Segundo parcial

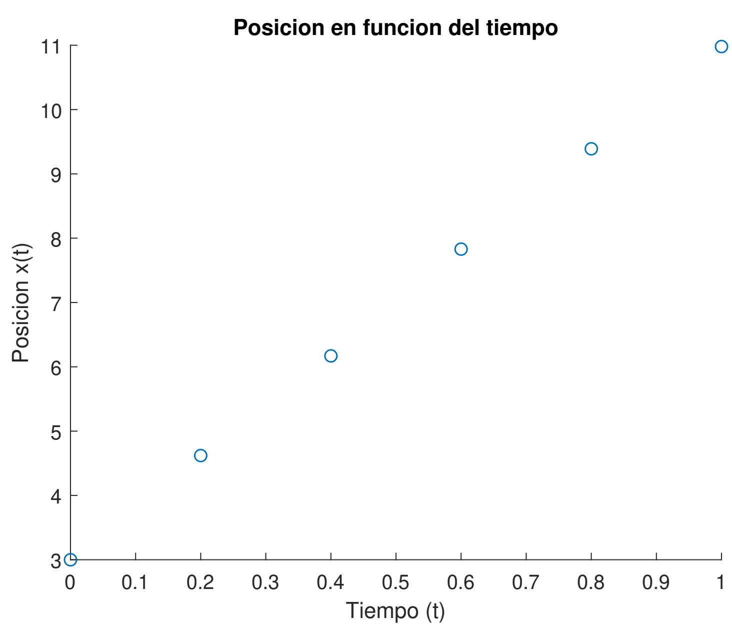
Estudiante: CABALLERO BURGOA, Carlos Eduardo
Carrera: Ingeniería Electromecánica
Correo: cijkb.j@gmail.com

1. La tabla a continuación presenta datos de posición en función del tiempo. Indique a que clase de movimiento corresponde, graficar y encuentre la ecuación empírica con su respectivo error.

i	$t_i[s]$	$x_i[m]$
1	0.0	3.00
2	0.2	4.62
3	0.4	6.17
4	0.6	7.83
5	0.8	9.39
6	1.0	10.98

Solución:

Se obtiene el siguiente gráfico:



Por tanto corresponde a un movimiento con $a = 0$, es decir **movimiento uniforme**.

Calculando los valores de la recta por el método de los mínimos cuadrados, se obtiene:

$$A = (3.00 \pm 0.02)[m]; 0.59\,\%$$

$$B = (7.98 \pm 0.03)[m/s]; 0.37\,\%$$

Con los parámetros obtenidos la relación $x = x(t)$ es:

$$x = 3 + 7.98t \tag{1}$$

Memoria de calculo:

```
# Datos importados (i1.csv):
0.0,3.00
0.2,4.62
0.4,6.17
0.6,7.83
0.8,9.39
1.0,10.98


# Comandos ejecutados (p1b.m):
% leer datos previamente formateados
table = readtable('i1.csv')

% asignacion de variables
x = table.Var1
y = table.Var2

% tamano de la muestra
n = length(x)

% calculo de las sumatorias
sx = sum(x)
sy = sum(y)
sxx = sum(x.*x)
sxy = sum(x.*y)

% Calculo de los valores de la recta
D = (n * sxx) - (sx)^2
A = ( (sy * sxx) - (sxy * sx) ) / D
B = ( (n * sxy) - (sx * sy) ) / D

% Calculo del error
Y = A + (B * x)
d = y - Y

sdd = sum(d.*d)
s2 = sdd / ( n - 2)

sA = sqrt( (s2 * sxx) / D )
sB = sqrt( (s2 * n) / D )

%calculando el error porcentual
```

```
EA = (sA / A) * 100
EB = (sB / B) * 100


# Salida del programa (o1b.txt):
table =
  6x2 table
    Var1    Var2
    ----    -----

      0         3
    0.2      4.62
    0.4      6.17
    0.6      7.83
    0.8      9.39
      1     10.98

x =
         0
    0.2000
    0.4000
    0.6000
    0.8000
    1.0000

y =
    3.0000
    4.6200
    6.1700
    7.8300
    9.3900
   10.9800

n = 6
sx = 3
sy = 41.9900
sxx = 2.2000
sxy = 26.5820
D = 4.2000
A = 3.0076
B = 7.9814

Y =
    3.0076
    4.6039
    6.2002
    7.7965
    9.3928
   10.9890

d =
   -0.0076
    0.0161
   -0.0302
    0.0335
   -0.0028
```

