

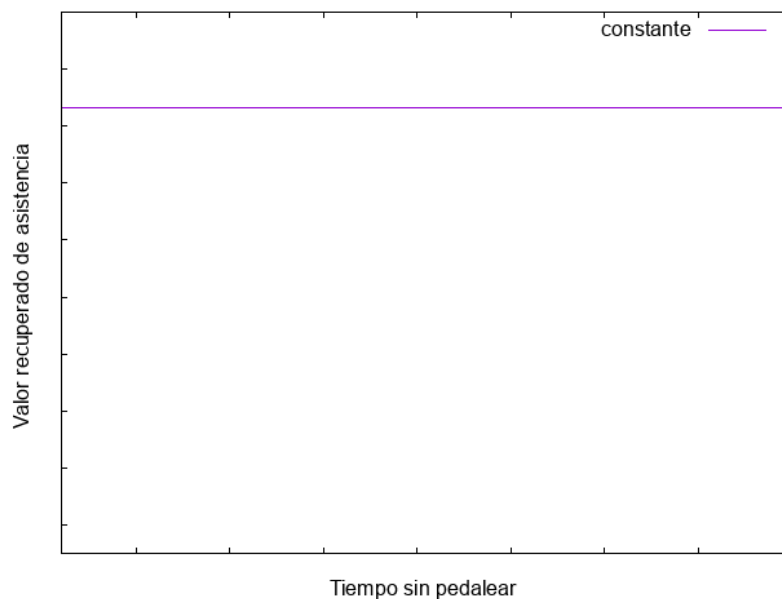
Autoprogresivos en pedaleo asistido

Introducción

- A veces dejamos de pedalear sin intención de reducir la velocidad o parar. Situaciones de ese tipo se producen de forma frecuente: para amortiguar un bache con las piernas, cambiar de posición en el sillín, adoptar una postura aerodinámica...
- En el grupo de desarrollo para legalización de la Fiido, se le ha dado el nombre de autoprogresivos al mecanismo mediante el cual el nivel de asistencia no parte desde el origen cuando se deja de pedalear durante un tiempo menor al marcado (por defecto 10 s).

Autoprogresivos constantes

- En su forma original, el valor recuperado de asistencia al volver a pedalear es siempre el mismo: el 83.3% del valor que tenía el nivel de asistencia en el momento que se deja de pedalear.



- Esto se implementa en el código del modo siguiente:

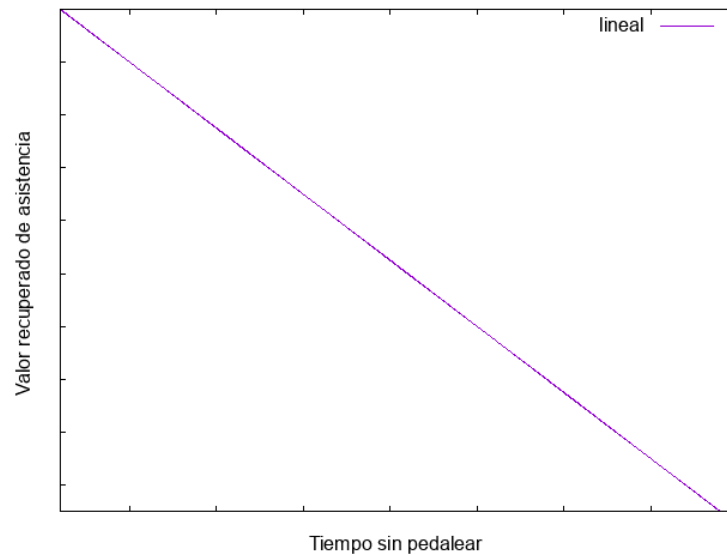
```
if (pulsos >= cadencia) {  
    if (contador_retardo_inicio_progresivo < retardo_inicio_progresivo && auto_progresivo){  
        contador_retardo_aceleracion = bkp_contador_retardo_aceleracion;  
        ...  
    }
```

donde el valor de `bkp_contador_retardo_aceleracion` se establece al dejar de pedalear como

```
bkp_contador_retardo_aceleracion = contador_retardo_aceleracion/1.2;
```

Autoprogresivos proporcionales

- A diferencia de los anteriores autoprogresivos, donde el valor recuperado de asistencia es siempre el mismo, se trata de que este sea mayor cuanto menor sea el tiempo que se haya dejado de pedalear



```
if (pulsos >= cadencia) {
  if (contador_retardo_inicio_progresivo < retardo_inicio_progresivo && auto_progresivo)
    if (bkp_contador_retardo_aceleracion > retardo_aceleracion){
      bkp_contador_retardo_aceleracion = retardo_aceleracion;
    }
    contador_retardo_aceleracion = bkp_contador_retardo_aceleracion*
      (fac_a+fac_b*contador_retardo_inicio_progresivo);
  ...
}
```

