding to the three variables x, y and z of the Rössler system.

The original formulation is retrieved in all the cases. As an example, the equations obtained with the shortest time series are the following:

```
# visualize all the models obtained from i.c. v1
visuOutGP(out7)
```

For shorter time series, it becomes more difficult to detect the model.

```
# min values
min(abs((out1$models$model5 - K) / K) * 100, na.rm = TRUE)
min(abs((out4$models$model5 - K) / K) * 100, na.rm = TRUE)
min(abs((out5$models$model5 - K) / K) * 100, na.rm = TRUE)
min(abs((out6$models$model5 - K) / K) * 100, na.rm = TRUE)
min(abs((out7$models$model5 - K) / K) * 100, na.rm = TRUE)
# max values
max(abs((out1$models$model5 - K) / K) * 100, na.rm = TRUE)
max(abs((out4$models$model5 - K) / K) * 100, na.rm = TRUE)
max(abs((out5$models$model5 - K) / K) * 100, na.rm = TRUE)
max(abs((out6$models$model5 - K) / K) * 100, na.rm = TRUE)
max(abs((out7$models$model5 - K) / K) * 100, na.rm = TRUE)
```

In the present case, the signal length variations do not lead to significant differences in the precision of the parameters: levels of error remains between 0.07 % and 1.0 %.

Sensitivity to subsampling and resampling

It was shown that subsampling could have a very big influence on the phase space reconstruction[1] (see also vignette `II_Preprocessing`). In the present section, we will investigate the possibility to retrieve the

---

[1]S. Mangiarotti, 2018. The global modelling classification technique applied to the detection of chaotic attractors. *Supplementary Material A* to "Can the global modelling technique be used for crop classification?" by S. Mangiarotti, A.K. Sharma, S. Corgne, L. Hubert-Moy, L. Ruiz, M. Sekhar, Y. Kerr, 2018. *Chaos, Solitons & Fractals*, **106**, 363-378.

original Rössler model from the subsampled and resampled time series.

Data

Original (in gray), subsampled (in red) and resampled (in green) time series are presented hereafter:

```r
plot(outNumi1$reconstr[,1], outNumi1$reconstr[,2],
     type = 'l', col = 'gray', xlab = 't', ylab = 'x(t)', lwd = 3)
# subsample
sub1 <- 50
tsub1 <- outNumi1$reconstr[seq(1, 2000, by = sub1),][,1]
datsub1 <- outNumi1$reconstr[seq(1, 2000, by = sub1),][,2:4]
lines(tsub1, datsub1[,1], type = 'p', col = 'red', lwd = 3)
# resample
xout <- seq(min(tsub1), max(tsub1), by = 0.01)
rspl1x <- spline(tsub1, y = datsub1[,1], method = "fmm", xout = xout)
rspl1y <- spline(tsub1, y = datsub1[,2], method = "fmm", xout = xout)
rspl1z <- spline(tsub1, y = datsub1[,3], method = "fmm", xout = xout)
lines(rspl1x$x, rspl1x$y, type='l', col='green')
legend(0, 5, c("original", "subsampled", "resampled"), col=c('gray', 'red', 'green'),
       lty=c(1, NA, 1), pch = c(NA,1,NA), cex = 0.8)
```
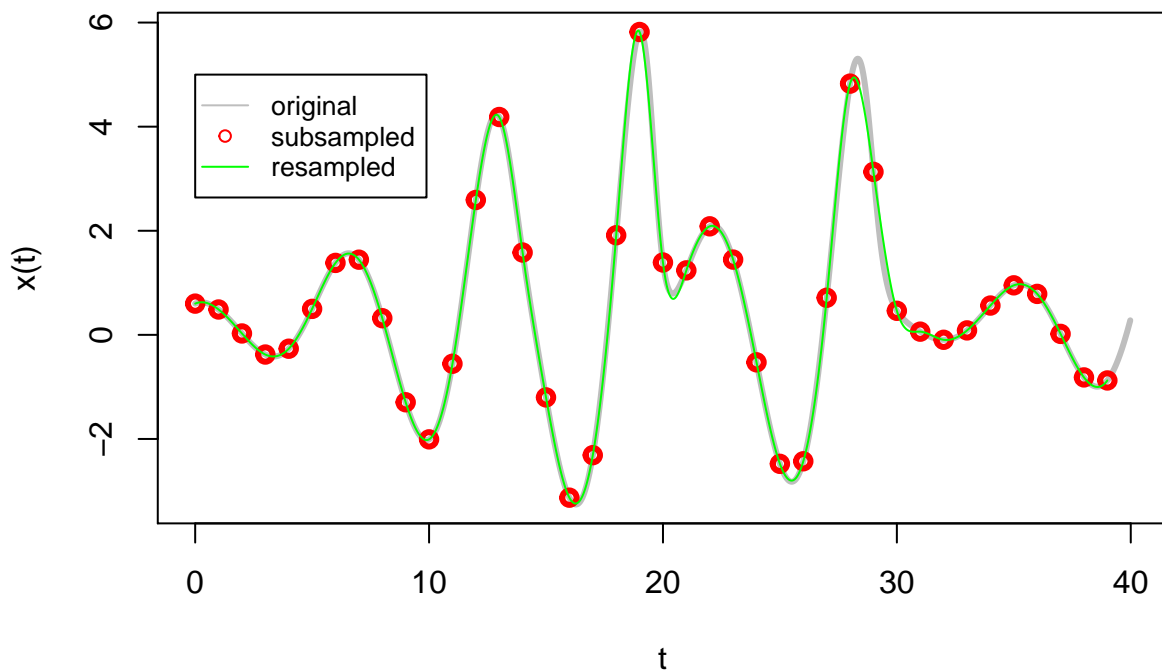

