

SignalSmoothing.jl

Signal Smoothing and Bottom Detection in Sonar Data

This script loads sonar data, performs bottom detection, applies signal smoothing using two methods, and visualizes the results.

1. Signal Smoothing

Line 26

```
n = 1
```

- Chooses the index of the ping to process.

Line 27

```
data_selected = data[transect]
```

- Selects the sonar ping array from the chosen transect.

Line 30

```
smoothed_ping, window_size, snr_val = smoothSeqEcho(data_selected.P,  
n, 10, 1.0)
```

- Applies `smoothSeqEcho` across neighboring pings to reduce noise.

Lines 33–34

```
ping_signal = copy(data_selected.P[n, :])  
reliability, smoothed_range = smoothRange(ping_signal, 5, 2.0)
```

- Applies `smoothRange` to the 1D range profile of the selected ping.
-

2. Plotting

Lines 37–45

```
x = 1:size(data_selected.P, 2)  
y_original = data_selected.P[n, :]
```

```
fig = Figure(size = (800, 500))  
ax = Axis(fig[1, 1], xlabel = "Bin index", ylabel = "Signal  
amplitude", title = "Signal smoothing comparison")
```

```
lines!(ax, x, y_original, label = "Original Ping $n", linewidth = 2)  
lines!(ax, x, smoothed_range, label = "smoothRange (robust local  
fit)", linewidth = 2, linestyle = :dot)  
lines!(ax, x, smoothed_ping, label = "smoothSeqEcho (median over  
pings)", linewidth = 2, linestyle = :dash)
```

```
axislegend(ax, position = :lt)
```

```
display(fig)
```

- Creates a comparison plot of:
 - Original ping signal
 - Signal smoothed along the ping (`smoothRange`)

- Signal smoothed across pings (`smoothSeqEcho`)
- Uses `CairoMakie` for high-quality, interactive plotting.

