

# Tracker: cargar video y seguir objeto

Tracker

File Edit Video Track Coordinate System View Help

Now available: version 6.0.1

mass A m 1.000 kg

Track Control

New mass A

Autotracker: mass A position

Search Step Back Search Next

Frame 30: Template Match

Template: Evolve 20% Tether 5% Automark 4

Search:  X-axis Only  Look Ahead  Autoskip

Target: Track mass A Point position

Frame 30 (match score 8.6): The match was marked automatically.

Help Show Key Frame Delete Close

mass A selected (Get mass on toolbar, shift-click to re-mark highlighted position)

030 100%

BallTossOut.mov

Plots mass A Sync

mass A (t, x)

mass A (t, y)

Columns mass A

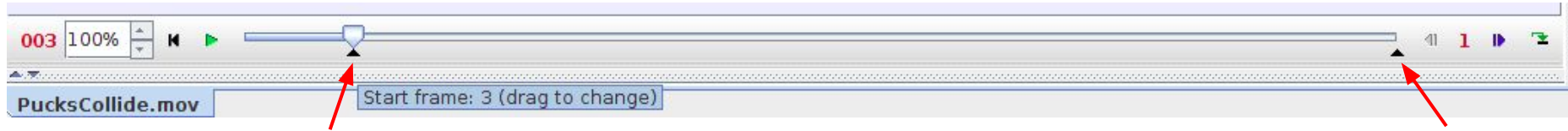
t (s)	x (m)	y (m)
0.000	5.852E-2	-2.617E-2
0.033	0.109	3.884E-2
0.067	0.160	9.360E-2
0.100	0.211	0.138
0.133	0.263	0.174
0.167	0.314	0.199
0.200	0.365	0.215
0.233	0.418	0.220
0.267	0.468	0.217
0.300	0.520	0.203
0.333	0.571	0.180
0.367	0.624	0.145
0.400	0.675	0.103
0.433	0.728	4.901E-2
0.467	0.780	-1.495E-2
0.500	0.833	-8.923E-2
0.533	0.885	-0.174
0.567	0.939	-0.269
0.600	0.992	-0.374
0.633	1.047	-0.490
0.667	1.103	-0.618

# Tracker: cargar video y seguir objeto

(1) Importar video

Archivo/Importar/Video

(2) Seleccionar rango de interés del video → Cuadro inicial y final

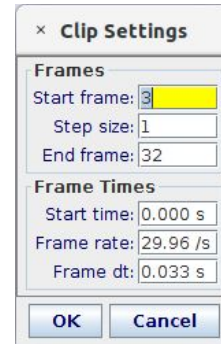


(3) Verificar paso temporal entre cuadros

Video/Ajustes del Corte [Clip settings]

o

Click derecho sobre video → *Clip Settings*

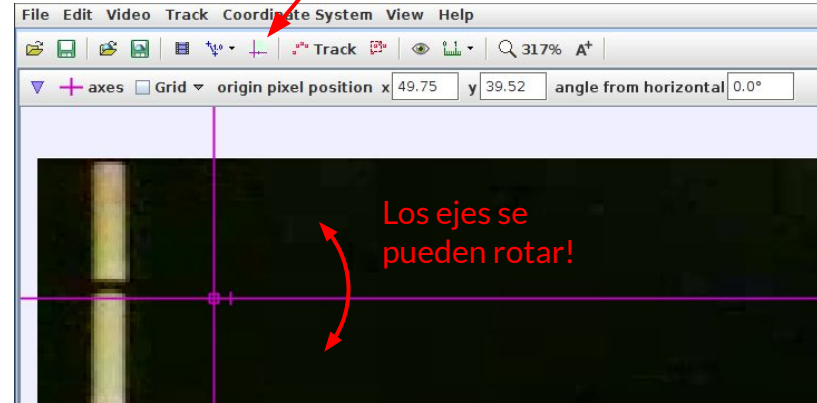


Frame rate = cuadros por segundo (frecuencia!)

