Input

Master IVI

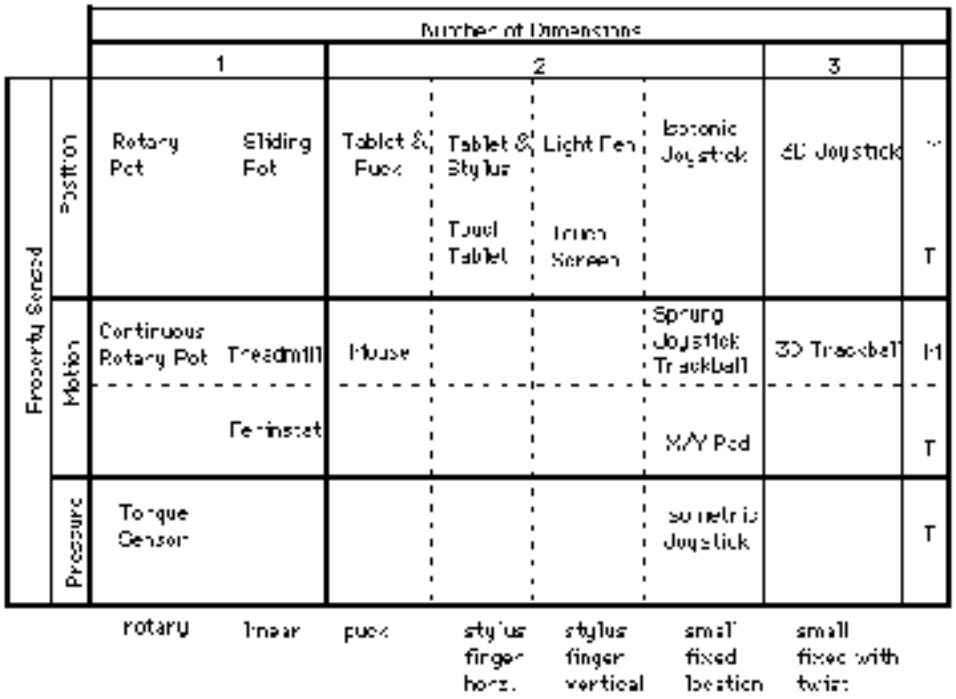
T. Pietrzak

Université de Lille 1 SCIENCES ET TECHNOLOGIES

http://research.microsoft.com/en-us/um/people/bibuxton/buxtoncollection/

http://www.billbuxton.com/inputManuscript.html

TAXONOMIES

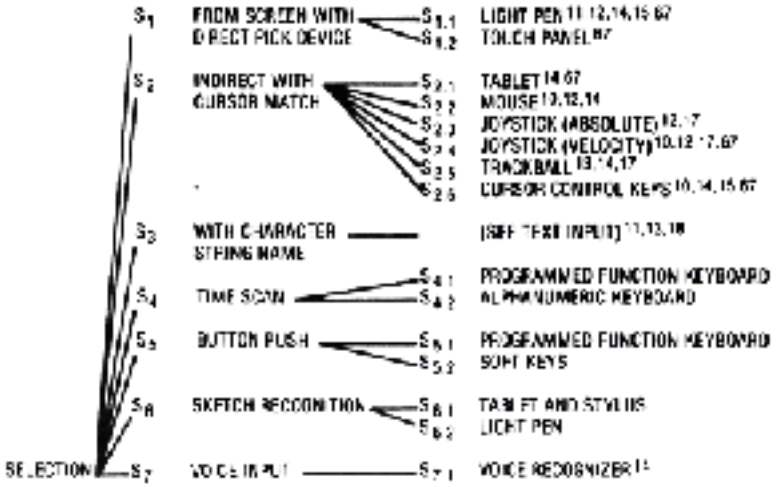


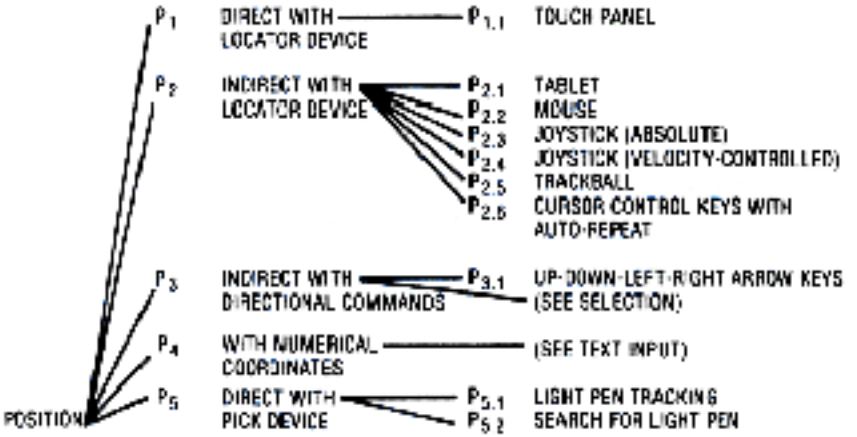
Bill Buxton, 1983
Lexical and Pragmatic Considerations of Input Structures
Computer Graphics, 17 (1), 31-37.

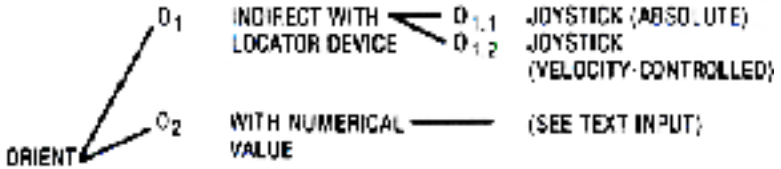
Tâches élémentaires :

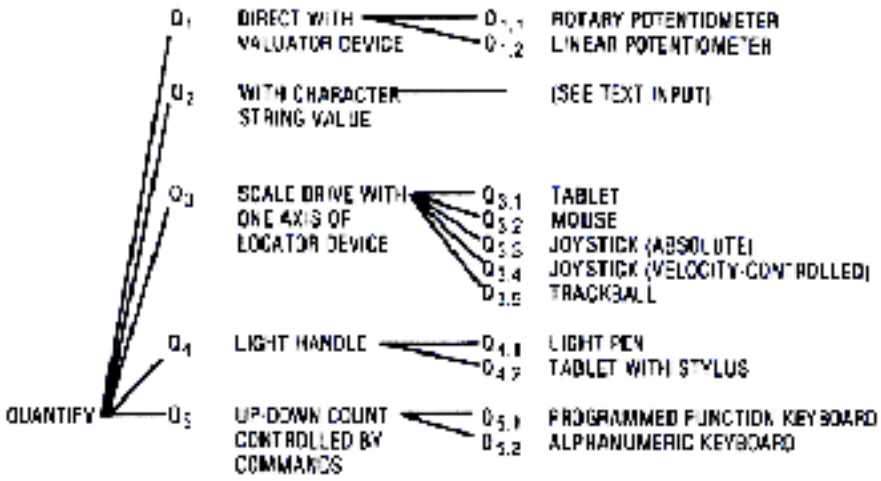
- **Select** : pointer un objet (menu, bouton, etc.)
- **Position** : placer un objet sur 1, 2, 3 ou plus de dimensions
- **Orient** : orienter un objet sur 1, 2, 3 ou plus de dimensions
- **Path** : dessiner une ligne, courbe, etc.
- **Quantify** : saisir une valeur scalaire
- **Text** : saisir du texte

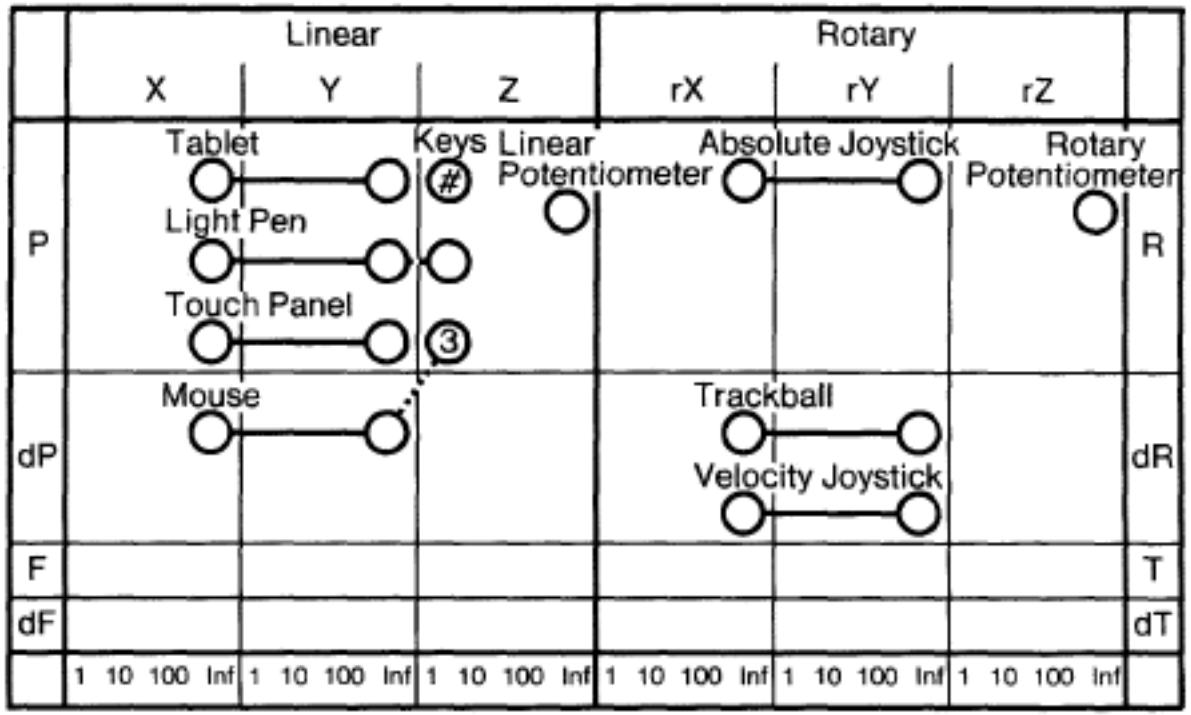
James D. Foley , Victor L. Wallace , Peggy Chan, 1984
The human factors of computer graphics interaction techniques
IEEE Computer Graphics and Applications, 4(11), 13-48


SELECTION
S1
FROM SCREEN WITH DIRECT PICK DEVICE
S1.1 LIGHT PEN
S1.2 TOUCH PANEL
S2
INDIRECT WITH CURSOR MATCH
S2.1 TABLET
S2.2 MOUSE
S2.3 JOYSTICK (ABSOLUTE)
S2.4 JOYSTICK (VELOCITY)
S2.5 TRACKBALL
S2.6 CURSOR CONTROL KEYS
S3
WITH CHARACTER STRING NAME
(SEE TEXT INPUT)
S4
TIME SCAN
S4.1 PROGRAMMED FUNCTION KEYBOARD
S4.2 ALPHANUMERIC KEYBOARD
S5
BUTTON PUSH
S5.1 PROGRAMMED FUNCTION KEYBOARD
S5.2 SOFT KEYS
S6
SKETCH RECOGNITION
S6.1 TABLET AND STYLUS
S6.2 LIGHT PEN
S7
VOICE INPUT
S7.1 VOICE RECOGNIZER

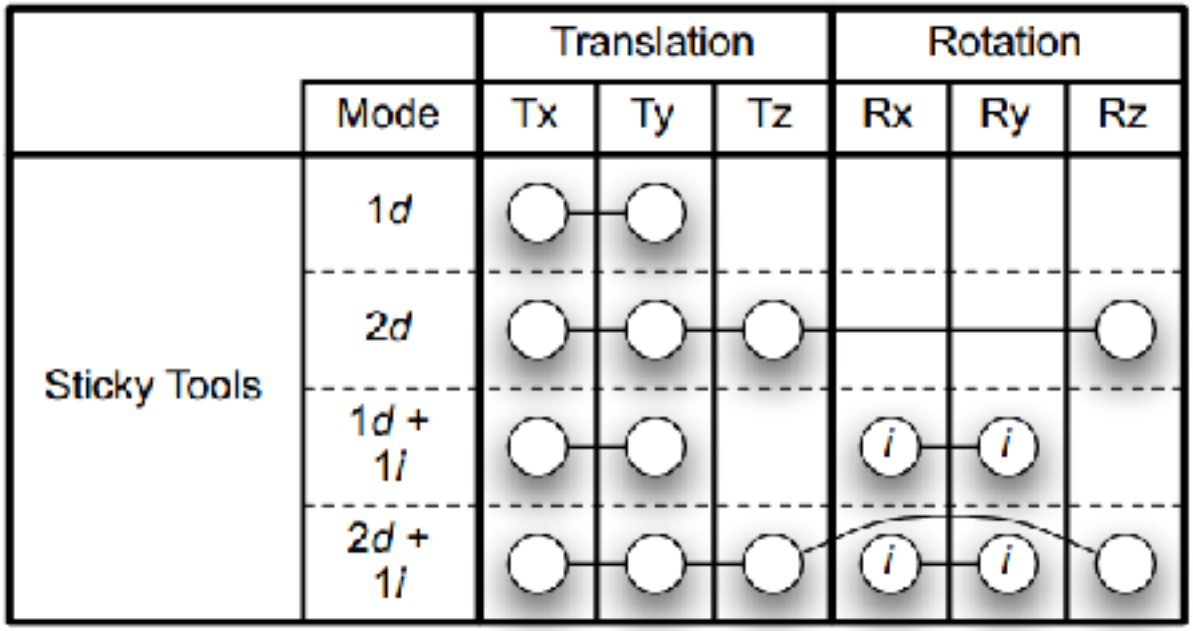

POSITION
P1
DIRECT WITH LOCATOR DEVICE
P1.1 TOUCH PANEL
P2
INDIRECT WITH LOCATOR DEVICE
P2.1 TABLET
P2.2 MOUSE
P2.3 JOYSTICK (ABSOLUTE)
P2.4 JOYSTICK (VELOCITY-CONTROLLED)
P2.5 TRACKBALL
P2.6 CURSOR CONTROL KEYS WITH AUTO-REPEAT
P3
INDIRECT WITH DIRECTIONAL COMMANDS
P3.1 UP-DOWN-LEFT-RIGHT ARROW KEYS
(SEE SELECTION)
P4
WITH NUMERICAL COORDINATES
(SEE TEXT INPUT)
P5
DIRECT WITH PICK DEVICE
P5.1 LIGHT PEN TRACKING
P5.2 SEARCH FOR LIGHT PEN


