

# Movimiento a través de una curva

Dan Casas

# Movimiento a través de una curva

## 1. Introducción

Una vez definida la curva sobre la cual queremos movernos, el siguiente paso es definir ese **movimiento**.

Este movimiento se realiza mediante incrementos en el parámetro que define la curva realizado por el animador, de manera que obtengamos los resultados esperados.

Debemos conocer la **relación** existente en el cambio entre el **parámetro de la función** y la **distancia recorrida** a lo largo de la curva.

# Movimiento a través de una curva

Necesitamos

- 1) Método para moverse mediante escalones por la curva con incrementos iguales.
- 2) Métodos que nos permitirán acelerar o aminorar nuestro movimiento por la curva definida.

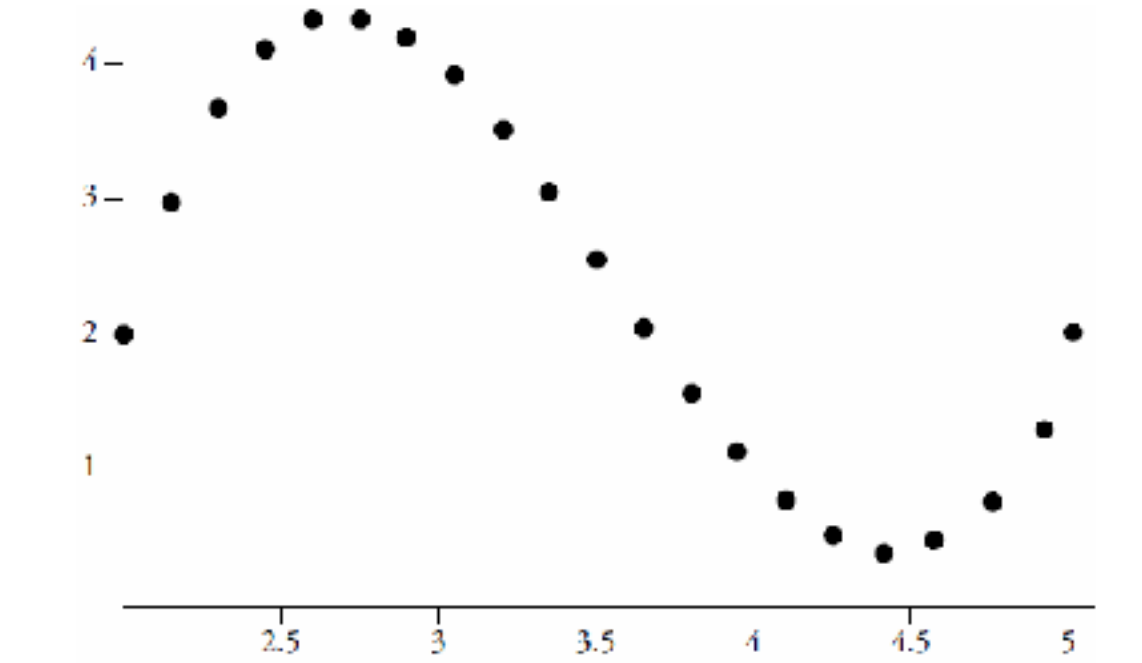
Asumimos que ha sido elegido un método de interpolación de los vistos anteriormente.