Subsampled time series

The global modelling was first applied to the subsampled data set.

```r
# modelling using half the initial time series
out8 <- gPoMo(data = datsub1, tin = tsub1, dMax = 2, nS = c(1,1,1), show = 0,
              IstepMin = 10, IstepMax = 400, nPmin = 2, nPmax = 12, method = 'rk4')
```

None of the models enabled to reproduce the original phase portrait satisfyingly: the original model could not be retrieved from this subsampled data set, even partially.

Resampled time series

The global modelling was then applied to the resampled data set:

```
# modelling using half the initial time series
out9 <- gPoMo(data = cbind(rspl1x$y, rspl1y$y, rspl1z$y), tin = rspl1x$x,
             dMax = 2, nS = c(1,1,1), show = 0, IstepMin = 10, IstepMax = 400,
             nPmin = 2, nPmax = 12, method = 'rk4')
```

The simplest model presenting an appropriate phase portrait is the model #44.

```
# model #44
visuOutGP(out9, selecmod = 44, prioMinMax = "model")
```

```
# Equations
visuEq(nVar = 3, dMax = 2, out9$models$model44, substit = 1, approx = 4)
```

```
## dx/dt = -0.82517 z  -1.06898 y  + 0.11477 y z
##
## dy/dt = 0.51901 y  + 0.99557 x
##
## dz/dt = 0.97736  -2.11703 z  + 0.5947 x z
```

This model has one more algebraic term than the original system. This means that interpretation should be considered with more caution when considering time series under subsampled conditions. One important source of error in the case of resampled time series can result from the computation of derivatives. It may thus be interesting to investigate the possibility to obtain simpler formulation by tuning the window length used to compute the derivatives as follows:

```
# modelling using half the initial time series
out10 <- gPoMo(data = cbind(rspl1x$y, rspl1y$y, rspl1z$y), tin = rspl1x$x,
              dMax = 2, nS = c(1,1,1), show = 0, winL = 13,
              IstepMin = 10, IstepMax = 400, nPmin = 2, nPmax = 7,
              method = 'rk4')
```

Using a slightly stronger smoothing (`winL = 13` instead of `winL = 9` by default), a simpler formulation can be found which effectively corresponds to the original system:

```
# model #25
visuOutGP(out10, selecmod = 25, prioMinMax = "model")
```

```
# Equations
visuEq(nVar = 3, dMax = 2, out10$models$model25, substit = 1, approx = 4)
```

```
## dx/dt = -1.00721 z  -0.98138 y
##
## dy/dt = 0.51883 y  + 0.99526 x
##
## dz/dt = 0.97635  -2.11514 z  + 0.59419 x z
```

The error on the model parameters is very heterogeneous and much higher than what was found in previous tests (from 0.2 % to 50 %). By experience, other parameters than the smoothing window may be considered to try to obtain better and/or simpler models. In particular, alternative windows of the time series may be a good way to avoid some local inconsistencies that may result from noise, resampling, or potentially from local nonstationary conditions.

## Sensitivity to measurement noise (after smoothing)

Measurement noise can be another source of perturbation. In the present section, we will investigate the possibility to retrieve the original Rössler model from the time series contaminated by measurement noise.