ORIENT
O1
INDIRECT WITH LOCATOR DEVICE
O1.1 JOYSTICK (ABSOLUTE)
O1.2 JOYSTICK (VELOCITY-CONTROLLED)
O2
WITH NUMERICAL VALUE
(SEE TEXT INPUT)

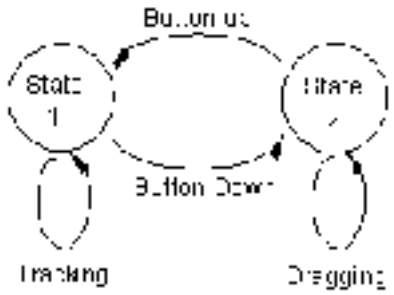

QUANTIFY
O1
DIRECT WITH VALUATOR DEVICE
O1.1 ROTARY POTENTIOMETER
O1.2 LINEAR POTENTIOMETER
O2
WITH CHARACTER STRING VALUE
(SEE TEXT INPUT)
O3
SCALE DRIVE WITH ONE AXIS OF LOCATOR DEVICE
O3.1 TABLET
O3.2 MOUSE
O3.3 JOYSTICK (ABSOLUTE)
O3.4 JOYSTICK (VELOCITY-CONTROLLED)
O3.5 TRACKBALL
O4
LIGHT HANDLE
O4.1 LIGHT PEN
O4.2 TABLET WITH STYLUS
O5
UP-DOWN COUNT CONTROLLED BY COMMANDS
O5.1 PROGRAMMED FUNCTION KEYBOARD
O5.2 ALPHANUMERIC KEYBOARD

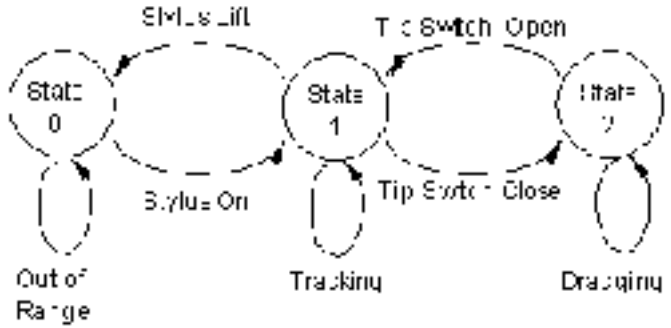


Jock D. Mackinley, Stuart K. Card, George G. Robertson , 1990
A Semantic Analysis of the Design Space of Input Devices
Journal of Human-Computer Interaction, 5 (2), 145-190.



Anthony Martinet, Géry Casiez, Laurent Grisoni, 2010
The Effect of DOF Separation in 3D Manipulation Tasks with Multi-touch Displays
VRST'10, 111-118





Bill Buxton, 1990
A Three-State Model of Graphical Input
INTERACT '90, 449-456.