# Movimiento a través de una curva

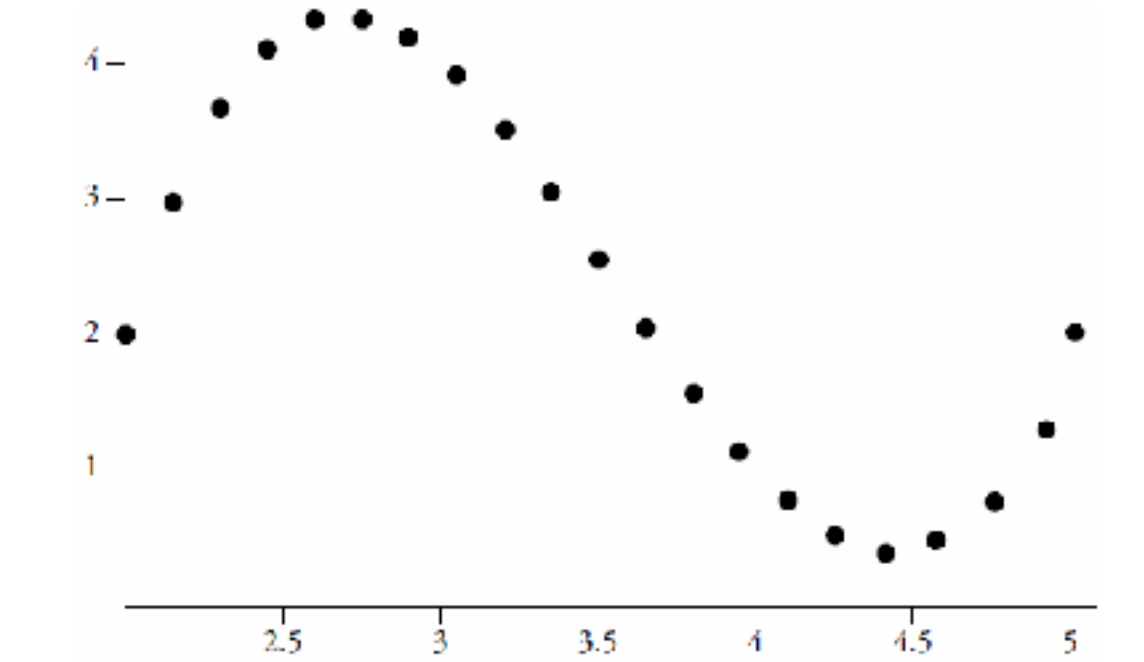
Si queremos asegurar esa velocidad constante, debemos considerar la distancia real recorrida, es decir, la función de interpolación debe ser parametrizada con la longitud del arco, la distancia a lo largo de la curva.

Ejemplo de puntos producidos por incrementos iguales de un parámetro de interpolación para una curva cúbica típica.

