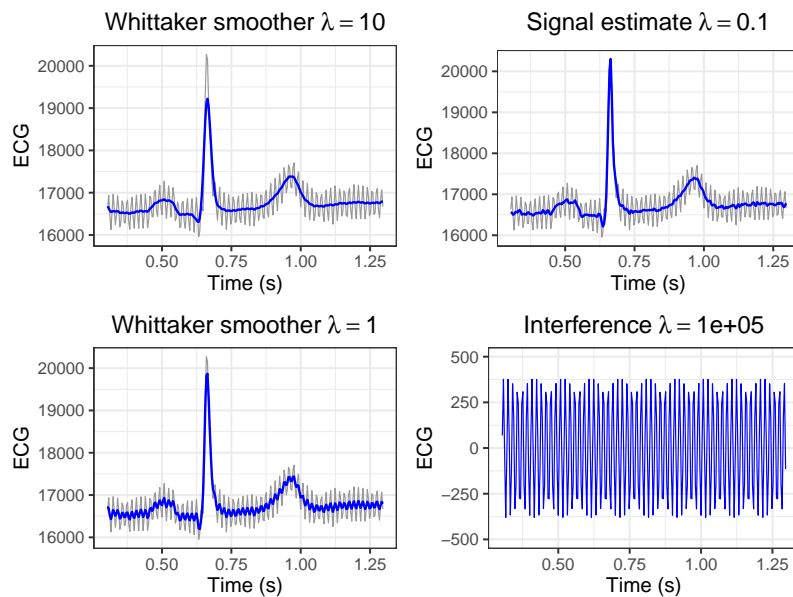


Whittaker trend and sinus, signal separation (ECG data)



Disturbance removal in a small section of an ECG by separating two signals using different penalties. Panels in left column: data (grey) and results of Whittaker smoothing (blue) with two different λ s. Top-right: data (grey) and extracted ECG (blue). Lower-right: extracted 60Hz disturbance. R code in `f-fit-hum.R`

```
# Whittaker trend and sinus, signal separation (ECG data)
# A graph in the book 'Practical Smoothing. The Joys of P-splines'
# Paul Eilers and Brian Marx, 2019

# Read Matlab data from Physiobank
library(spam)
library(ggplot2)
library(gridExtra)
library(JOPS)

# Get the data
data(ECG)
X = ECG
m0 = nrow(X)
time = (1:m0) * 0.004
sel = which(0.3 < time & time < 1.3)
y = X[sel, 3]
time = time[sel]
m = length(y)

# Basis
E = diag.spam(m)
B = cbind(E, E)

# Penalty 1
D = diff(E, diff = 2)
P1 = t(D) %*% D

# Penalty 2
p = (1/60) / 0.004
om = cos(2 * pi / p)
D2 = matrix(0, m - 2, m)
for (i in 1:(m - 2)) {
```

```

D2[i, i] = D2[i, i + 2] = 1
D2[i, i + 1] = -om * 2
}
D2 = as.spam(D2)
P2 = t(D2) %%% D2

# Make penalty matrix
P = diag.spam(2 * m)
r1 = 1:m
lambda1 = 0.1
P[r1, r1] = lambda1 * P1
r2 = r1 + m
lambda2 = 100000
P[r2, r2] = lambda2 * P2

# Solve the system
BtB = t(B) %%% B
z = solve(BtB + P, t(B) %%% y)
z1 = z[r1]
z2 = z[r2]
mu = B %%% z

# Diagnostics
K = solve(BtB + P, BtB)
dk1 = diag(K[r1, r1])
ed1 = sum(dk1)
dk2 = diag(K[r2, r2])
ed2 = sum(dk2)
cat('lambda1, lambda2, d1, ed2', lambda1, lambda2, ed1, ed2, '\n')

# Simple smooth
lambda3 = 1
z3 = solve(E + lambda3 * P1, y)
lambda4 = 10
z4 = solve(E + lambda4 * P1, y)

#Plot data and fits
lw = 0.3
lwd = 0.6
Data = data.frame(x = time, y = y)
Dfit = data.frame(x = time, y = z4)
titl = bquote("Whittaker smoother" ~ lambda== 10)
plt1 = ggplot(Data, aes(x = x, y = y)) +
  geom_line(aes(x = x, y = y), size = lw, color = grey(0.60)) +
  geom_line(data = Dfit, aes(x = x, y = y), size = lwd, color = 'blue') +
  xlab('Time (s)') + ylab('ECG') +
  ggtitle(titl) +
  JOPS_theme()

Data = data.frame(x = time, y = y)
Dfit = data.frame(x = time, y = z3)
titl = bquote("Whittaker smoother" ~ lambda== 1)
plt2 = ggplot(Data, aes(x = x, y = y)) +
  geom_line(aes(x = x, y = y), size = lw, color = grey(0.60)) +
  geom_line(data = Dfit, aes(x = x, y = y), size = lwd, color = 'blue') +
  xlab('Time (s)') + ylab('ECG') +
  ggtitle(titl) +
  JOPS_theme()

Data = data.frame(x = time, y = y)
Dfit = data.frame(x = time, y = z1)
titl = bquote("Signal estimate" ~ lambda== .(lambda1))
plt3 = ggplot(Data, aes(x = x, y = y)) +
  geom_line(aes(x = x, y = y), size = lw, color = grey(0.60)) +
  geom_line(data = Dfit, aes(x = x, y = y), size = lwd, color = 'blue') +
  xlab('Time (s)') + ylab('ECG') +
  ggtitle(titl) +

```

```
JOPS_theme()

Data = data.frame(x = time, y = z2)
Dfit = data.frame(x = time, y = z4)
titl = bquote("Interference" ~ lambda== .(lambda2))
plt4 = ggplot(Data, aes(x = x, y = y)) +
  geom_line(aes(x = x, y = y), size = lw, color = "blue") +
  xlab('Time (s)') + ylab('ECG') +
  ggtitle(titl) +
  ylim(500 * c(-1, 1)) +
  JOPS_theme()

# Create plots and pdf
grid.arrange(plt1, plt3, plt2, plt4, ncol = 2, nrow = 2)
```