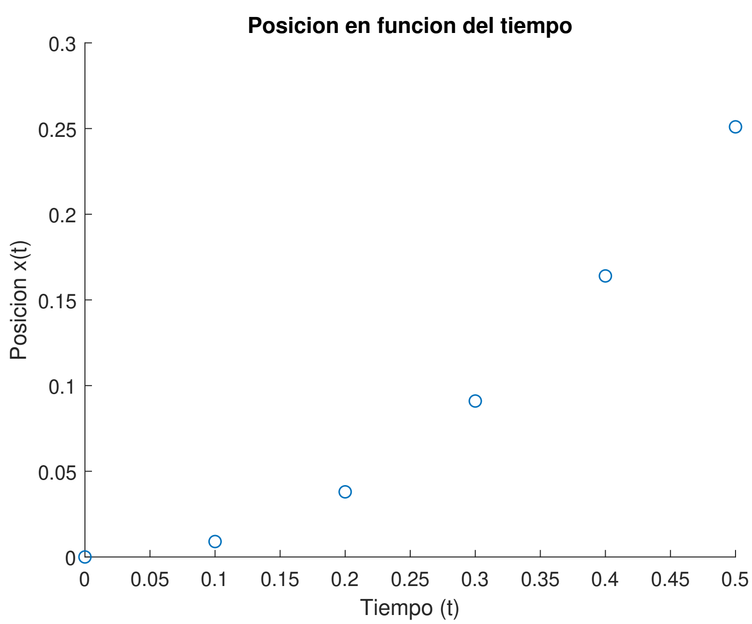
```
   -0.0090

sdd = 0.0024
s2 = 6.1048e-04
sA = 0.0179
sB = 0.0295
EA = 0.5946
EB = 0.3700
```

2. La tabla a continuación presenta datos de posición en función del tiempo. Indique a que clase de movimiento corresponde, graficar y encuentre la ecuación empírica con su respectivo error.

i	$t_i[s]$	$x_i[m]$
1	0.0	0.000
2	0.1	0.009
3	0.2	0.038
4	0.3	0.091
5	0.4	0.164
6	0.5	0.251

Solución:

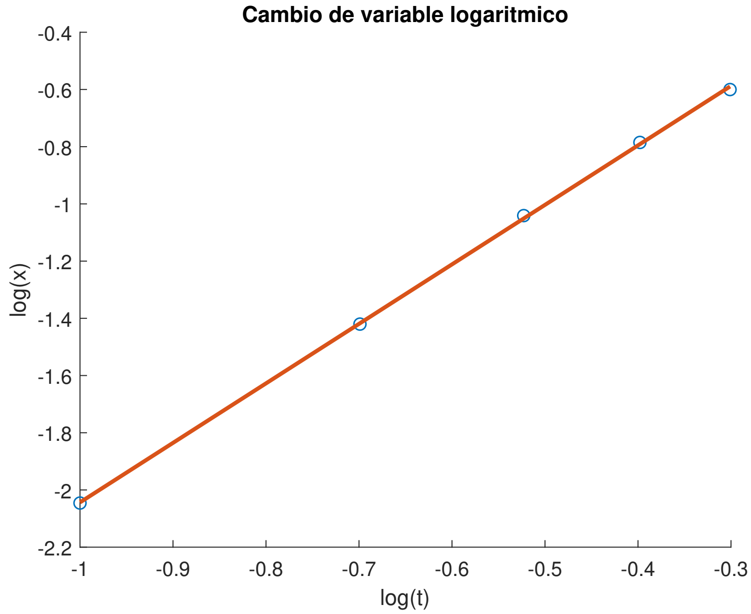
Se obtiene el siguiente gráfico:



Por tanto corresponde a un movimiento con $a \neq 0$, es decir **movimiento uniformemente acelerado**.

