

Kapitel 4 Biblioteksmodulen ti_plotlib

Successiva positioner hos rörelse

Vid tillämpningen i kapitel 4 kommer du att undersöka en rörelse hos en golfboll utifrån mätningar med ett stroboskop och därefter ska du bygga en grafisk modell av rörelsen med programmering i Python, och med hjälp av biblioteksmodulen ti_plotlib.

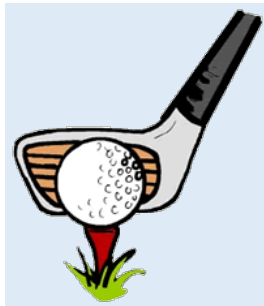
Syfte:

- Grafiskt representera en datamängd
- Beräkna och representera vektorer i ett system som modelleras av en punkt

Problemet: Förflyttningen hos en golfboll studeras från en inspelning av rörelsen där man tog bilder med ett visst tidsintervall. Vi vill nu grafiskt visa hastighetsvektorns förändring över tiden.

Bollens position visas i tabellen med tidsintervallet 0,066 s.

I kapitel 5 får du lära dig hur du importerar data till Python-appen från räknarens listor med hjälp av modulen ti_system.



t(s)	0	0.066	0.132	0.198	0.264	0.33	0.396	0.462	0.528	0.594	0.66
x (m)	0.01	0.25	0.57	0.91	1.22	1.54	1.87	2.16	2.49	2.81	3.15
y (m)	0.015	0.34	0.681	1.01	1.297	1.559	1.768	1.95	2.08	2.158	2.193

Komma igång:

- Starta ett nytt skript och döp det till KAP4APPL
- Importera sedan de grafiska biblioteksmodulen ti_plotlib
- Mata nu in mätresultaten för x och y

Beräkna hastighetsvektorerna

Hastighetsvektorn vid tiden t ges av uttrycket

$$\vec{V}_i = \frac{M_i M_{i+1}}{t_{i+1} - t_i}$$

Du kan inte rita hastighetsvektorn från den sista punkten i listan, det måste du ta hänsyn till i koden.

Tips: För att göra den grafiska representationen läsbar har vi en skalfaktor på 2.

Kommentar: Instruktionen len(x) gör att du kan erhålla antalet objekt i listan x. len finns i list-menyn.

Instruktionen .append lägger till ett objekt i en lista. Exempel:

```
>>>listA = [2,4,6,8]
>>>listA.append(10)
>>>print(listA)
[2,4,6,8,10]
```

Inställning av parametrar för den grafiska representationen:

```
EDITOR: KAP4APPL
PROGRAM LINE 0006
import ti_plotlib as plt
#data
dt=0.066
x=[0.01,0.25,0.57,0.91,1.22,1.54
,1.87,2.16,2.49,2.81,3.15]
y=[0.015,0.34,0.681,1.01,1.297,1
.559,1.768,1.95,2.08,2.158,
2.193]
#hastighetsvektorer_
vx=[]
vy=[]
```

```
EDITOR: KAP4APPL
PROGRAM LINE 0014
.559,1.768,1.95,2.08,2.158,
2.193]
#hastighetsvektorer
vx=[]
vy=[]
n=len(x)
for i in range(0,n-1):
**vx.append((x[i+1]-x[i])/dt)
**vy.append((y[i+1]-y[i])/dt)
skala=2
#grafisk representation
```

- `plt.cls()` för att rensa skärmen
- `plt.title("titel")` lägga till en titel till den grafiska bilden.
- `plt.window(x_min, x_max, y_min, y_max)` ställa in fönstermått
- `plt.grid(xscl, yscl, "type")` visa ett rutnät med en viss gradering
- `plt.color(255, g, 255)` ställa in färg för visning av punkterna i banan. Här är inställningen en lika blandning av rött och blått, dvs magenta.
- `plt.scatter(xlist, ylist, "type")` visning av data som ett spridningsdiagram
- `plt.color(0, 0, 0)` visning av axlar helt i svart.
- `plt.pen("medium", "solid")` axlar som mellantjocka linjer.
- `plt.labels("x(m)", "y(m)")` visning av axelbeteckningar. Visas på rad 2 och 12 som förinställd standard.

Vektorspårningen görs som en avgränsad loop med $n-1$ värden.

- `plt.line(x0, y0, x1, y1)` för spårning av hastighetsvektorn
- `plt.show()` visning av hela den grafiska representationen.

Påminnelse: Bollens placering mellan tidpunkterna t_1 och t_2 kan identifieras i ett kartesiskt koordinatsystem med $x_i + vx_i \times dt$ för x -axeln och $y_i + vy_i \times dt$ för y -axeln.

Kör nu skriptet. Du bör få en grafisk representation som på skärmen här.

Kommentarer: Genom att trycka på **F3-knappen** har du tillgång till en verktygslåda med instruktioner för att bl.a. kopiera och klistra in för att göra det enklare att redigera skriptet.

För komplexa skript har du också möjlighet att duplicera ett skript. För att göra detta, visa skriptlistan, placera markören framför skriptnamnet och för att duplicera, trycker du sedan på **f5 (Manage)** och välj i menyn **1: Replicate Program**.

Du kommer att bli tillfrågad om ett nytt namn. Om antalet data är stort eller kommer från ett experiment med en konsol för infångning av data är det möjligt att använda TI Connect CE-programvaran för att importera data i **CSV-format**.

```
EDITOR: KAP4APPL
PROGRAM LINE 0024
#grafisk representation
plt.cls()
plt.title("bana för golfboll")
plt.window(-1,4.5,-1,3,)
plt.grid(0.5,0.5,"dot")
plt.color(255,0,255)
plt.scatter(x,y,"o")
plt.color(0,0,0)
plt.pen("medium","solid")
plt.axes("on")
plt.labels("x(m)","y(m)")_
Fns... a A # Tools Run Files
```

```
EDITOR: KAP4APPL
PROGRAM LINE 0030
plt.pen("medium","solid")
plt.axes("on")
plt.labels("x(m)","y(m)")
for i in range(0,n-1):
    plt.pen("thin","solid")
    plt.line(x[i],y[i],x[i]+vx[i]*
            dt/skala,y[i]+vy[i]*dt/skal
            a,"arrow")
plt.show_plot()
Fns... a A # Tools Run Files
```