SAISIE DE TEXTE

Chorded, 1968

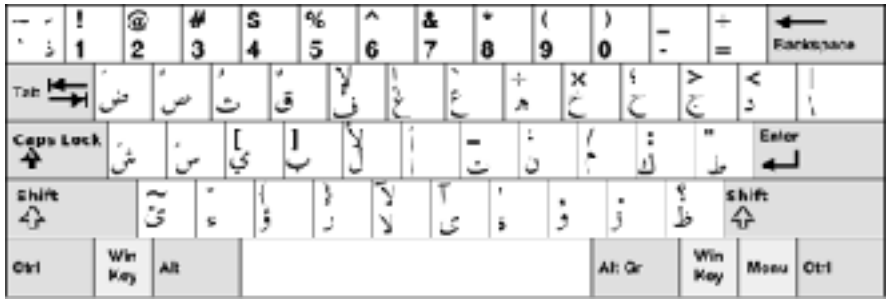
Datahand, 1990

Maltron, 1994

Orbitouch, 2002

Optimus, 2007


Technologie



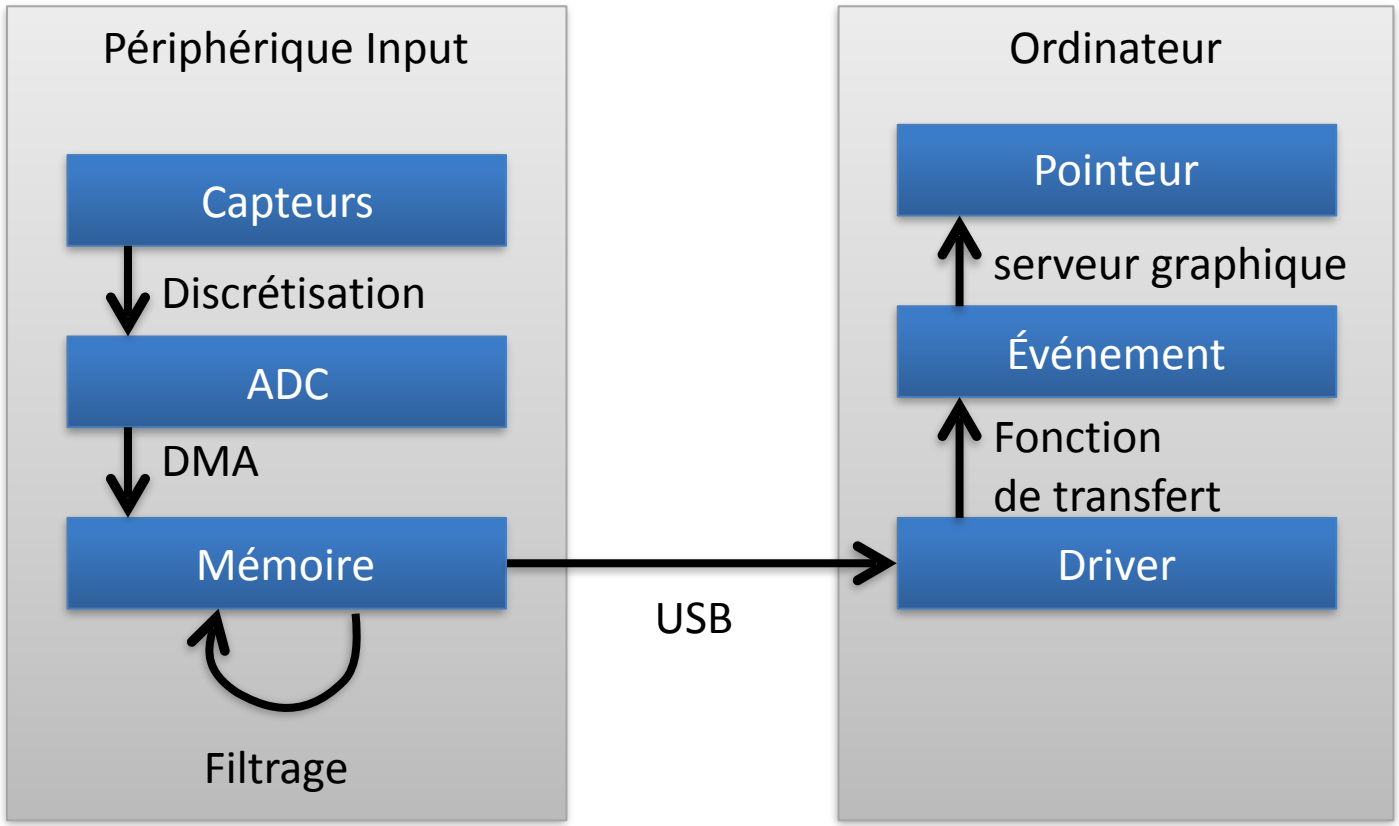
Saisie de phrases

- Fréquence des lettres
- Avec/sans correction
- Avec ou sans regard

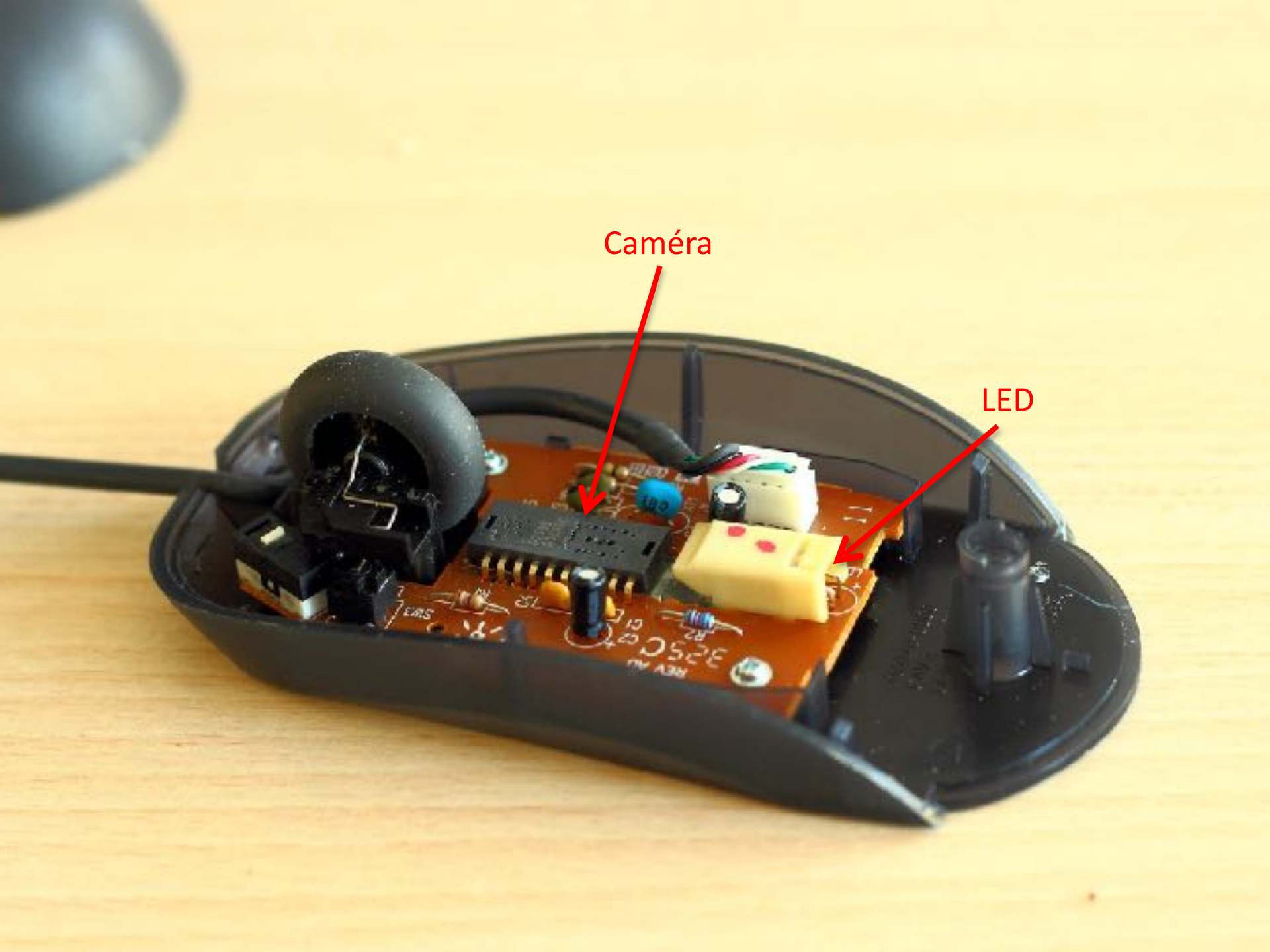
WPM : words per minute

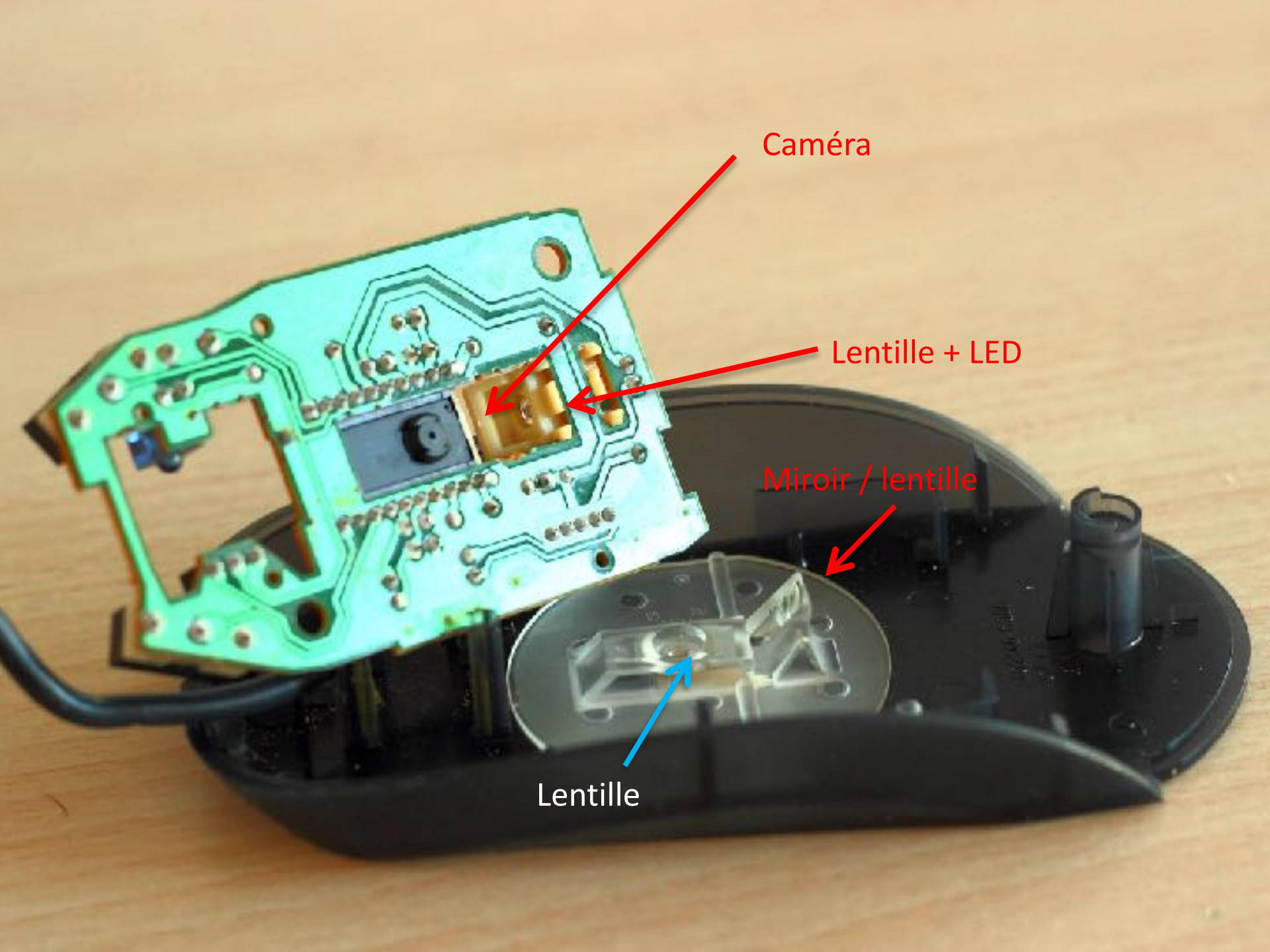
KSPC : key strokes per character

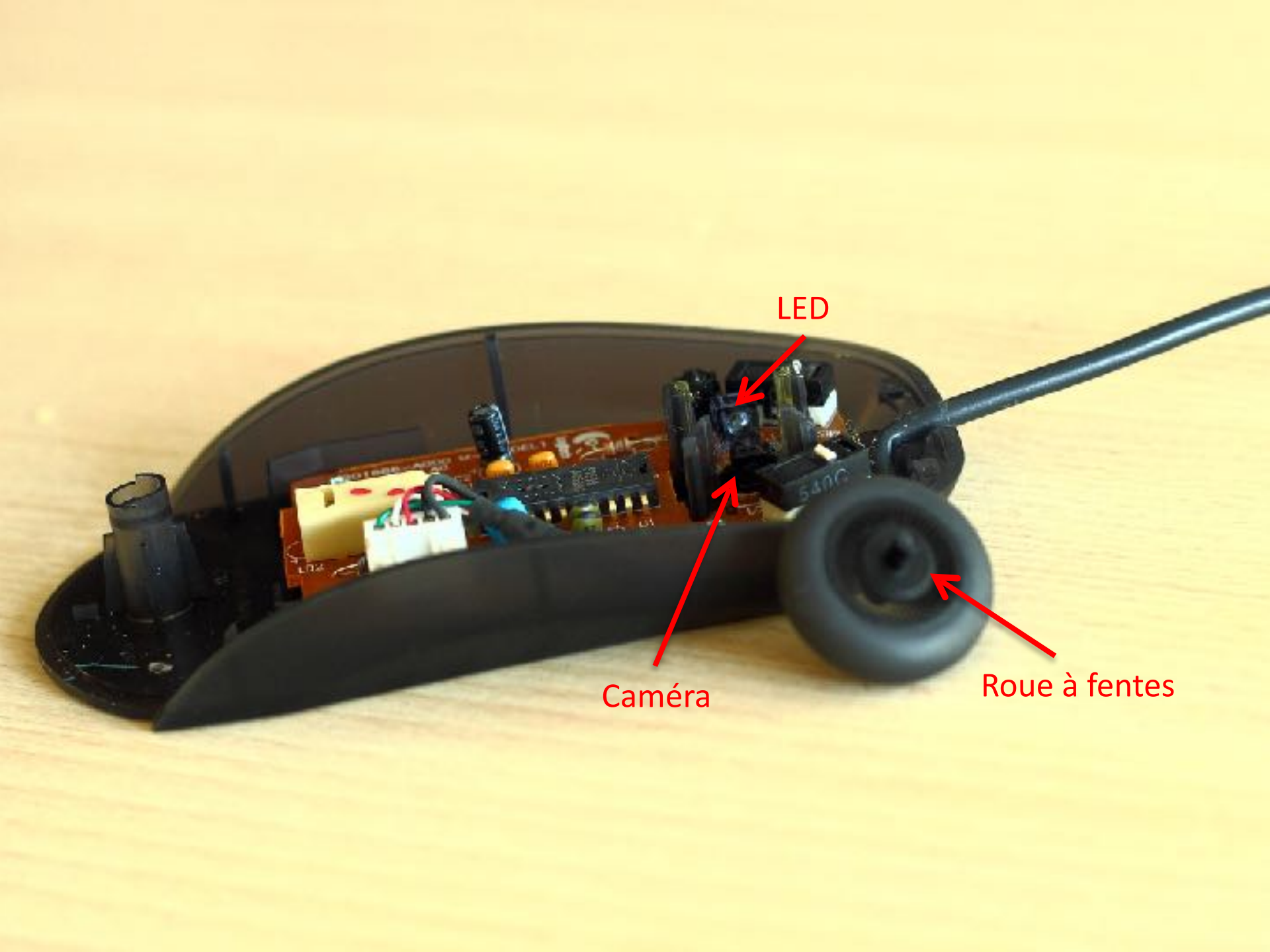
POINTAGE


Périphérique Input
Capteurs
Discrétisation
ADC
DMA
Mémoire
Filtrage
USB
Ordinateur
Pointeur
serveur graphique
Événement
Fonction
de transfert
Driver