Aplicando linealización por logaritmos:

i	$\log(x_i)$	$\log(t_i)$
1	-	-
2	-1.0000	-2.0458
3	-0.6990	-1.4202
4	-0.5229	-1.0410
5	-0.3979	-0.7852
6	-0.3010	-0.6003



Memoria de calculo:

```
# Datos importados (i2.csv):
0.0,0.000
0.1,0.009
0.2,0.038
0.3,0.091
0.4,0.164
0.5,0.251


# Comandos ejecutados (p2b.m):
% leer datos previamente formateados
table = readtable('i2.csv')

% cambio de variable
X = log10(table.Var1(2:end))
Y = log10(table.Var2(2:end))

% calcular la ecuacion de la recta
p = polyfit(X, Y, 1)
v = polyval(p, X)

% personalizar grafica
title('Cambio de variable logaritmico')
xlabel('log(t)')
ylabel('log(x)')

% graficar puntos y lineas
hold on
plot(X, Y, 'o')
plot(X, v, 'LineWidth', 2)
hold off
```

```
# Salida del programa (o2b.txt):
table =
  6x2 table
    Var1    Var2
    ----    -----

      0         0
    0.1     0.009
    0.2     0.038
    0.3     0.091
    0.4     0.164
    0.5     0.251

X =
   -1.0000
   -0.6990
   -0.5229
   -0.3979
   -0.3010

Y =
   -2.0458
   -1.4202
   -1.0410
   -0.7852
   -0.6003

p = 2.0800    0.0366

v =
   -2.0434
   -1.4173
   -1.0510
   -0.7911
   -0.5896
```

Calculando los valores de la recta por el método de los mínimos cuadrados, se obtiene:

$$A = (0.03 \pm 0.01)[m]; 29.63\,\%$$

$$B = (2.08 \pm 0.02)[m/s]; 0.82\,\%$$

La ecuación de la recta es:

$$Y' = 0.03 + 2.08X'$$

A partir de los parámetros de recta A y B, calculamos los parámetros a y b, de la curva original y sus errores por el método de propagación de errores:

$$a = antilog(A) = antilog(0.03) = 1.08$$
$$b = B = 2.08$$
$$e_a = 10^A ln(10)e_A = 10^{(0.03)} ln(10)0.01 = 0.02$$
$$e_b = e_B = 0.02$$

Obteniendo finalmente los valores de la curva:

$$a = (1.08 \pm 0.02)[m/s^2]; 2.49\,\%$$

$$b = (2.08 \pm 0.02)[u]; 0.82\,\%$$

La ecuación de la curva resultante es:

$$x = 1.08t^2 \tag{2}$$

Memoria de calculo:

```
# Datos importados (i2.csv):
0.0,0.000
0.1,0.009
0.2,0.038
0.3,0.091
0.4,0.164
0.5,0.251


# Comandos ejecutados (p2c.m):
% leer datos previamente formateados
table = readtable('i2.csv')

% asignacion de variables
x = log10(table.Var1(2:end))
y = log10(table.Var2(2:end))

% tamano de la muestra
n = length(x)

% calculo de las sumatorias
sx = sum(x)
sy = sum(y)
sxx = sum(x.*x)
sxy = sum(x.*y)

% Calculo de los valores de la recta
D = (n * sxx) - (sx)^2
```

```
A = ( (sy * sxx) - (sxy * sx) ) / D
B = ( (n * sxy) - (sx * sy) ) / D

% Calculo del error
Y = A + (B * x)
d = y - Y

sdd = sum(d.*d)
s2 = sdd / ( n - 2)

sA = sqrt( (s2 * sxx) / D )
sB = sqrt( (s2 * n) / D )

%calculando el error porcentual
EA = (sA / A) * 100
EB = (sB / B) * 100

%calculando los valores originales
a = 10^A
b = B

sa = (10^A) * log(10) * sA
sb = sB

%calculando el error porcentual
Ea = (sa / a) * 100
Eb = (sb / b) * 100


# Salida del programa (o2c.txt):
table =
  6x2 table

    Var1    Var2
    ----    -----

      0        0
    0.1     0.009
    0.2     0.038
    0.3     0.091
    0.4     0.164
    0.5     0.251

x =
   -1.0000
   -0.6990
   -0.5229
   -0.3979
   -0.3010

y =
   -2.0458
   -1.4202
   -1.0410
   -0.7852
   -0.6003
```