Data

Original time series (in gray) and noisy (in red) time series (5% of noise of the original signal variance)

```r
plot(outNumi1$reconstr[,1], outNumi1$reconstr[,2],
     type = 'l', col = 'gray', xlab = 't', ylab = 'x(t)', lwd = 3)
# variance of (x, y, z): (5.795879, 5.629243, 4.654374)
# add noise of 5% of the variance
An1 <- 0.05
t <- outNumi1$reconstr[,1]
datn1 <- outNumi1$reconstr[,2:4]
datn1[,1] <- datn1[,1] + rnorm(dim(datn1)[1], mean = 0, sd = sqrt(5.795879) * An1)
datn1[,2] <- datn1[,2] + rnorm(dim(datn1)[1], mean = 0, sd = sqrt(5.629243) * An1)
datn1[,3] <- datn1[,3] + rnorm(dim(datn1)[1], mean = 0, sd = sqrt(4.654374) * An1)
lines(t, datn1[,1],
      type = 'l', col = 'red', lwd = 1)

bf <- butter(2, 1/50, type="low")
b1x <- as.vector(filter(bf, datn1[,1]))
b1y <- as.vector(filter(bf, datn1[,2]))
b1z <- as.vector(filter(bf, datn1[,3]))
b1 <- cbind(b1x, b1y, b1z)
points(t[100:2000] - 0.5, b1[100:2000,1], col="black", pch=20)
legend(0, 5, c("original", "noisy (5%)"), col=c('gray', 'red'), lty=1, cex = 0.8)
```
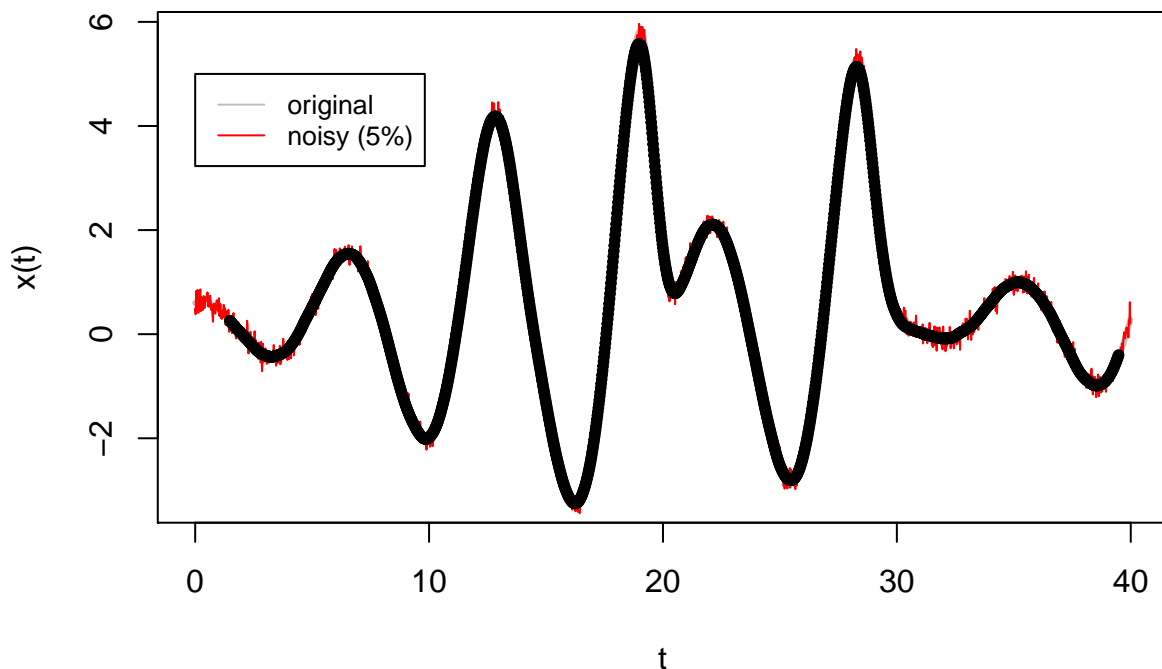

The time series (in black) was smoothed using a low pass filter (the butter function of the R package `signal` can be used for this purpose).