Logitech
This device complies with Canada ICES-003


Caméra
LED

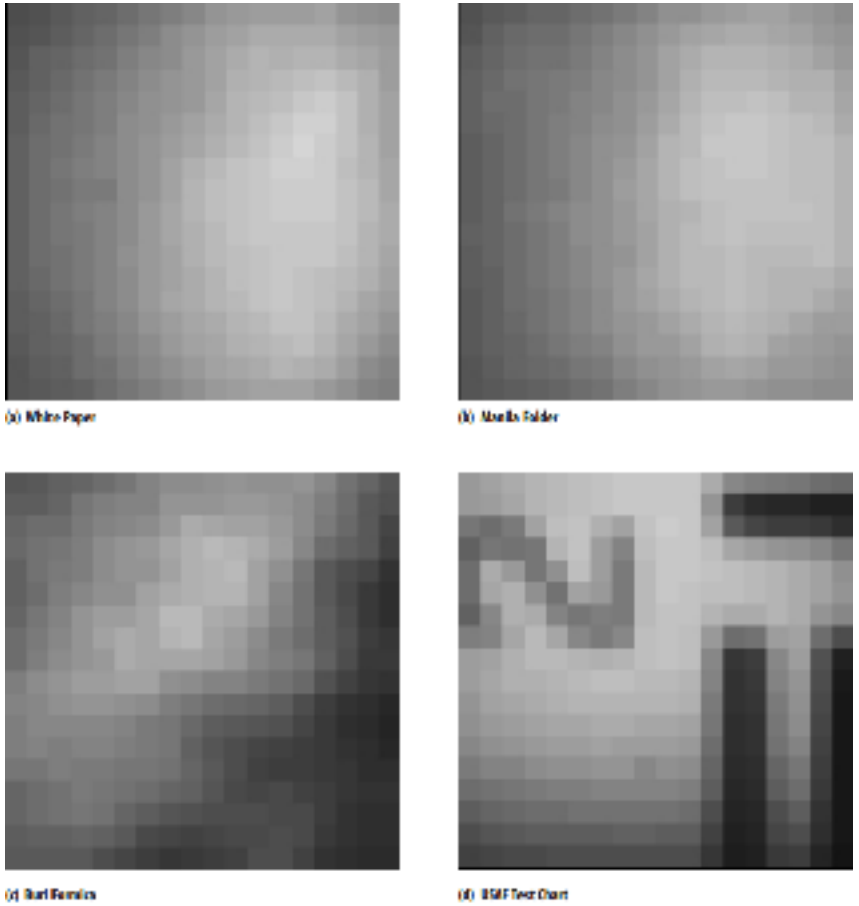

Caméra
Lentille + LED
Miroir / lentille
Lentille


LED
Caméra
Roue à fentes

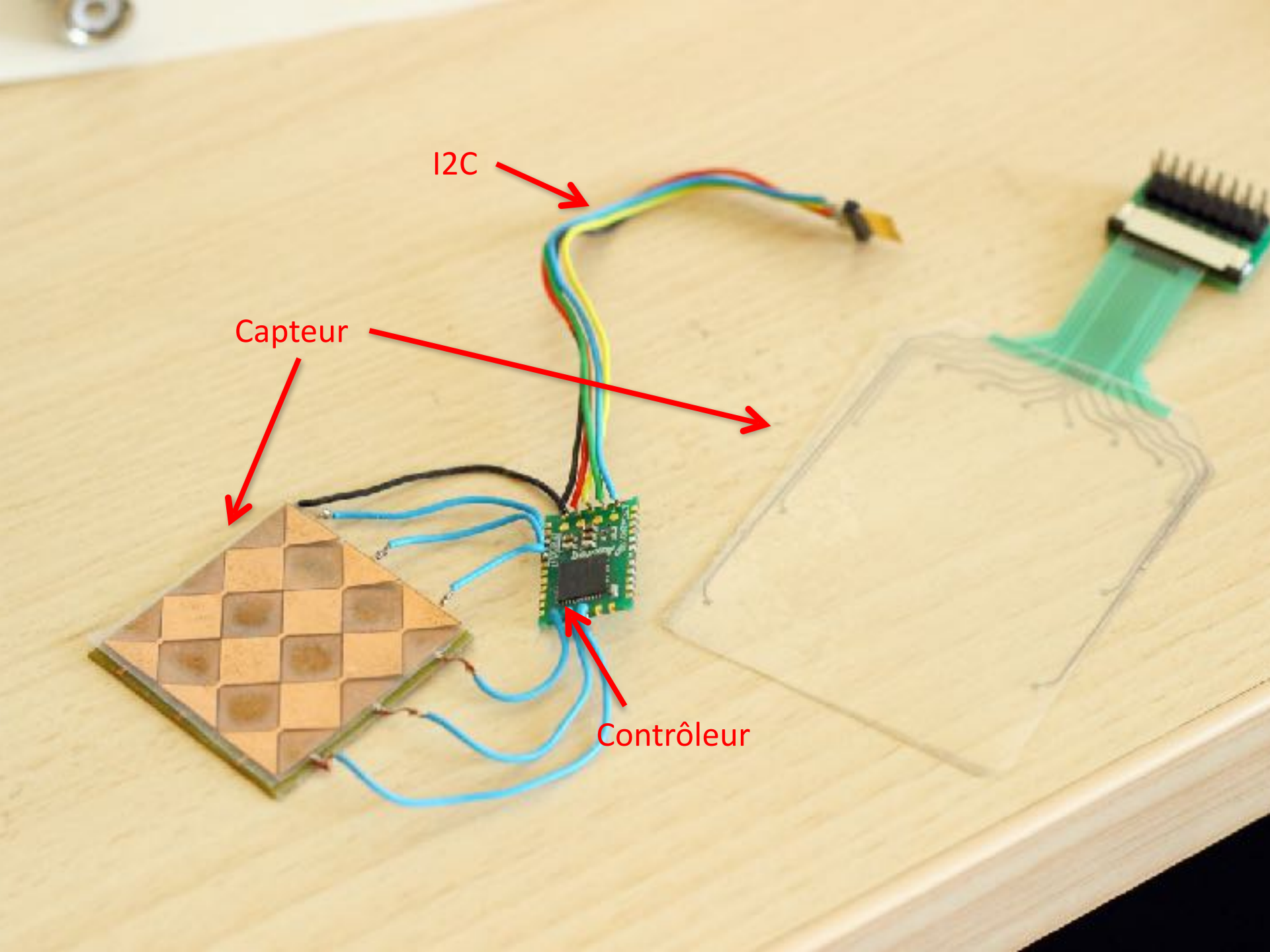
3000 images/s

≈ 20 x 20 pixels

Traitement d’images




Touchpad


I2C
Capteur
Contrôleur

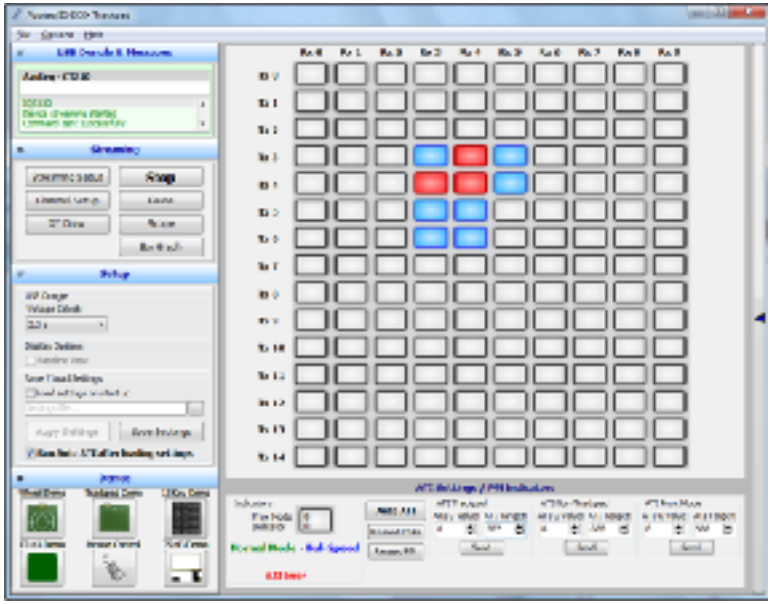
Mesure de capacité

Lignes × Colonnes

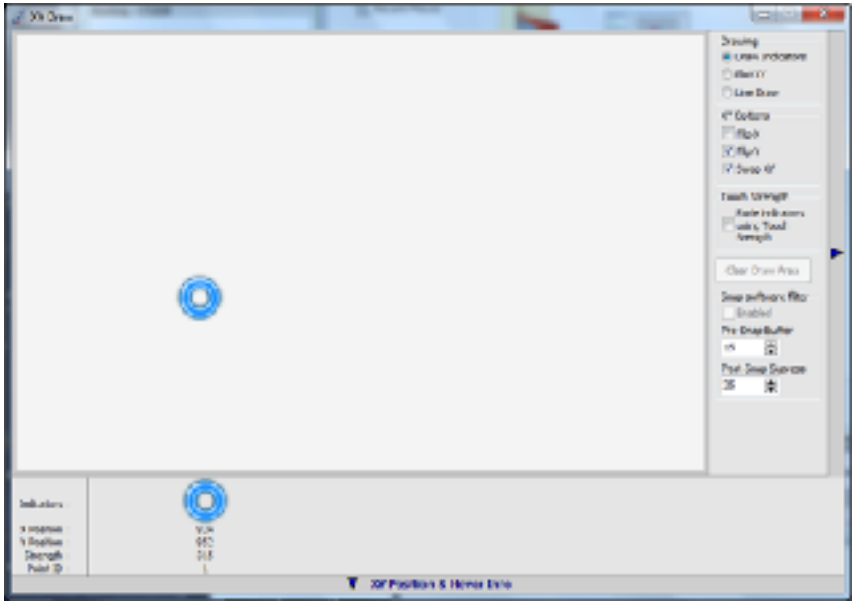
2 couches

Maximiser la surface

Mesure à chaque intersection



Calcul des centroïdes



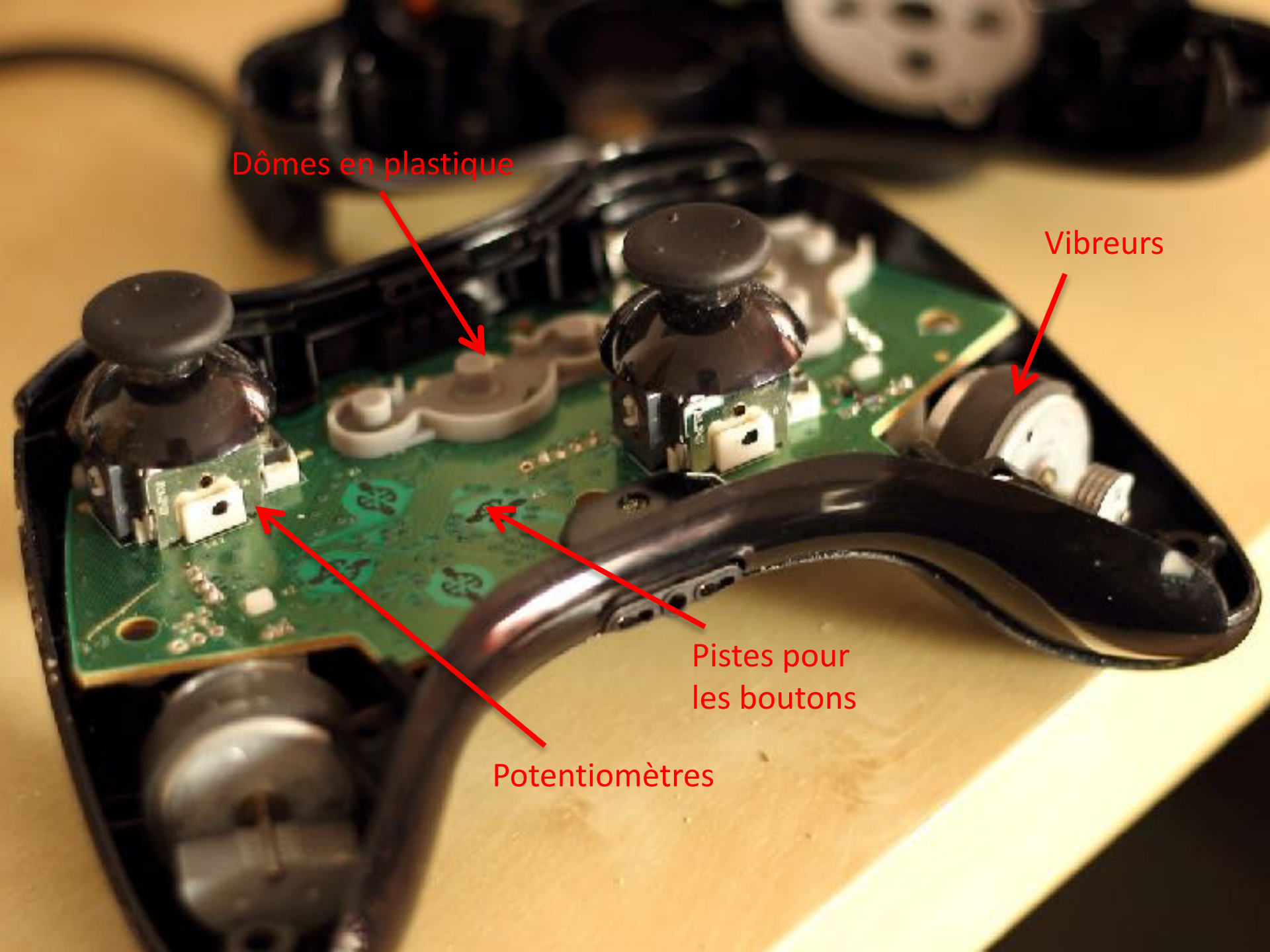
Modèle du temps de pointage

D : distance

W : taille de cible

$$MT = a + b \times log_2 \left(\frac{2D}{W} \right)$$


Joypad


Dômes en plastique
Vibreurs
Pistes pour
les boutons
Potentiomètres

FILTRAGE

- Problème : données bruitées
- Solution : filtre

- Compromis
  - Bruit : mouvement quand le périphérique est immobile
  - Latence : décalage entre le mouvement du périphérique et du curseur

- Problème : données bruitées
- Solution : filtre

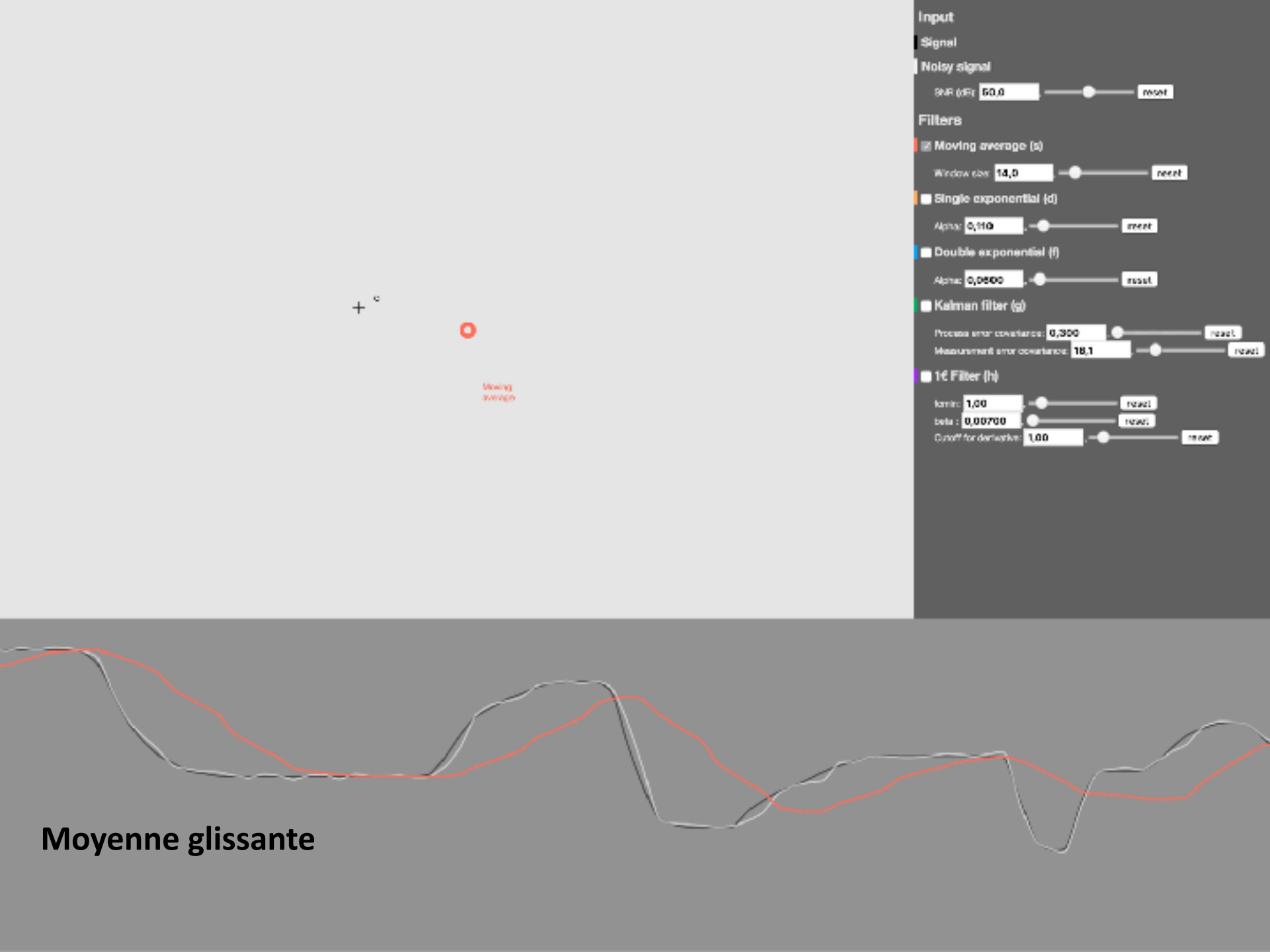
- Compromis
  - Bruit : mouvement quand le périphérique est immobile
  - Latence : décalage entre le mouvement du périphérique et du curseur

Moyenne sur les x dernières valeurs

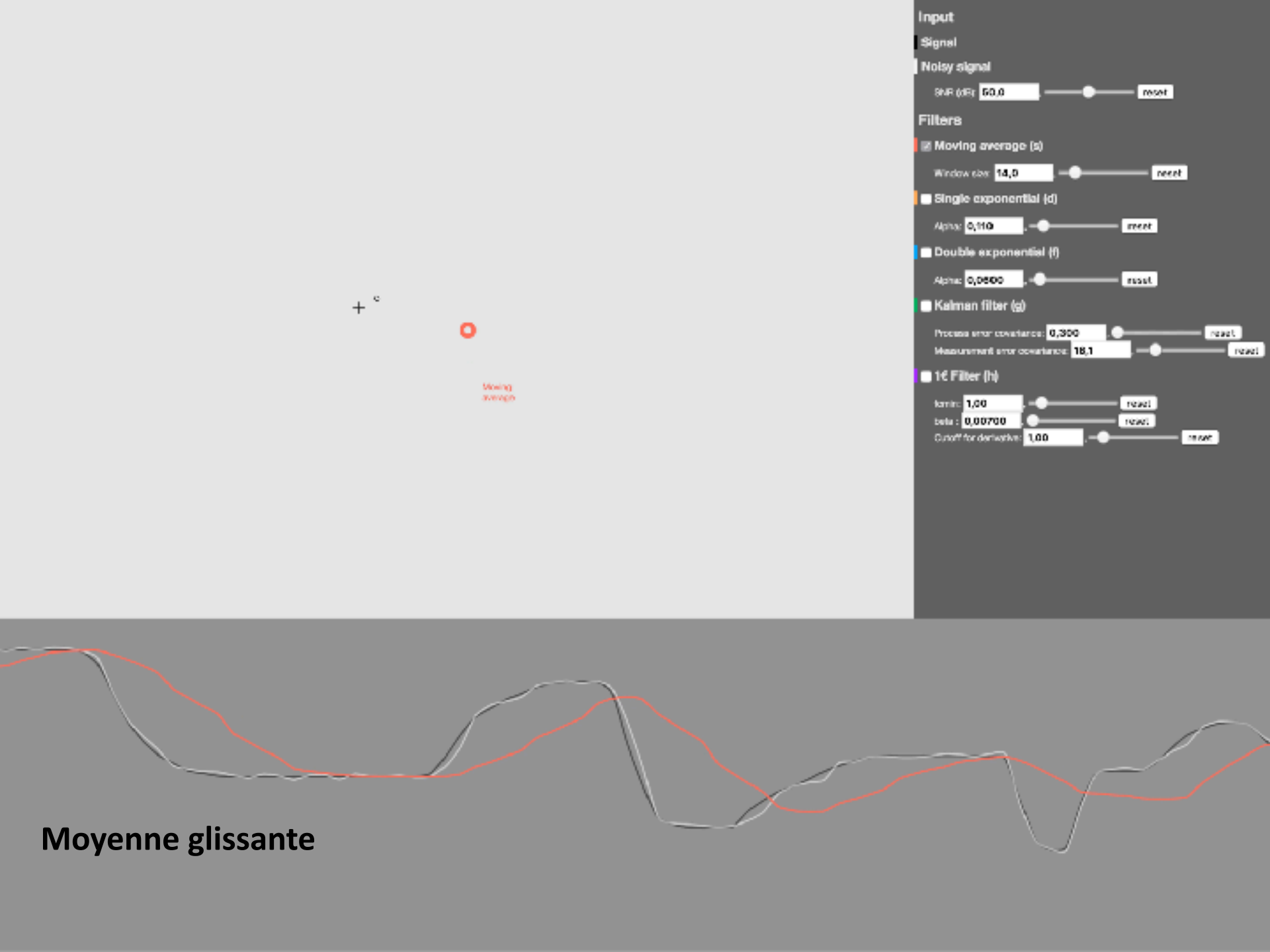
Lissage des points ⇒ gomme les mouvements brusques