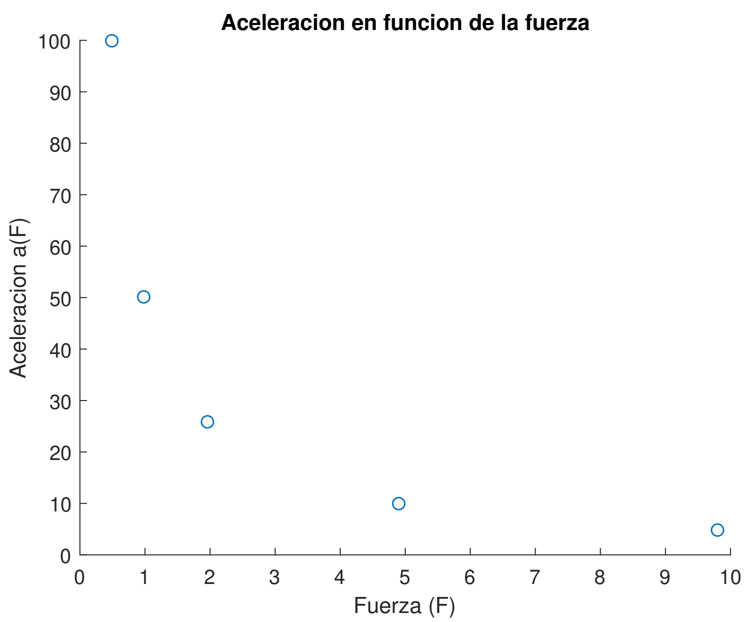
```
n = 5
sx = -2.9208
sy = -5.8924
sxx = 2.0109
sxy = 4.0759
D = 1.5235
A = 0.0366
B = 2.0800

Y =
   -2.0434
   -1.4173
   -1.0510
   -0.7911
   -0.5896

d =
   -0.0023
   -0.0029
    0.0101
    0.0060
   -0.0108

sdd = 2.6670e-04
s2 = 8.8899e-05
sA = 0.0108
sB = 0.0171
EA = 29.6288
EB = 0.8212
a = 1.0878
b = 2.0800
sa = 0.0271
sb = 0.0171
Ea = 2.4943
Eb = 0.8212
```

3. La tabla presenta datos de aceleración (a^*) en función de la fuerza (F) para una práctica de dinámica. Determine la ecuación empírica aplicando MMC. Obtenga además el valor de la fuerza con su respectivo error.

i	$F_i[s]$	$a_i^*[m]$
1	0.49	99.92
2	0.98	50.14
3	1.96	25.85
4	4.90	9.96
5	9.80	4.82

Solución:

Se obtiene el siguiente gráfico:

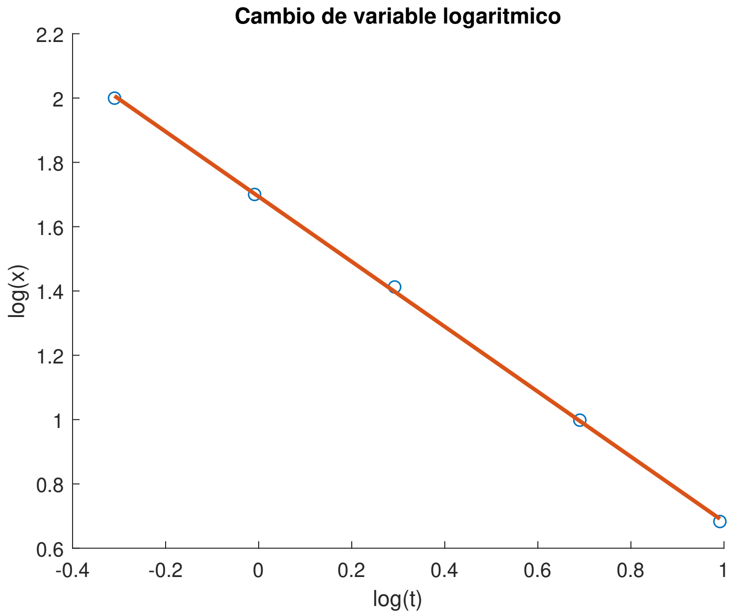


Por lo tanto la relación entre fuerza (F) y aceleración (a), es inversamente proporcional.

$$a \propto \frac{1}{F}$$

Aplicando linealización por logaritmos:

i	$\log(F_i)$	$\log(a_i)$
1	-0.3098	1.9997
2	-0.0088	1.7002
3	0.2923	1.4125
4	0.6902	0.9983
5	0.9912	0.6830