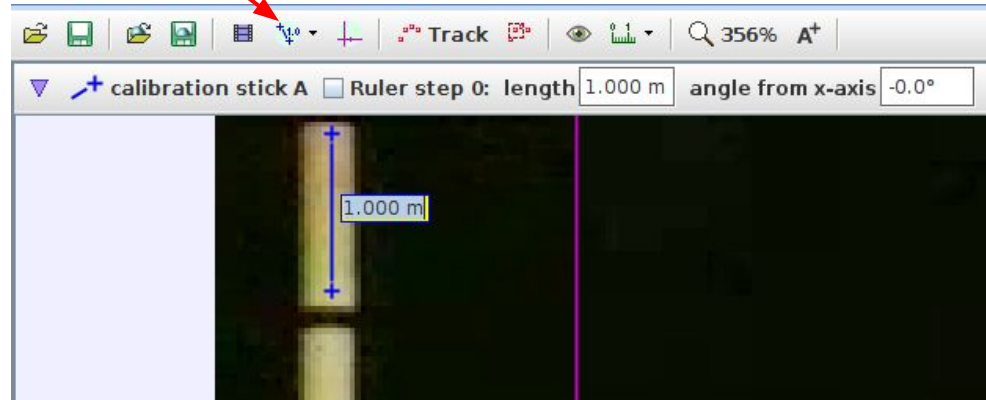
Frame dt = paso temporal =  $1/\text{frame rate}$

# Tracker: cargar video y seguir objeto

(4) Fijar sistema de referencia



(5) Definir escala



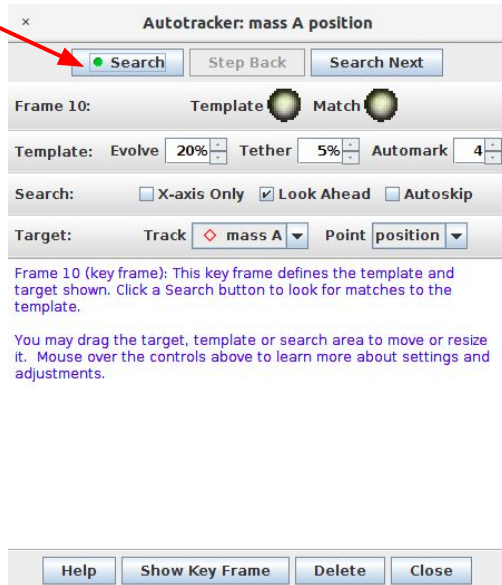
# Tracker: cargar video y seguir objeto

(6) Identificar objeto a seguir y trackear

*Track/Nuevo/Punto de masa*

→ *shift + ctrl + click* en 'centro' del objeto de interés

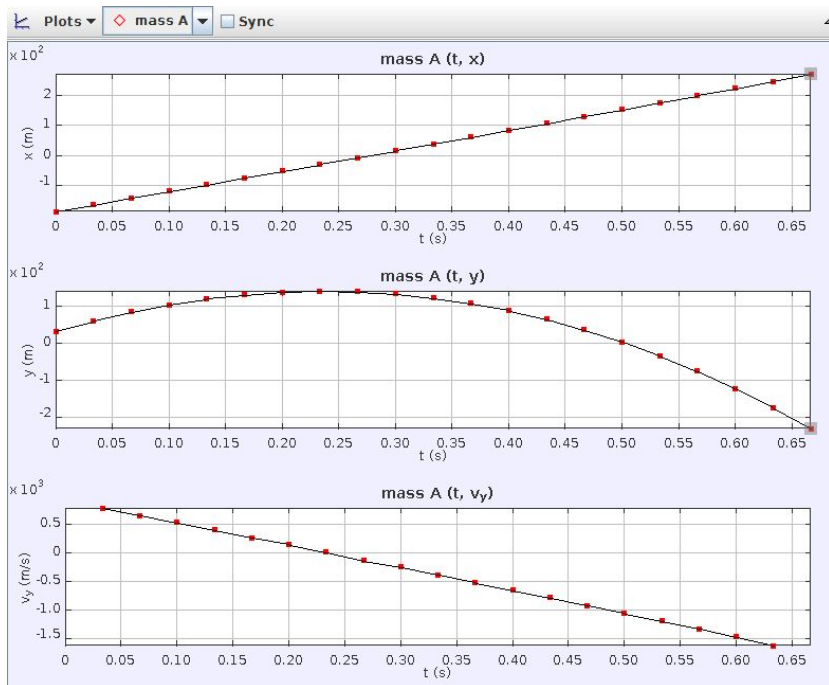
Seguir objeto en todos los cuadros de interés del video