9

The same type of perturbation was applied with a higher level of noise (10%) added.

```
plot(outNumi1$reconstr[,1], outNumi1$reconstr[,2],
     type = 'l', col = 'gray', xlab = 't', ylab = 'x(t)', lwd = 3)
# add noise of 10% of the variance
An2 <- 0.2
t <- outNumi1$reconstr[,1]
datn2 <- outNumi1$reconstr[,2:4]
datn2[,1] <- datn2[,1] + rnorm(dim(datn2)[1], mean = 0, sd = sqrt(5.795879) * An2)
datn2[,2] <- datn2[,2] + rnorm(dim(datn2)[1], mean = 0, sd = sqrt(5.629243) * An2)
datn2[,3] <- datn2[,3] + rnorm(dim(datn2)[1], mean = 0, sd = sqrt(4.654374) * An2)
lines(t, datn2[,1],
      type = 'l', col = 'red', lwd = 1)
bf <- butter(2, 1/70, type="low")
b2x <- as.vector(filter(bf, datn2[,1]))
b2y <- as.vector(filter(bf, datn2[,2]))
b2z <- as.vector(filter(bf, datn2[,3]))
b2 <- cbind(b2x, b2y, b2z)
points(t[50:2000] - 0.5, b2[50:2000,1], col="black", pch=20)
legend(0, 5, c("original", "noisy (10%)"), col=c('gray', 'red'), lty=1, cex = 0.8)
```
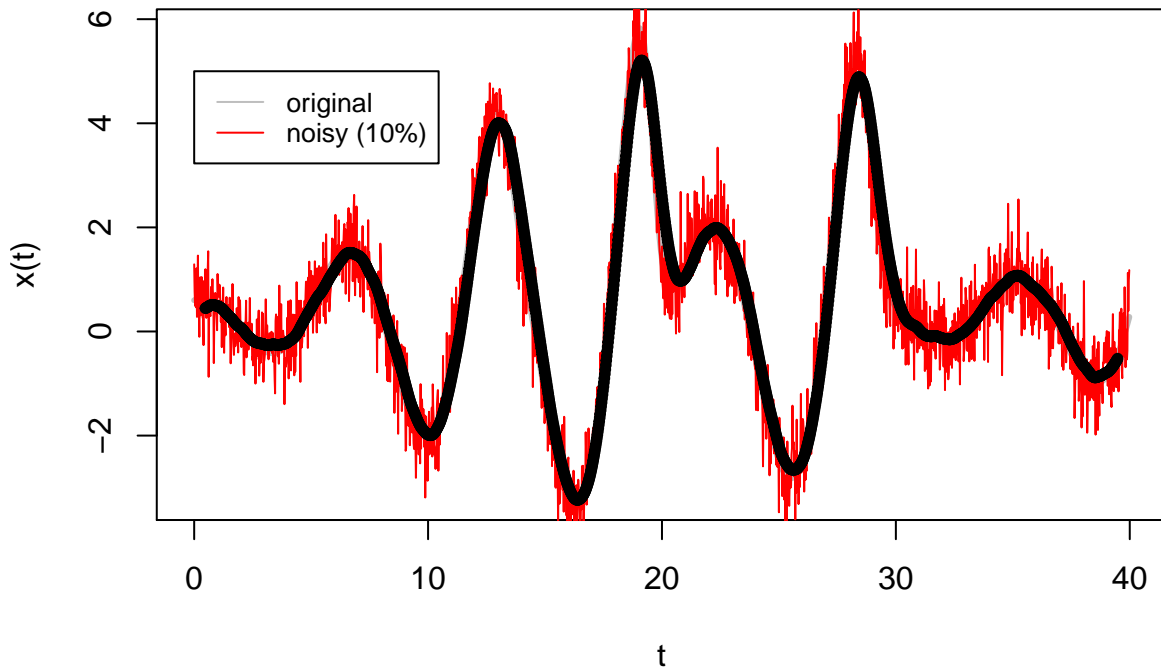

Global modelling

The global modelling technique is then applied to these noisy data sets

```
# model #25
visuOutGP(out11, selecmod = 25, prioMinMax = "model")
```

The simplest model presenting an appropriate phase portrait is the model #25 which has the same formulation as the original system:

```
# Equations
visuEq(nVar = 3, dMax = 2, out11$models$model25, substit = 1, approx = 4)
```

```
## dx/dt = -0.99604 z  -0.99883 y
##
## dy/dt = 0.52174 y  + 0.99986 x
##
## dz/dt = 1.66105  -3.32546 z  + 0.83747 x z
```

The error on the parameters can vary from 0.7 % to 17 %.

```
# min values
min(abs((out11$models$model25 - K) / K) * 100, na.rm = TRUE)
# max values
max(abs((out11$models$model25 - K) / K) * 100, na.rm = TRUE)
```

The same analysis could also be performed with a level of noise of 20% of variance compared to the original signal. In practice some variations can happen in the present experiment since each simulation will depend on the particular realization of noise generated by the random generator (the `rnorm` function was used here). The original equations could be retrieved in both cases with a maximum level of parameter error of around 24% (a smoother filter had to be applied for these cases).

Conclusions

In the present vignette, it was shown that, up to a certain extent, the approach can be considered as robust to (1) initial conditions changes, (2) measurement noise, (3) subsampling and (4) short length time series. When the conditions become more difficult, the precision of the model parameters will first decrease. Some spurious terms will start to appear in the model formulation under harder conditions. Further, no model at all will generally be obtained.

Because it is useful to try to separate the difficulties when attempting to understand a problem, these limitations were considered separately, one by one. Of course, more problems will be found when several of these difficulties will take place at the same time, which is generally the case when a data set from real world conditions is considered. In such cases, more investigations may be required (for choosing the period of analysis, the resampling and smoothing parameters, etc.), and intepretation should be considered with more cautions.

Many other limitations should be considered. The same chaotic system (the Rössler system) was systematically considered in order to keep the coherency between the various experiments. Based on the previous illustrations, what would be the difficulty when considering other dynamical systems remains unexplored. This question will start to be investigated in the next vignette `VII_retromodelling`.