\+ Facile à implémenter

\- Latence importante



Moyenne glissante



Moyenne glissante

Moyenne glissante dont le poids des valeurs décroit

Simple exponentiel

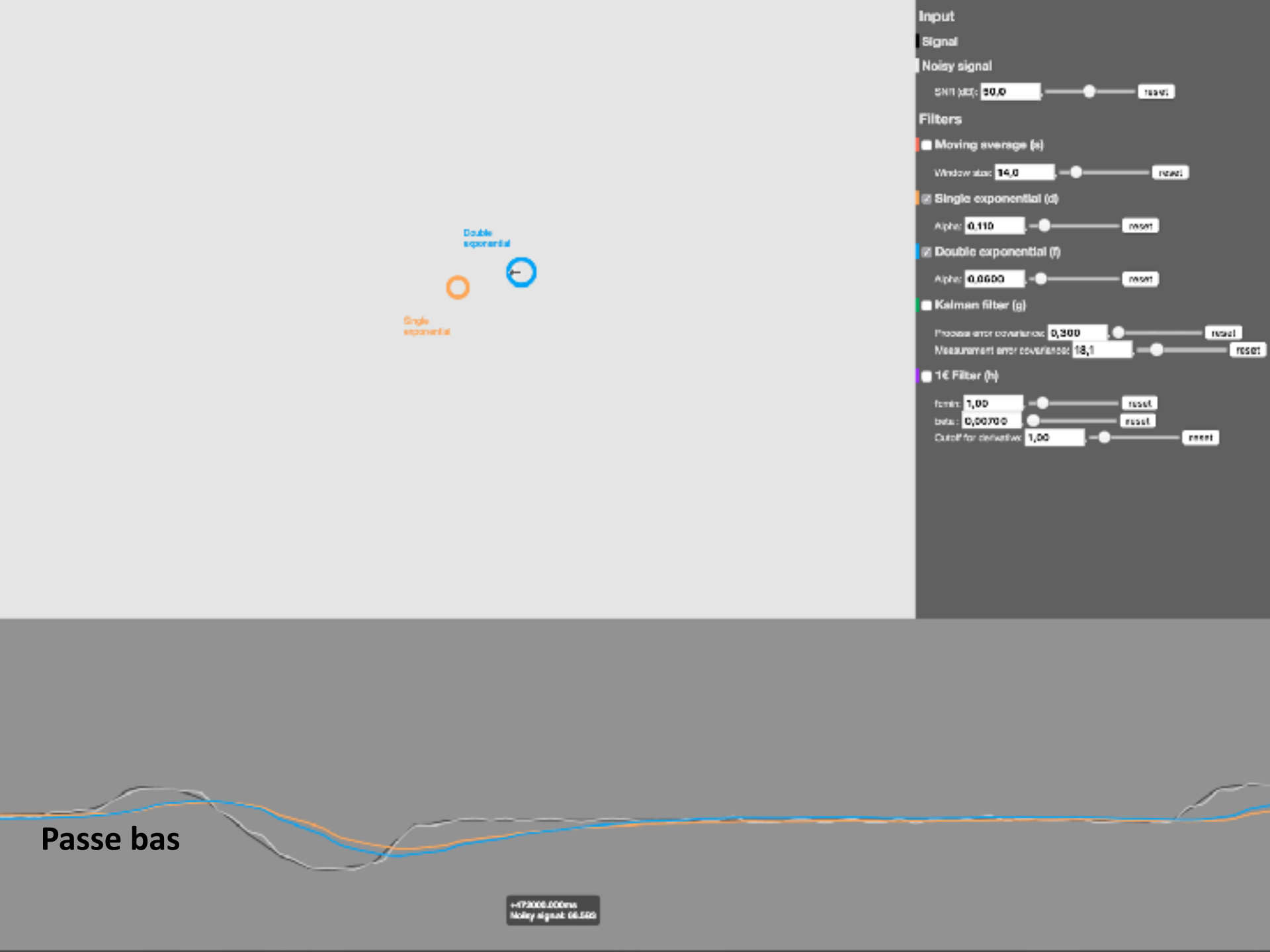
Double exponentiel

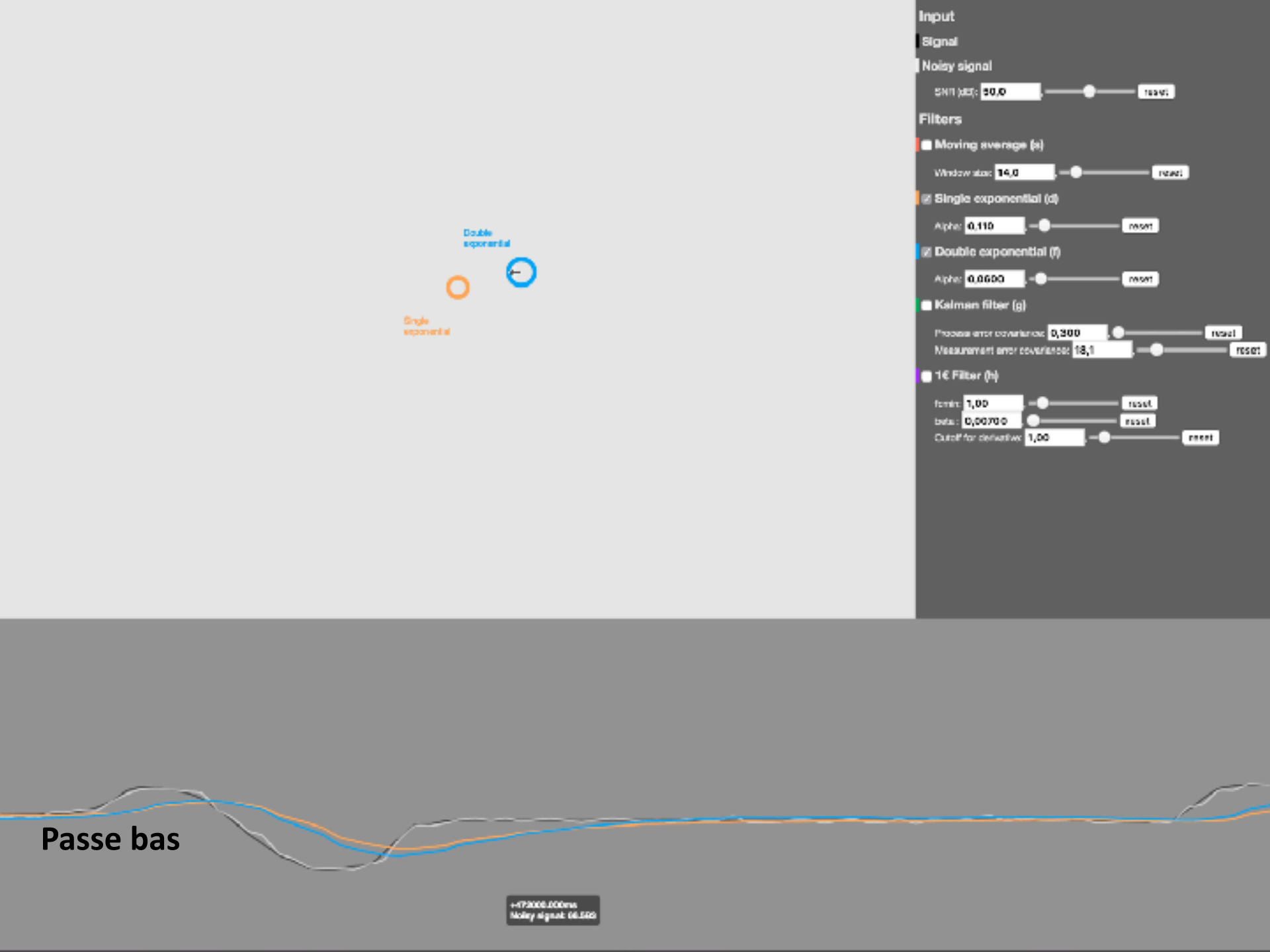
+ Facile à implémenter
- Latence importante

$$\hat{X}_i = \alpha X_i + (1-\alpha)\hat{X}_{i-1}$$

$$\hat{X}_i^{[2]} = \alpha \hat{X}_i + (1-\alpha)\hat{X}_{i-1}^{[2]}$$

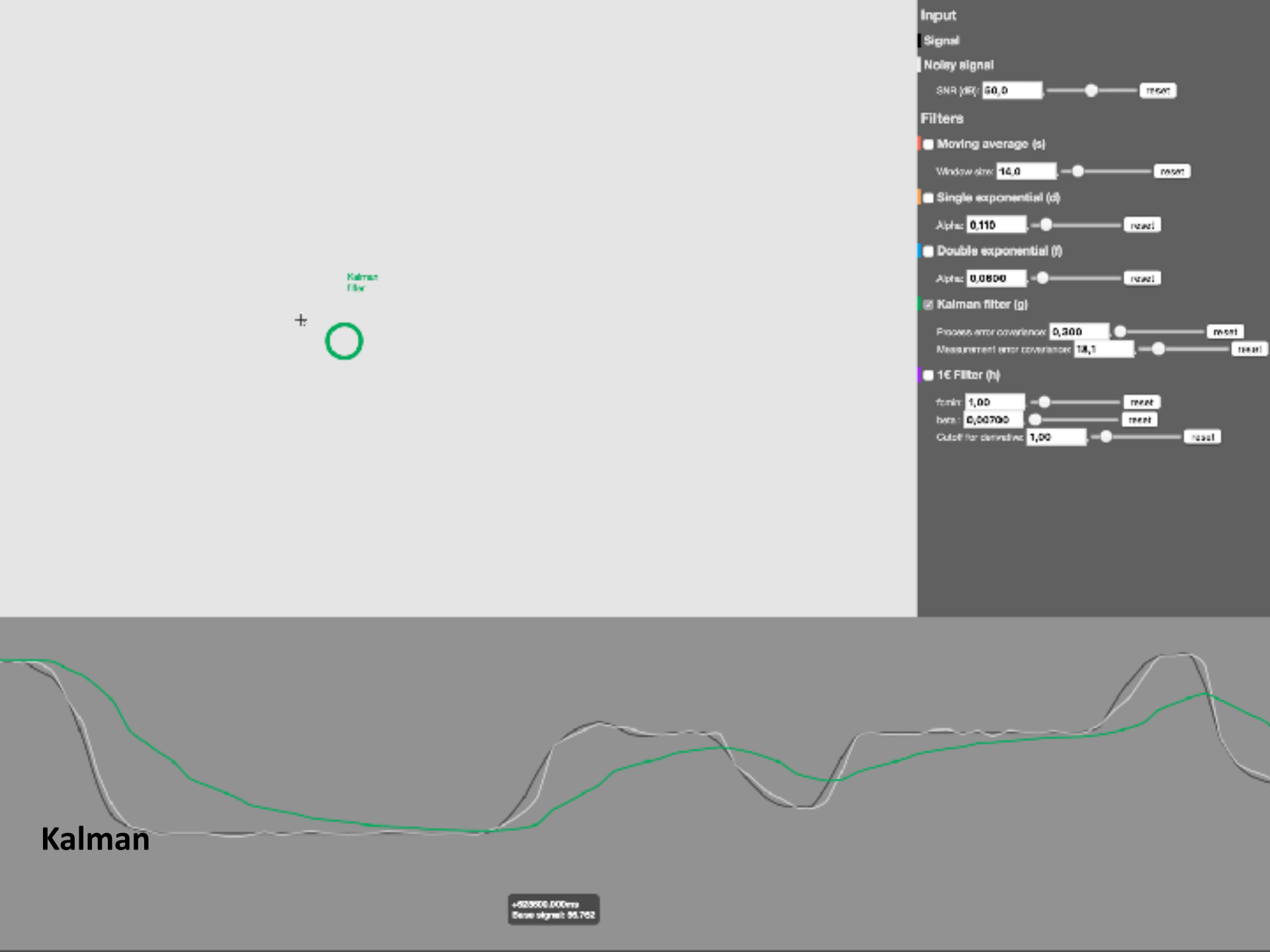
$$P_{t+\tau} = \left(2 + \frac{\alpha\tau}{1-\alpha}\right)\hat{X}_i - \left(1 + \frac{\alpha\tau}{1-\alpha}\right)\hat{X}_i^{[2]}$$


Input
Signal
Noisy signal
SNR (dB): 30,0
reset
Filters
Moving average (a)
Window size: 14,0
reset
Single exponential (d)
Alpha: 0.110
reset
Double exponential (f)
Alpha: 0.0600
reset
Kalman filter (g)
Process error covariance: 0,300
reset
Measurement error covariance: 18,1
reset
1€ Filter (h)
fcmin: 1,00
reset
beta: 0,00700
reset
Cutoff for derivative: 1,00
reset
Double exponential
Single exponential
Passe bas

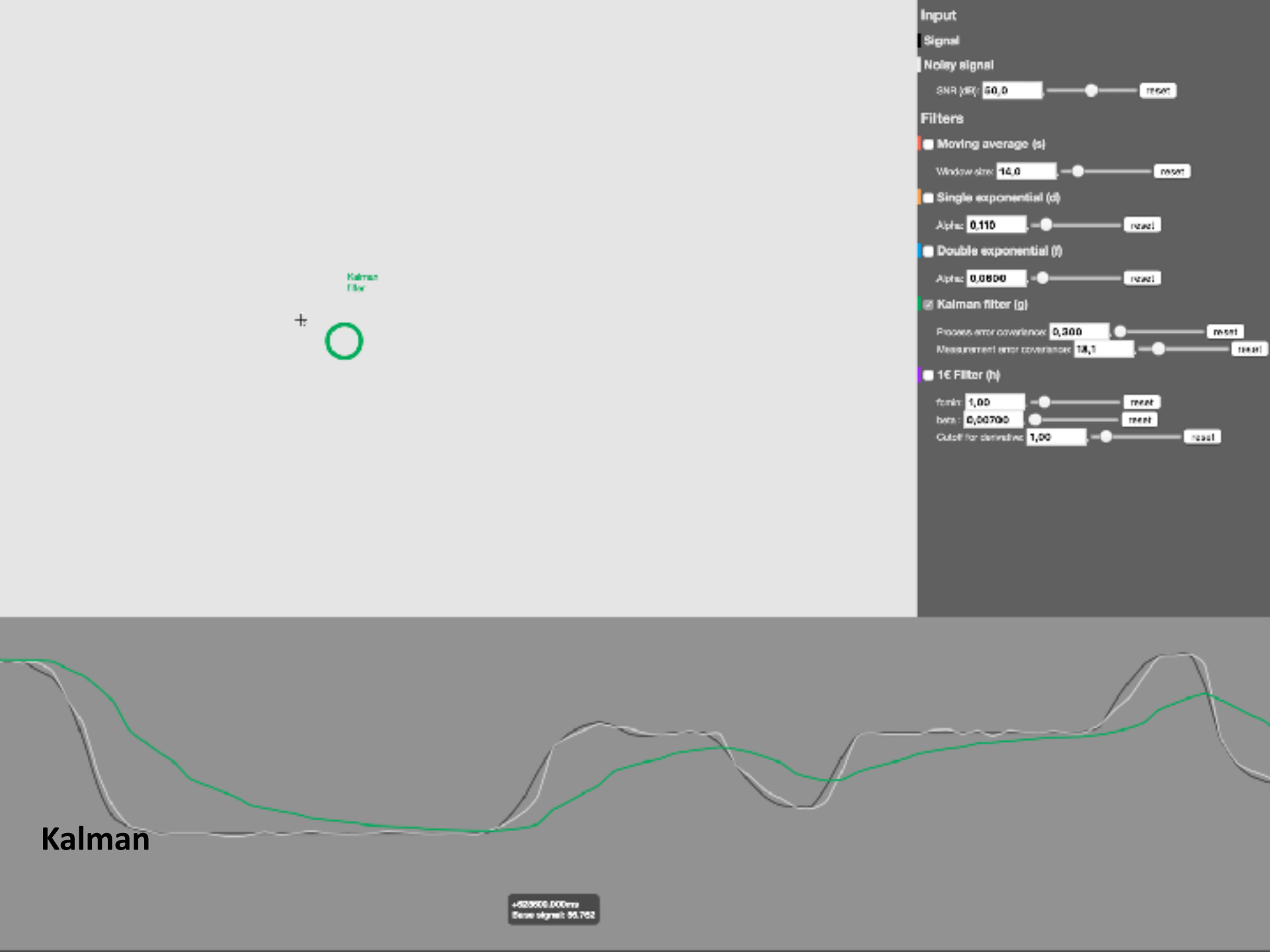

Input
Signal
Noisy signal
SNR (dB): 30,0
reset
Filters
Moving average (a)
Window size: 14,0
reset
Single exponential (d)
Alpha: 0.110
reset
Double exponential (f)
Alpha: 0.0600
reset
Kalman filter (g)
Process error covariance: 0,300
reset
Measurement error covariance: 18,1
reset
1€ Filter (h)
fcmin: 1,00
reset
beta: 0,00700
reset
Cutoff for derivative: 1,00
reset
Double exponential
Single exponential
Passe bas