Región de búsqueda



# Tracker: visualizar datos

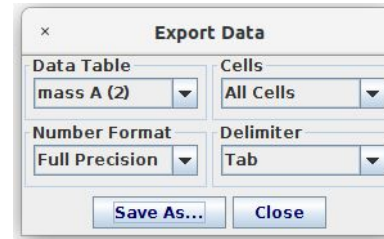


Columns: mass A

$t$ (s)	$x$ (m)	$y$ (m)	$v_x$ (m/s)	$v_y$ (m/s)
0.000	-189.0	32.41		
0.033	-166.3	61.23	677.1	797.2
0.067	-143.8	85.56	679.6	664.6
0.100	-121.0	105.5	683.4	537.3
0.133	-98.30	121.3	681.8	406.8
0.167	-75.61	132.6	680.2	273.3
0.200	-52.95	139.6	684.7	153.5
0.233	-29.96	142.9	686.9	14.25
0.267	-7.161	140.5	686.0	-127.1
0.300	15.77	134.4	688.8	-245.5
0.333	38.76	124.1	691.1	-378.1
0.367	61.85	109.2	691.5	-517.3
0.400	84.86	89.70	692.4	-652.1

# Exportar datos y visualizarlos en Python

Contenidos del archivo exportado 'data.txt' [nombre arbitrario!]



mass_A	x	y	v_{x}	v_{y}	a_{x}	a_{y}
t	0.05852191387806602	-0.02617357461012726	NaN	NaN	NaN	NaN
0.0	0.1095763629282316	0.038840941022693365	1.5260831581858385	1.7966683040331852	NaN	NaN
0.03333333333333331	0.1602607910904553	0.09360431232541849	1.5315128220147798	1.497159486071346	0.16464495311572436	-8.773151890094258
0.06666666666666668	0.21167721772921697	0.13865157342744983	1.5411774356916241	1.2120302839391508	0.11379412025806147	-8.743942324772553
0.1	0.26300595346989697	0.17440633125469523	1.537493130627552	0.9174024092320537	-0.12601569826881845	-8.8936949032829
0.13333333333333336	0.31417675977105386	0.19981173404292008	1.532642049162853	0.6156112517883282	0.6016060708509395	-9.15388593860285
0.16666666666666669						

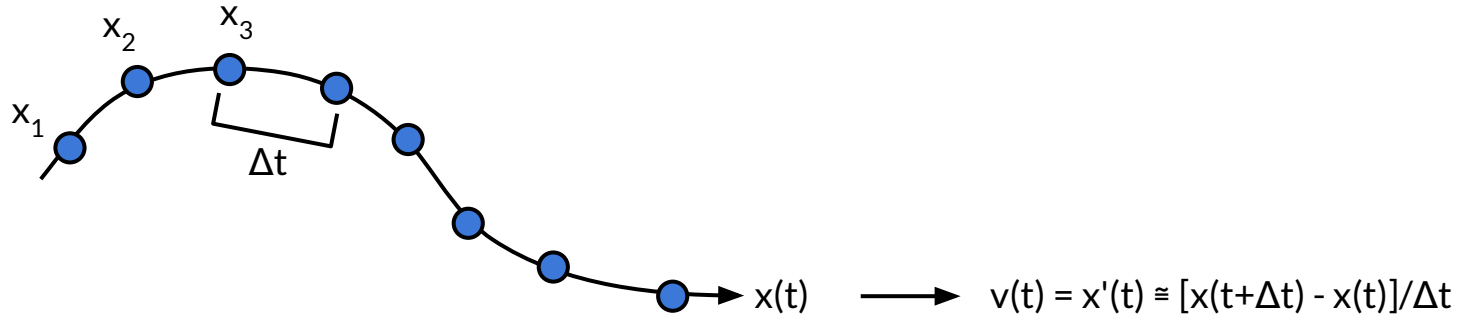
```
import numpy as np
```

```
archivo = 'data.txt'  
data = np.loadtxt(archivo, skiprows=2)
```

```
t = data[:,0]  
x = data[:,1]  
y = data[:,2]
```

# Tracker: calculando a(t)

¿Cómo calcula la aceleración Tracker? ¿Es confiable?



$$v(t) = x'(t) \cong [x(t+\Delta t) - x(t)]/\Delta t$$

$$x_n = x(t_n) \longrightarrow v_{n+1} \cong [x(t_{n+1}) - x(t_n)]/\Delta t \longrightarrow a(t) = v'(t) \dots$$

¿De qué otra manera podría estimarse a(t)?