Memoria de calculo:

```
# Datos importados (i3.csv):
0.49,99.92
0.98,50.14
1.96,25.85
4.90,9.96
9.80,4.82


# Comandos ejecutados (p3b.m):
% leer datos previamente formateados
table = readtable('i3.csv')

% cambio de variable
X = log10(table.Var1)
Y = log10(table.Var2)

% calcular la ecuacion de la recta
p = polyfit(X, Y, 1)
v = polyval(p, X)

% personalizar grafica
title('Cambio de variable logaritmico')
xlabel('log(t)')
ylabel('log(x)')

% graficar puntos y lineas
hold on
plot(X, Y, 'o')
plot(X, v, 'LineWidth', 2)
hold off
```

```
# Salida del programa (o3b.txt):
table =
  5x2 table

    Var1    Var2
    ----    -----

    0.49    99.92
    0.98    50.14
    1.96    25.85
     4.9     9.96
     9.8     4.82

X =
   -0.3098
   -0.0088
    0.2923
    0.6902
    0.9912

Y =
    1.9997
    1.7002
    1.4125
    0.9983
    0.6830

p = -1.0109    1.6933

v =
    2.0065
    1.7022
    1.3979
    0.9956
    0.6913
```

Calculando los valores de la recta por el método de los mínimos cuadrados, se obtiene:

$$A = (1.69 \pm 0.006)[m]; 0.34\,\%$$

$$B = (-1.01 \pm 0.01)[m/s]; 1.00\,\%$$

La ecuación de la recta es:

$$Y' = 1.69 - 1.01X'$$

A partir de los parámetros de recta A y B, calculamos los parámetros a y b, de la curva original y sus errores por el método de propagación de errores:

$$a = antilog(A) = antilog(1.68) = 49.36$$
$$b = B = -1.01$$
$$e_a = 10^A ln(10) e_A = 10^{(1.69)} ln(10) 0.006 = 0.66$$
$$e_b = e_B = 0.01$$

Obteniendo finalmente los valores de la curva:

$$a = (49.36 \pm 0.66)[u]; 1.34\,\%$$

$$b = (-1.01 \pm 0.01)[u]; 1.00\,\%$$

La ecuación de la curva resultante es:

$$a = \frac{49.36}{F} \tag{3}$$

Memoria de calculo:

```
# Datos importados (i3.csv):
0.49,99.92
0.98,50.14
1.96,25.85
4.90,9.96
9.80,4.82


# Comandos ejecutados (p3c.m):
% leer datos previamente formateados
table = readtable('i3.csv')

% asignacion de variables
x = log10(table.Var1)
y = log10(table.Var2)

% tamano de la muestra
n = length(x)

% calculo de las sumatorias
sx = sum(x)
sy = sum(y)
sxx = sum(x.*x)
sxy = sum(x.*y)

% Calculo de los valores de la recta
D = (n * sxx) - (sx)^2
A = ( (sy * sxx) - (sxy * sx) ) / D
```

```
B = ( (n * sxy) - (sx * sy) ) / D

% Calculo del error
Y = A + (B * x)
d = y - Y

sdd = sum(d.*d)
s2 = sdd / ( n - 2)

sA = sqrt( (s2 * sxx) / D )
sB = sqrt( (s2 * n) / D )

%calculando el error porcentual
EA = (sA / A) * 100
EB = (sB / B) * 100

%calculando los valores originales
a = 10^A
b = B

sa = (10^A) * log(10) * sA
sb = sB

%calculando el error porcentual
Ea = (sa / a) * 100
Eb = (sb / b) * 100


# Salida del programa (o3c.txt):
table =
  5x2 table

    Var1    Var2
    ----    -----

    0.49    99.92
    0.98    50.14
    1.96    25.85
     4.9     9.96
     9.8     4.82

x =
   -0.3098
   -0.0088
    0.2923
    0.6902
    0.9912

y =
    1.9997
    1.7002
    1.4125
    0.9983
    0.6830

n = 5
```

```
sx = 1.6551
sy = 6.7936
sxx = 1.6404
sxy = 1.1444
D = 5.4625
A = 1.6933
B = -1.0109

Y =
    2.0065
    1.7022
    1.3979
    0.9956
    0.6913

d =
   -0.0069
   -0.0020
    0.0146
    0.0026
   -0.0083

sdd = 3.3856e-04
s2 = 1.1285e-04
sA = 0.0058
sB = 0.0102
EA = 0.3438
EB = -1.0054
a = 49.3564
b = -1.0109
sa = 0.6616
sb = 0.0102
Ea = 1.3405
Eb = -1.0054
```