Prédit les nouvelles valeurs en fonction de leur évolution

+ Plus efficace que la moyenne glissante
- Compliqué à comprendre
- Compliqué à régler
- Compliqué à implémenter


Input
Signal
Noisy signal
SNR (dB):
reset
Filters
Moving average (s)
Window size:
reset
Single exponential (d)
Alpha:
reset
Double exponential (f)
Alpha:
reset
Kalman filter (g)
Process error covariance:
reset
Measurement error covariance:
reset
1€ Filter (h)
fcmin:
reset
beta:
reset
Cutoff for derivative:
reset
Kalman
Kalman
filter

Kalman


Input
Signal
Noisy signal
SNR (dB):
reset
Filters
Moving average (s)
Window size:
reset
Single exponential (d)
Alpha:
reset
Double exponential (f)
Alpha:
reset
Kalman filter (g)
Process error covariance:
reset
Measurement error covariance:
reset
1€ Filter (h)
fcmin:
reset
beta:
reset
Cutoff for derivative:
reset
Kalman
Kalman
filter

Kalman

Filtre passe bas à fréquence de coupure variable

Filtre peu lorsque les mouvements sont rapides ⇒ peu de latence

Filtre beaucoup quand les mouvements sont lents ⇒ peu de bruit

+ Efficace
+ Facile à régler
+ Facile à implémenter

http://www.lifl.fr/~casiez/1euro/

APPENDIX A - 1€ FILTER

Algorithm 1: 1€ filter

EXT: First time flag: *firstTime* set to *true*
Data update rate: *rate*
Minimum cutoff frequency: *mincutoff*
Cutoff slope: *beta*
Low-pass filter: *xfilt*
Cutoff frequency for derivate: *dcutoff*
Low-pass filter for derivate: *dxfilt*
IN : Noisy sample value: *x*
OUT: Filtered sample value

```
1 if firstTime then
2 |   firstTime ← false
3 |   dx ← 0
4 else
5 |   dx ← (x - xfilt.hatxprev()) * rate
6 end
7 edx ← dxfilt.filter(dx, alpha(rate, dcutoff))
8 cutoff ← mincutoff + beta * |edx|
9 return xfilt.filter(x, alpha(rate, cutoff))
```

Algorithm 2: *Filter* method of Low-pass filter

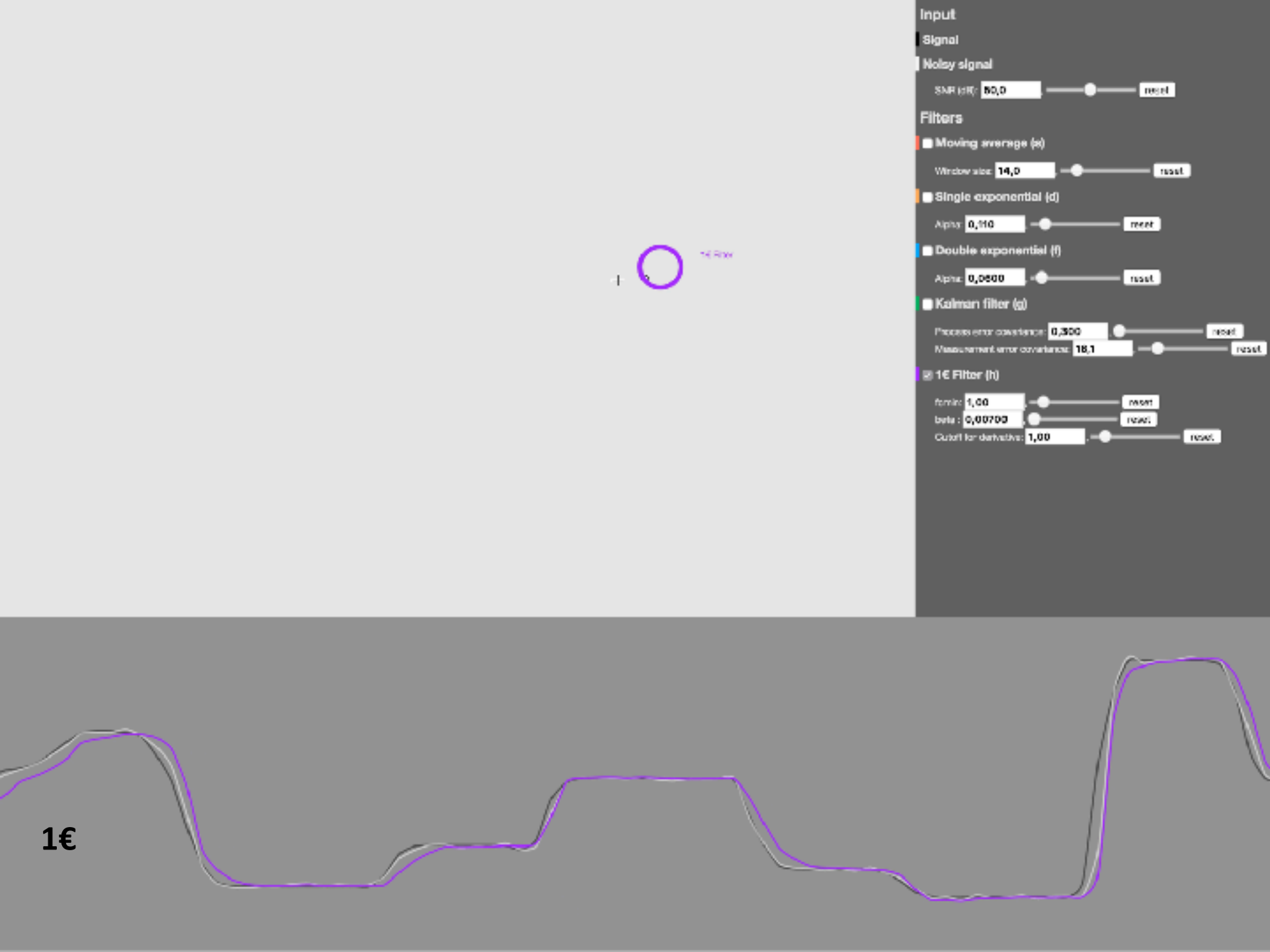
EXT: First time flag: *firstTime* set to *true*
IN : Noisy sample value : *x*
Alpha value : *alpha*
OUT: Filtered value

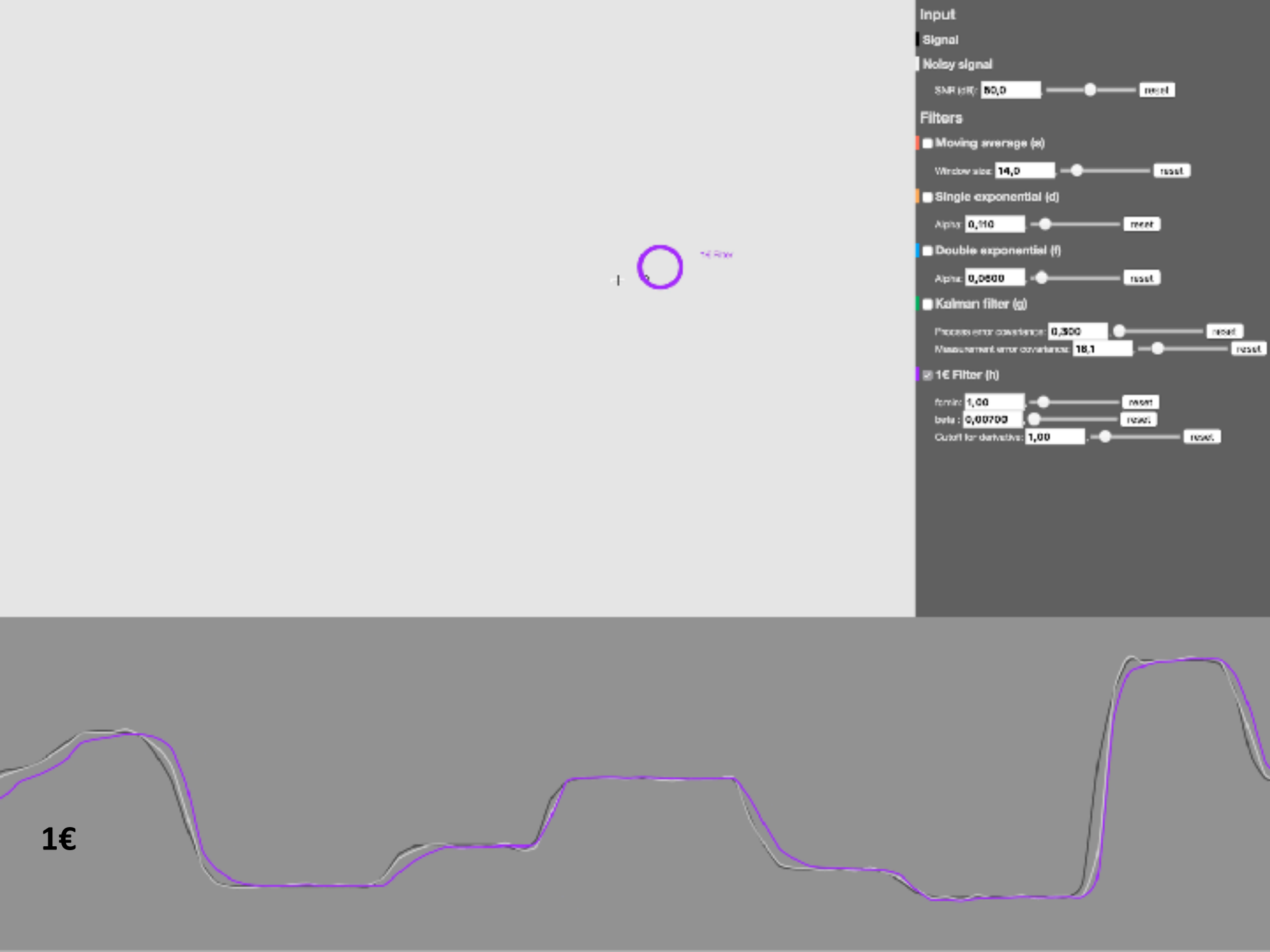
```
1 if firstTime then
2 |   firstTime ← false
3 |   hatxprev ← x
4 end
5 hatx ← alpha * x + (1 - alpha) * hatxprev
6 hatxprev ← hatx
7 return hatx
```

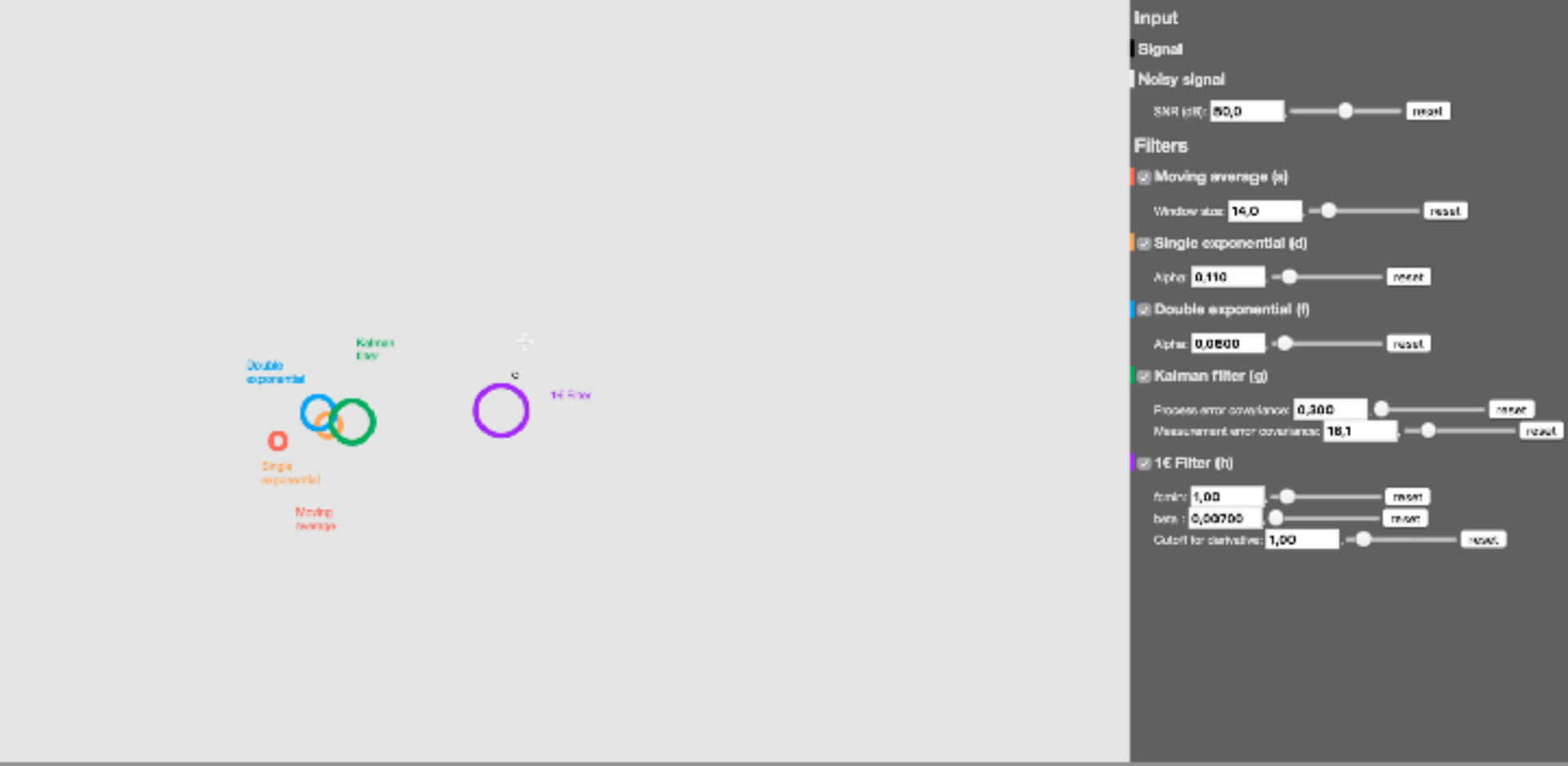
Algorithm 3: Alpha computation

IN : Data update rate in Hz: *rate*
Cutoff frequency in Hz: *cutoff*
OUT: Alpha value for low-pass filter