# Movimiento a través de una curva

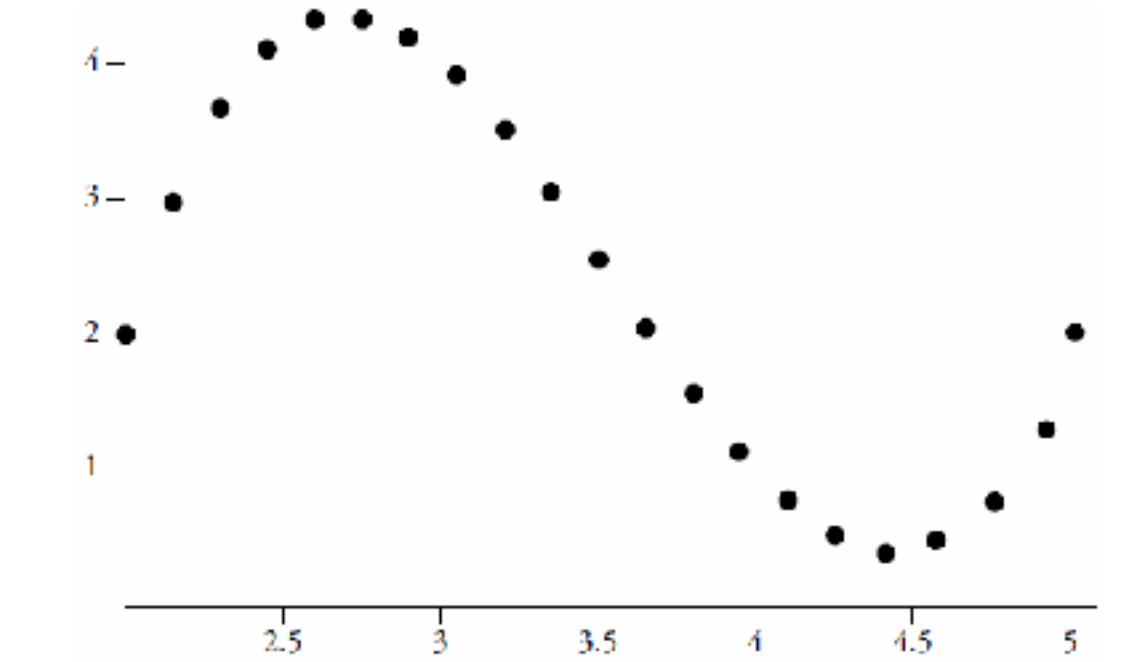


# Movimiento a través de una curva



$$P(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3$$

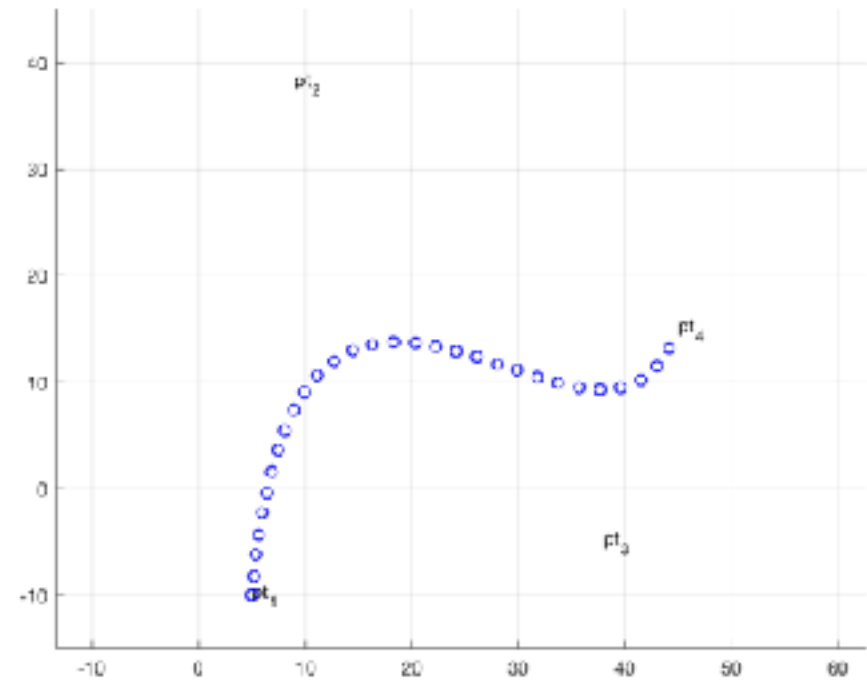
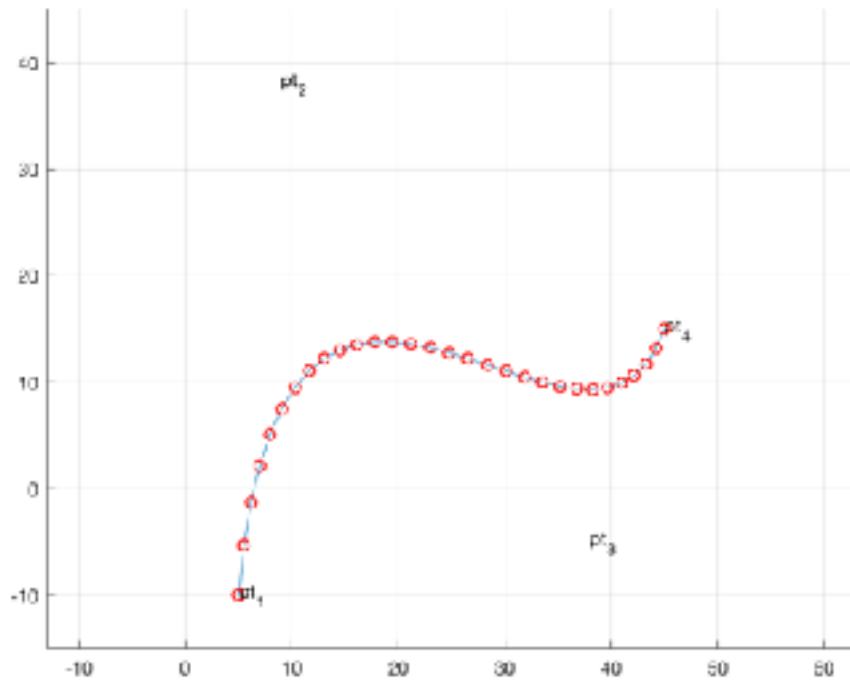
# Movimiento a través de una curva



$$P(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3$$

$$P(s) = ??$$

# Movimiento a través de una curva





# Movimiento a través de una curva

# Movimiento a través de una curva

Hay 2 métodos

1. Calcular analíticamente la longitud de la curva
2. Precomputar una tabla de valores

# Movimiento a través de una curva

## Aproximación analítica.

La longitud de la curva  $s$ , podría ser calculada de un punto  $u_1$  a un punto  $u_2$  como la siguiente integral:

$$P(u) = au^3 + bu^2 + cu + d$$

$$s = \int_{u_1}^{u_2} |dP/du| du$$

La mayoría de veces  
no tiene solución

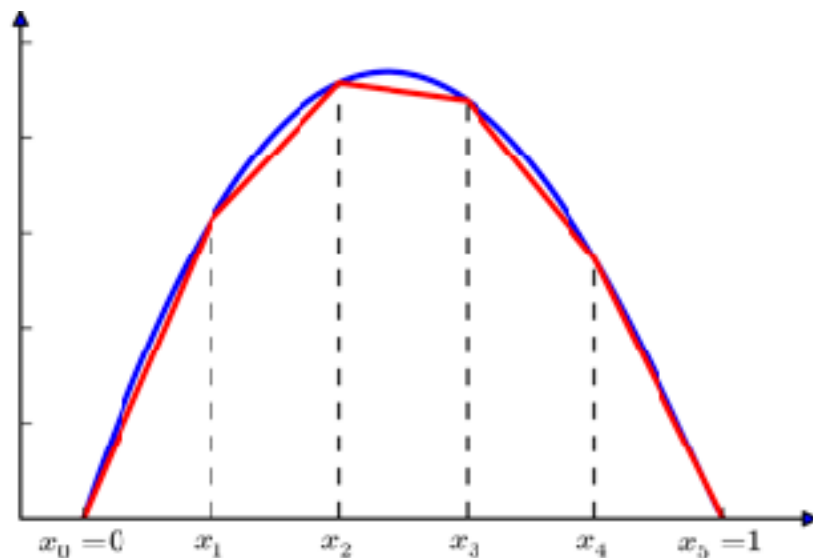
# Movimiento a través de una curva

## Estimar la longitud de arco por Forward Differencing

La manera más sencilla de establecer la correspondencia entre el parámetro y la longitud del arco es obtener desde la curva un gran número de valores paramétricos. Cada uno de ellos representa un punto en la curva. Estos puntos pueden usarse para estimar la longitud del arco.

Si se buscan puntos suficientemente próximos, se puede crear una tabla como la siguiente:

# Movimiento a través de una curva



Index	Parametric Entry	Arc Length ( $\zeta_i$ )
0	0.00	0.000
1	0.05	0.080
2	0.10	0.150
3	0.15	0.230
4	0.20	0.320
5	0.25	0.400
6	0.30	0.500
7	0.35	0.600
8	0.40	0.720
9	0.45	0.800
10	0.50	0.860
11	0.55	0.900
12	0.60	0.920
13	0.65	0.932
14	0.70	0.944
15	0.75	0.959
16	0.80	0.972
17	0.85	0.984
18	0.90	0.994
19	0.95	0.998
20	1.00	1.000

# Movimiento a través de una curva

## Aproximación adaptativa

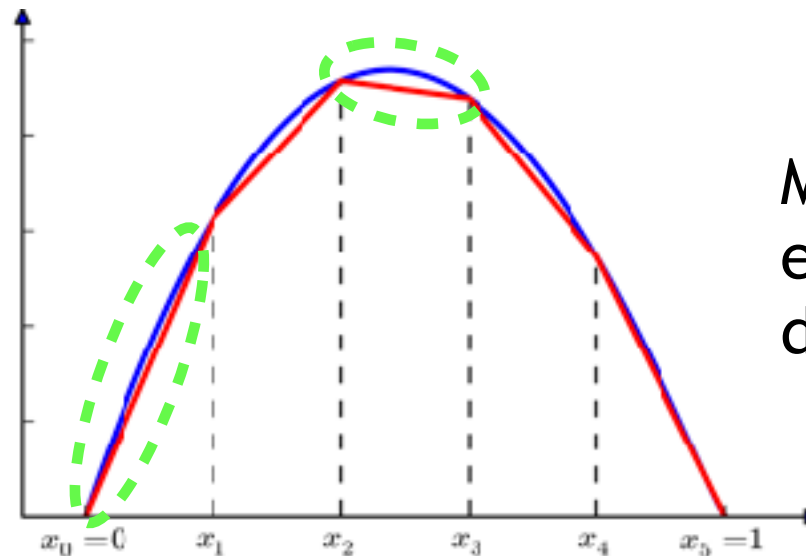
Para disminuir el error total, se plantea dedicar más computación en las zonas donde estimamos que puede haber mayor error.

Para ello se crea una tabla de manera similar a lo indicado en el caso anterior.

Se estudia la tabla así generada, de manera que se compare la longitud del arco, con la distancia lineal entre los dos puntos d la curva.

# Movimiento a través de una curva

Cuando la relación es muy grande, se entiende que la curva es muy pronunciada, y el ajuste puede cometer un mayor error, con lo que este segmento debe ser partido en trozos menores, hasta que no se supera un determinado umbral.



Misma distancia  
en  $x$ , pero muy  
diferente en  $s$

# Movimiento a través de una curva

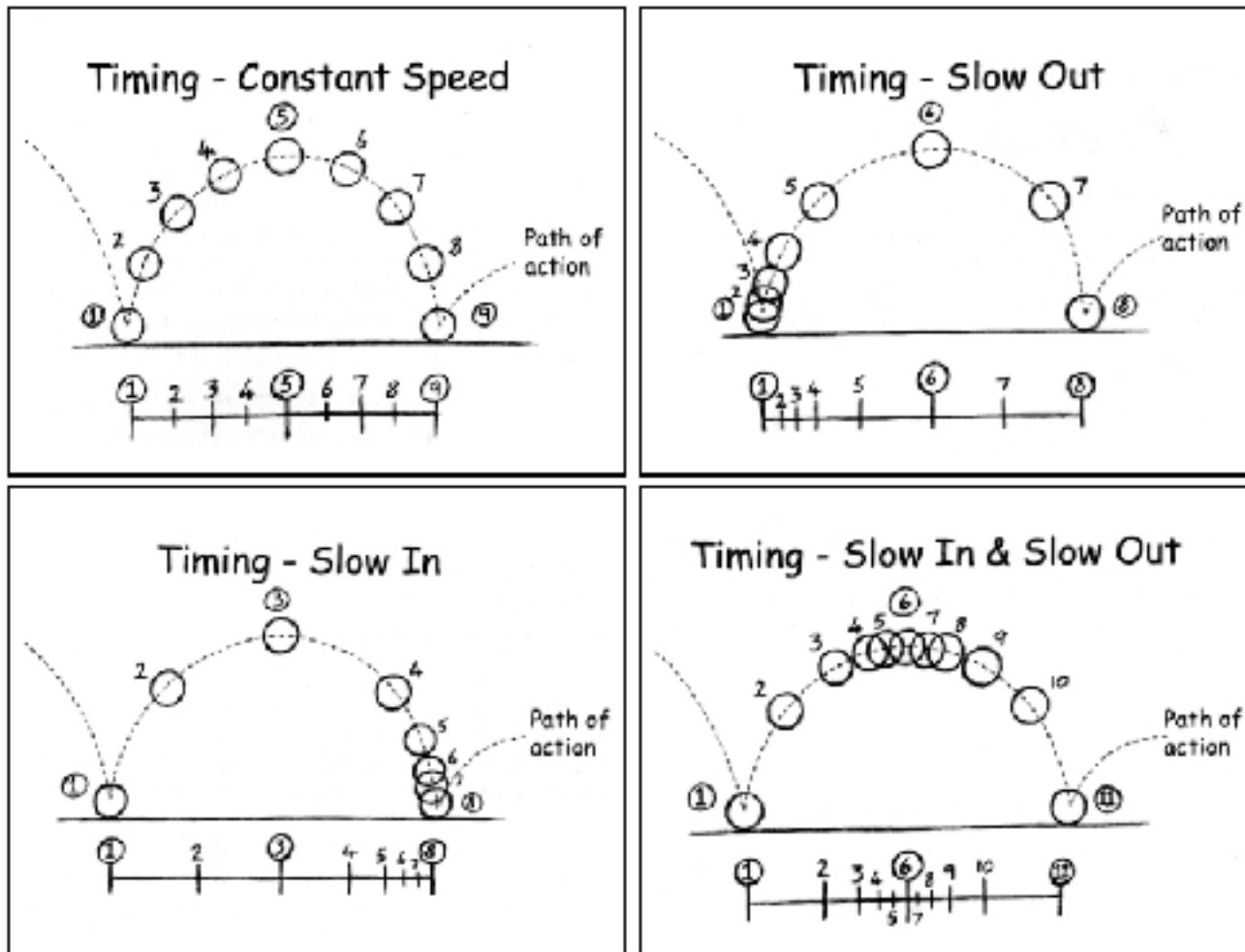
## Control de velocidad

Una vez hemos parametrizado una curva por la longitud de arco, según alguno de los métodos anteriores, es posible controlar la velocidad a la cual nos movemos por la curva.