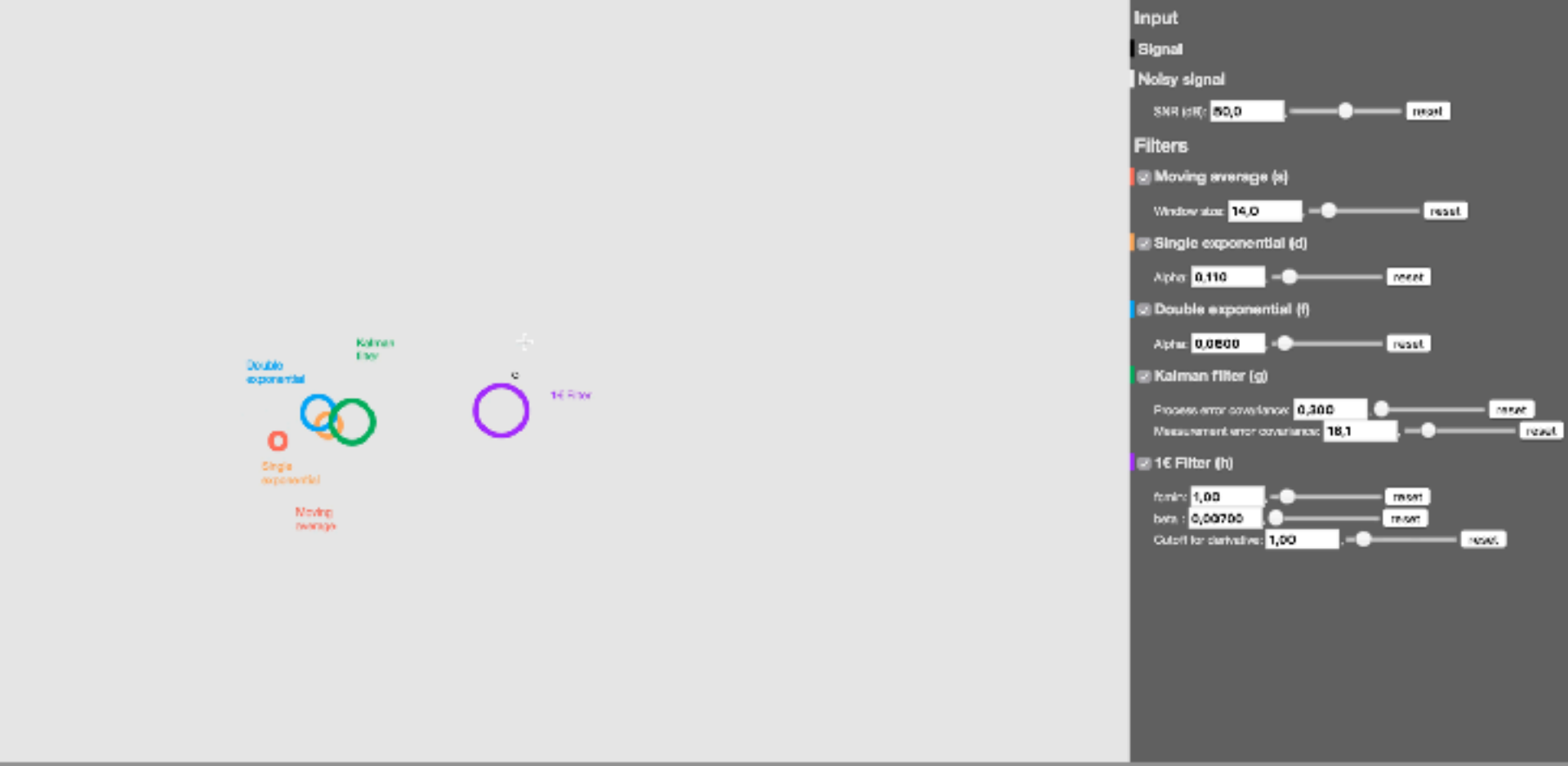
```
1 tau ← 1.0 / (2*π*cutoff)
2 te ← 1.0 / rate
3 return 1.0 / (1.0 + tau/te)
```


Input
Signal
Noisy signal
SNR (dB): 50,0
reset
Filters
Moving average (a)
Window size: 14,0
reset
Single exponential (d)
Alpha: 0,110
reset
Double exponential (f)
Alpha: 0,0600
reset
Kalman filter (g)
Process error covariance: 0,300
reset
Measurement error covariance: 18,1
reset
1€ Filter (h)
fcmin: 1,00
reset
beta : 0,00700
reset
Cutoff for derivative: 1,00
reset
1€


Input
Signal
Noisy signal
SNR (dB): 50,0
reset
Filters
Moving average (a)
Window size: 14,0
reset
Single exponential (d)
Alpha: 0,110
reset
Double exponential (f)
Alpha: 0,0600
reset
Kalman filter (g)
Process error covariance: 0,300
reset
Measurement error covariance: 18,1
reset
1€ Filter (h)
fcmin: 1,00
reset
beta : 0,00700
reset
Cutoff for derivative: 1,00
reset
1€

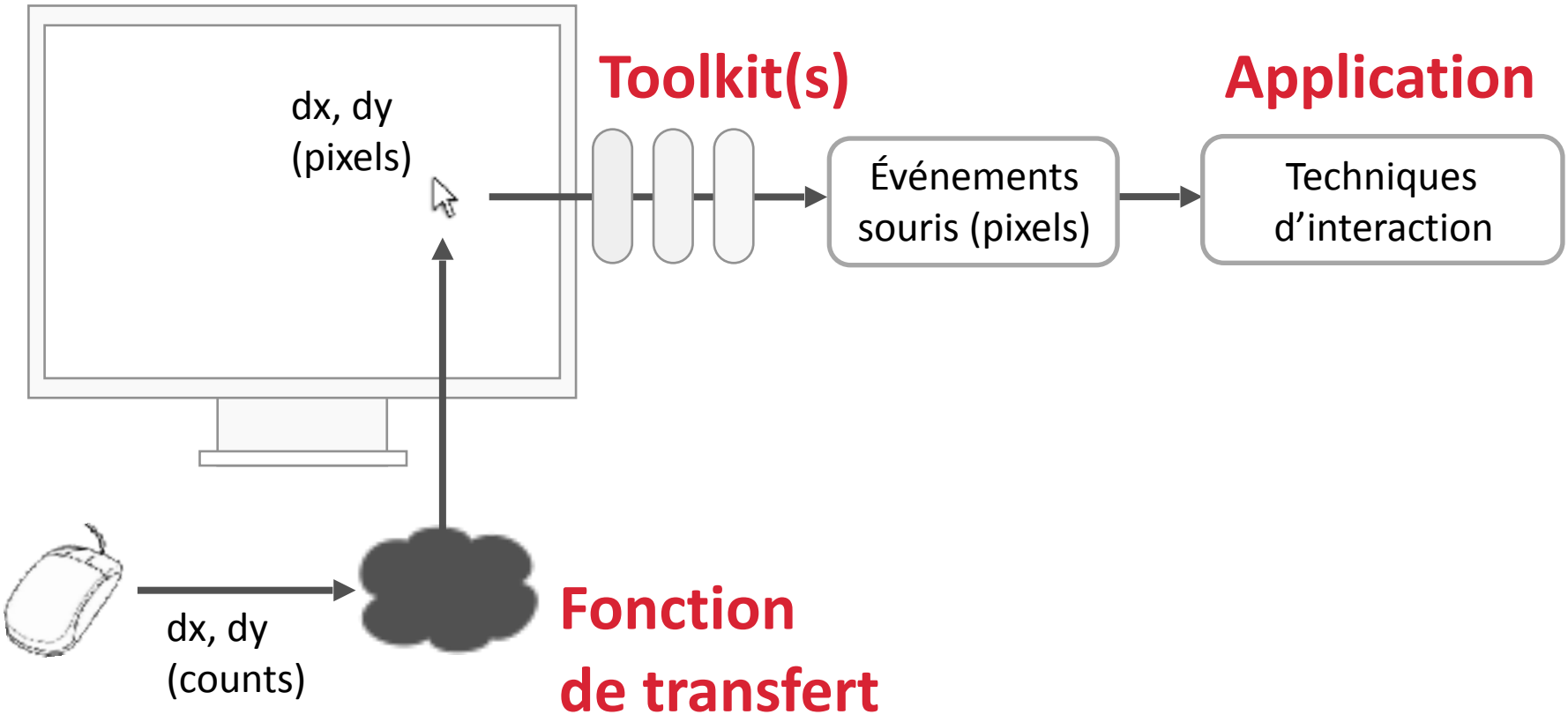

Input
Signal
Noisy signal
SNR (dB):
reset
Filters
Moving average (e)
Window size:
reset
Single exponential (d)
Alpha:
reset
Double exponential (f)
Alpha:
reset
Kalman filter (g)
Process error covariance:
reset
Measurement error covariance:
reset
1€ Filter (h)
fcmin:
reset
beta :
reset
Cutoff for derivative:
reset
Kalman filter
Double exponential
Single exponential
Moving average
1€ Filter

Tous


Input
Signal
Noisy signal
SNR (dB):
reset
Filters
Moving average (e)
Window size:
reset
Single exponential (d)
Alpha:
reset
Double exponential (f)
Alpha:
reset
Kalman filter (g)
Process error covariance:
reset
Measurement error covariance:
reset
1€ Filter (h)
fcmin:
reset
beta :
reset
Cutoff for derivative:
reset
Kalman filter
Double exponential
Single exponential
Moving average
1€ Filter

Tous

FONCTIONS DE TRANSFERT


dx, dy
(pixels)
Toolkit(s)
Application
Événements
souris (pixels)
Techniques
d'interaction
dx, dy
(counts)
Fonction
de transfert

De combien se déplace le curseur ?

Contrôle en position

ex : souris

Contrôle en vitesse

ex : joystick

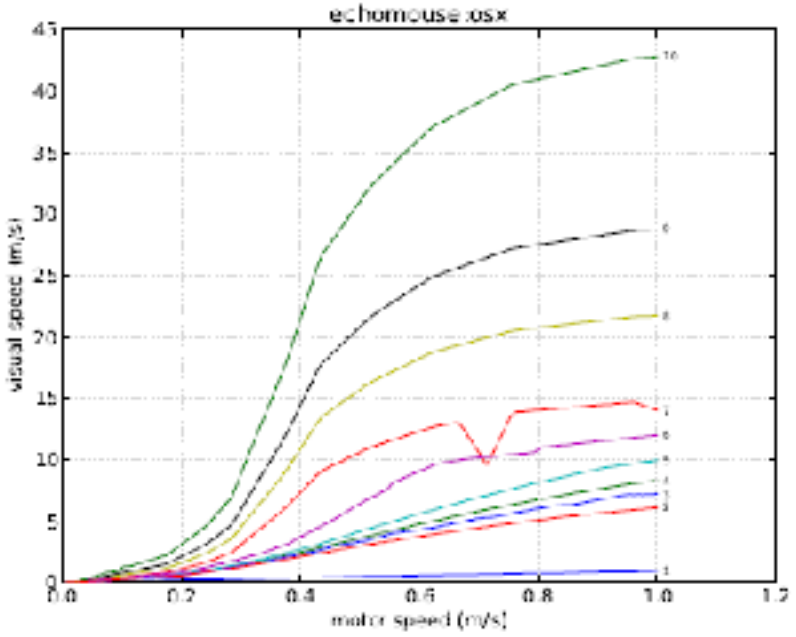
$$CD_{gain} = \frac{V_{pointeur}}{V_{périphérique}}$$

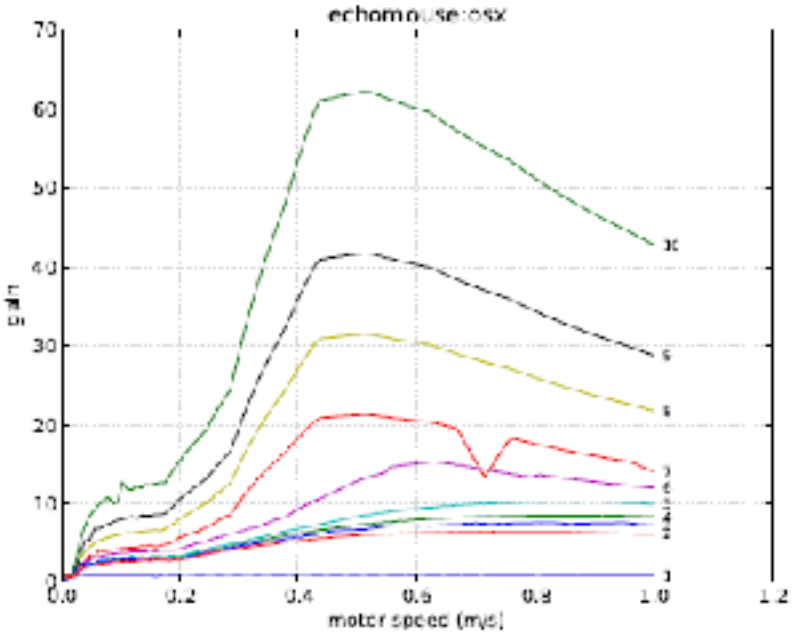
Fonction linéaire

Gain constant

Fonction non linéaire

Gain variable, par exemple en fonction de la vitesse


echomouse.osx
visual speed (m/s)
motor speed (m/s)


echomouse.osx
gain
motor speed (m/s)