Si avanzamos en saltos iguales de longitud de arco, entonces avanzamos a velocidad constante.



# Control de velocidad



# Movimiento a través de una curva

En esta relación entre  $t$  y  $s$ , la forma de la curva ya es irrelevante. La space curve podría ser lineal, pero la longitud del arco ser controlada por una función cubica respecto al tiempo  $s(t)$ .

Por tanto, como resumen podemos decir que la space curve dice Donde, mientras que  $s(t)$  dice Cuando.

Un valor  $t$  define un arco recorrido, y esta longitud de arco se relaciona mediante cualquiera de los métodos anteriores con una posición en la space curve.

# Movimiento a través de una curva

## Ease in/ease out

Es una de las maneras más habituales de controlar el movimiento a lo largo de una curva.

Para ello se asume que el movimiento empieza y acaba en reposo, y que no hay saltos instantáneos en la velocidad, por tanto existe continuidad de la función en la primera derivada.

Puede o no haber un intervalo de velocidad constante.

# Movimiento a través de una curva

Una manera sencilla seria la interpolación por la función seno.

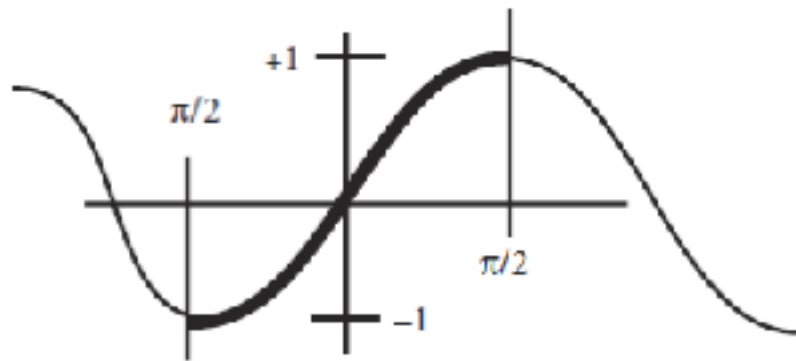
En este caso se normaliza la función  $s(t)$  entre 0 y 1, y se usa el fragmento de la función seno entre -90 y +90. La función  $s(t)$  obtenida es:

$$s(t) = ease(t) = \frac{\sin\left(t \cdot \pi - \frac{\pi}{2}\right) + 1}{2}$$

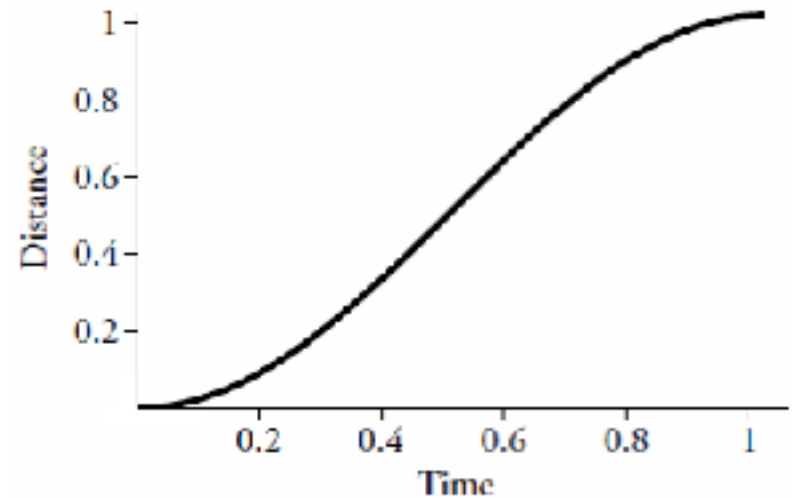
<http://gizma.com/easing/#expo3>

# Movimiento a través de una curva

Ambas funciones tienen la siguiente forma:



Sine curve segment to use as ease-in/ease-out control



Sine curve segment mapped to useful values

- **MATLAB DEMO**