Phase balistique

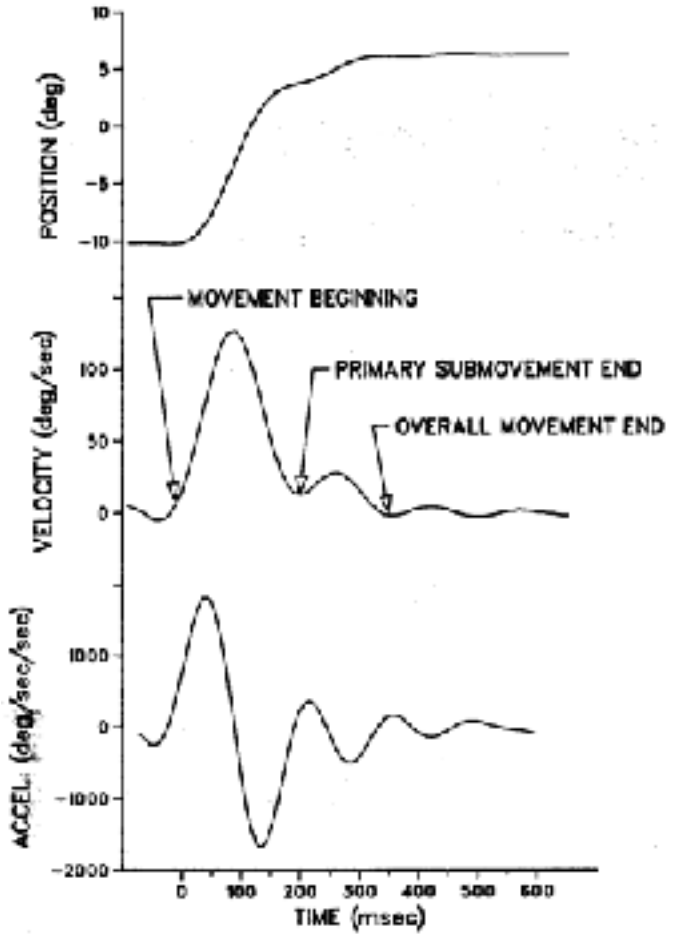
Rapide et imprécise

⇒ gain élevé

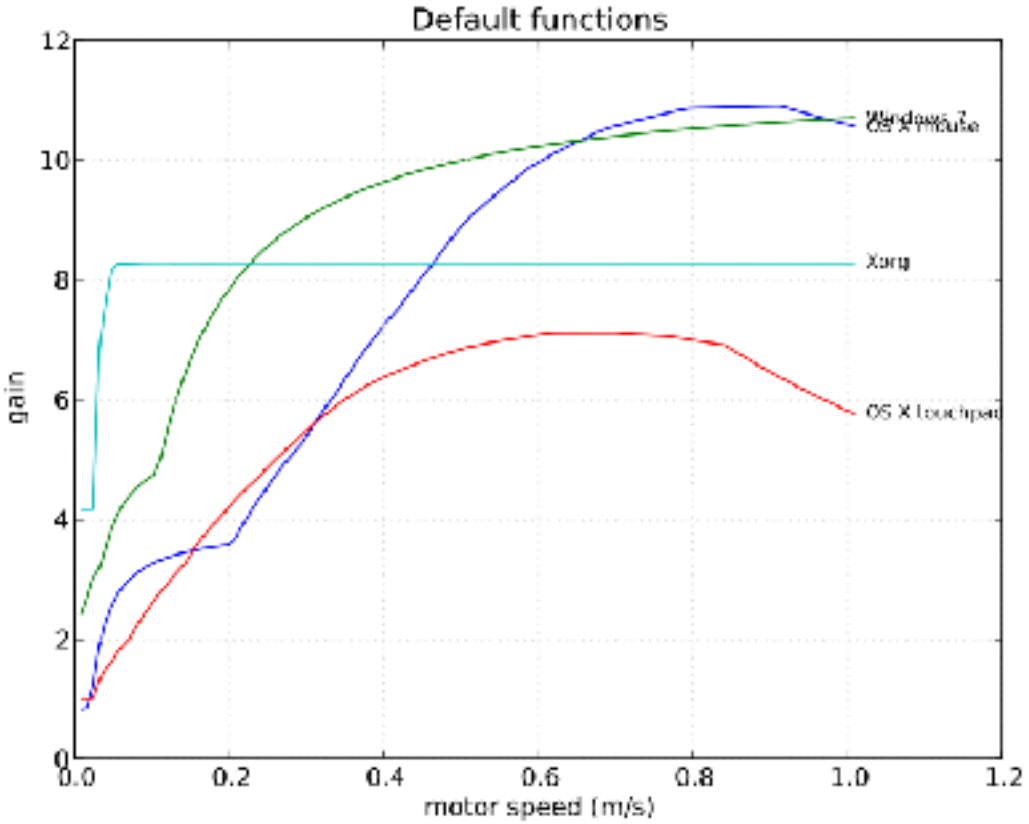
Phase corrective

Lente et précise

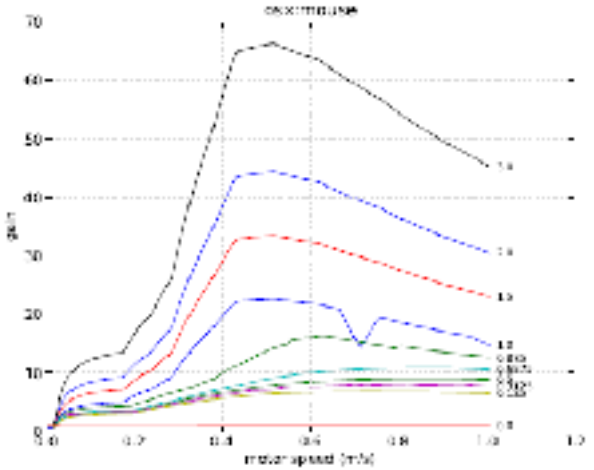
⇒ gain faible

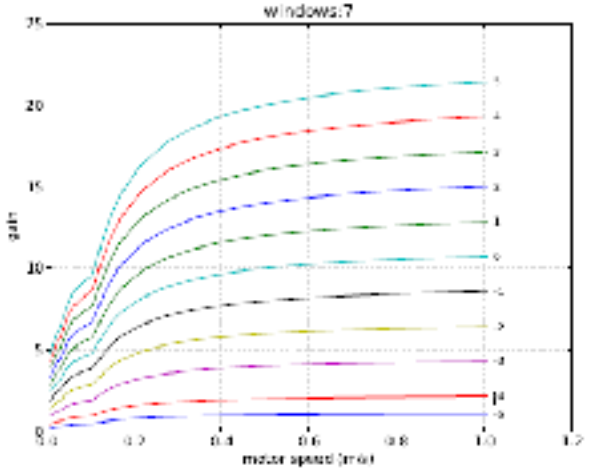


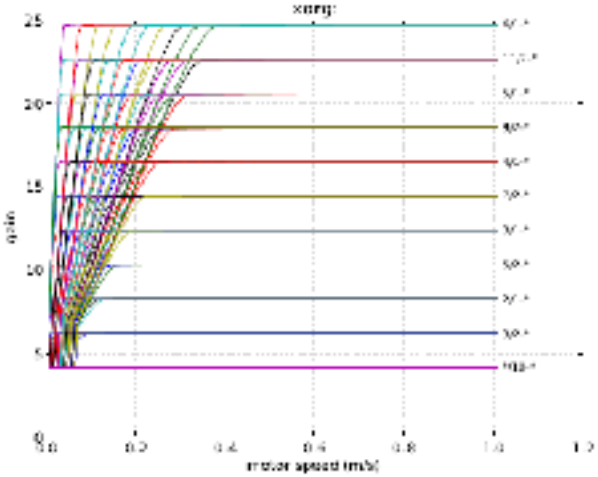
David E. Meyer , et al., 1988
Optimality in human motor performance: ideal control of rapid aimed movements
Psychological Review, 95, 340-370

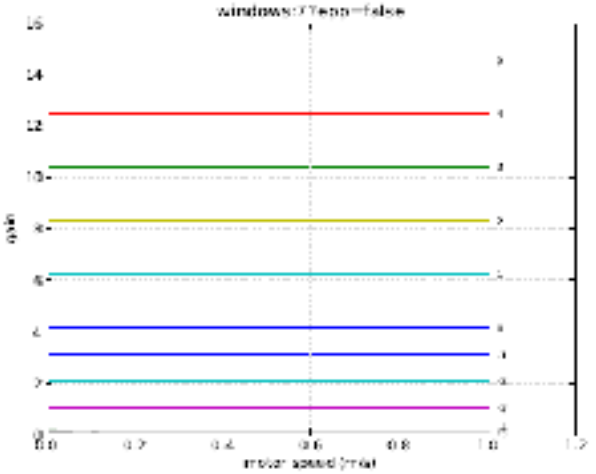


Géry Casiez and Nicolas Roussel, 2011
No more bricolage! Methods and tools to characterize, replicate and compare pointing transfer functions
UIST'11, 603-614

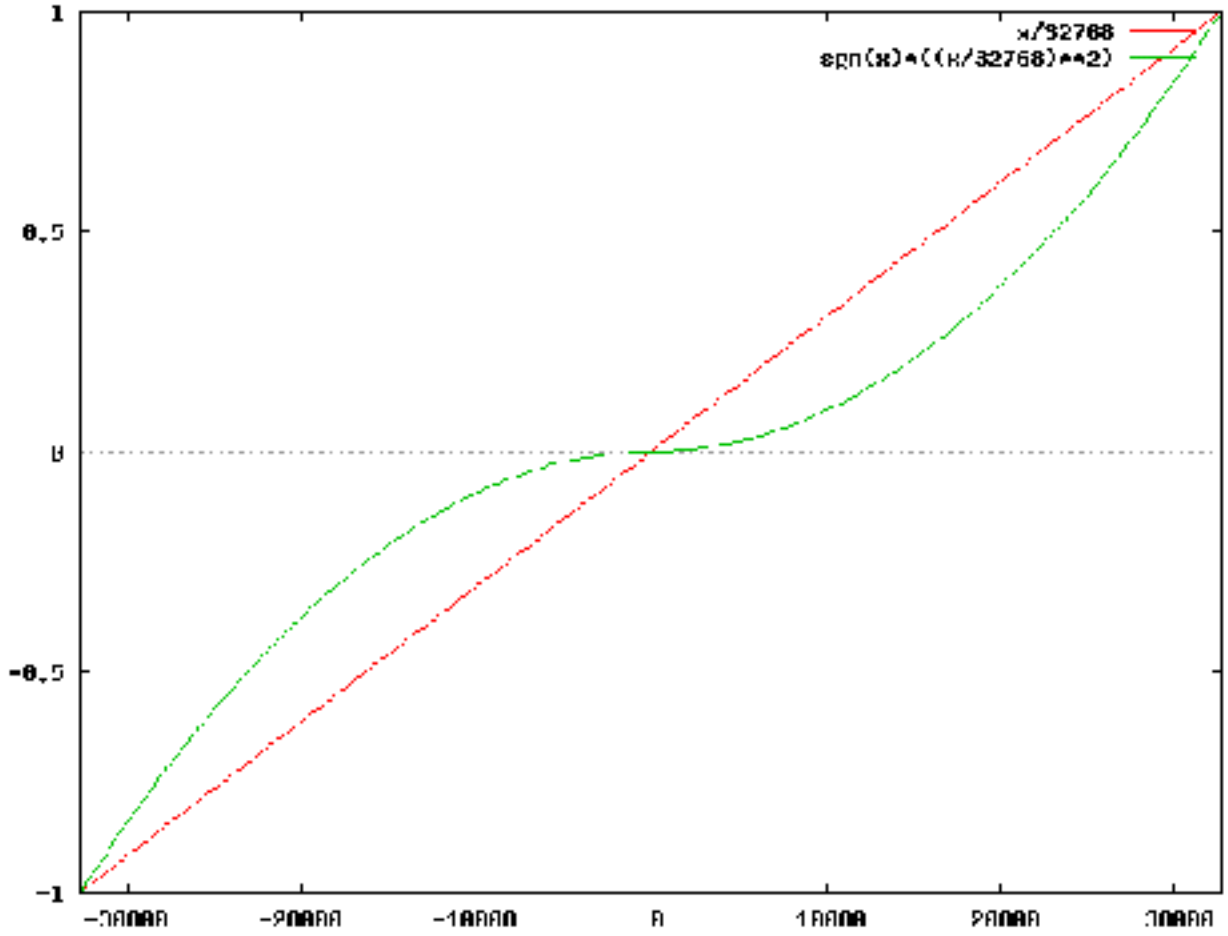








www.libpointing.org


x/32768
sgn(x)*((x/32768)**2)
1
0.5
0
-0.5
-1
-30000
-20000
-10000
0
10000
20000
30000

PROGRAMMATION

Interroger le périphérique en boucle

+ Simple
- Gaspille le CPU
- Gaspille de l’énergie
- Peu précis

```
int main()
{
  while (true)
  {
    input();
    update();
    draw();
  }
}
```

Réagir aux inputs

+ Précis
+ Consomme peu de CPU
+ Consomme peu d'énergie
- Compliqué

```
void onKeyDown(…)
{
  …
}

void onMouseMove(…)
{
  …
}

int main()
{
  while (true)
  {
    update();
    draw();
  }
}
```