- Donde los factores a y b , para los valores por defecto,

```
- tiempo_cadencia = 250
- retardo_aceleracion = 5
- retardo_inicio_progresivo = 10
```

que implican que,

```
- contador_retardo_aceleracion varíe entre 1 y 20
- contador_retardo_inicio_progresivo varíe entre 1 y 40
```

se determinarían de la siguiente forma:

$$\begin{cases} 1 = a + b \times 1 \\ 1/20 = a + b \times 40 \end{cases}$$

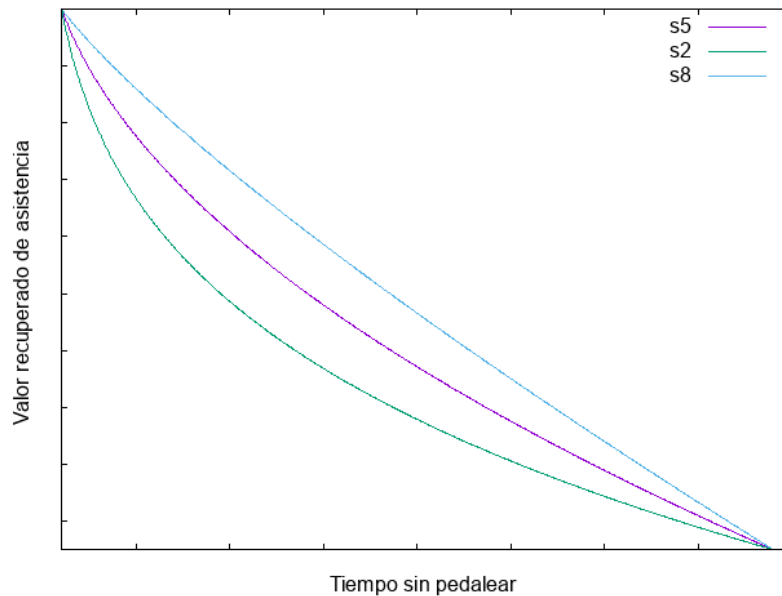
$$b = \frac{1/20 - 1}{39} \quad a = 1 - b$$

- Lo anterior, ya generalizado, corresponde en el código a las líneas del `setup()`

```
if (retardo_inicio_progresivo > 0){
    fac_b = (1.0/retardo_aceleracion-1.0)/(retardo_inicio_progresivo-1.0);
    fac_a = 1.0-fac_b;
}
```

Autoprogresivos proporcionales flexibles

- La caída lineal de la asistencia en función del tiempo cuando se deja de pedalear es solo un caso particular que puede no resultar satisfactorio para todos los usuarios.
- Así que mejor utilizar un factor más c , que el usuario pueda modificar según sus gustos, para tener diferentes tipos de relaciones entre el valor de asistencia recuperada y el tiempo sin pedalear.



```
if (pulsos >= cadencia) {
    if (contador_retardo_inicio_progresivo < retardo_inicio_progresivo && auto_progresivo){
        if (bkp_contador_retardo_aceleracion > retardo_aceleracion){
            bkp_contador_retardo_aceleracion = retardo_aceleracion;
        }
        contador_retardo_aceleracion = bkp_contador_retardo_aceleracion*
            (fac_a+fac_b*pow(contador_retardo_inicio_progresivo,fac_c));
        ...
    }
```

- Donde los factores a y b , para los valores por defecto, con $c = 0.5$, se determinarían de la siguiente forma:

$$\begin{cases} 1 = a + b \times 1^{0.5} \\ 1/20 = a + b \times 40^{0.5} \end{cases}$$

$$b = \frac{1/20 - 1}{39^{0.5} - 1^{0.5}} \quad a = 1 - b \times 1^{0.5}$$

- Lo anterior, ya generalizado, corresponde en el código a las líneas del `setup()`

```
if (retardo_inicio_progresivo > 0){
    fac_b = (1.0/retardo_aceleracion-1.0)/(pow((retardo_inicio_progresivo-1.0),fac_c)-pow(1.0,fac_c));
    fac_a = 1.0-pow(1.0,fac_c)*fac_b;
}
```

- En cuanto al factor c , este se establece en la cabecera a través de la línea:

```
float fac_c = suavidad_autoprogresivos/10.0;
```

donde `suavidad_autoprogresivos` es un parámetro que define el usuario entre 1 y 10 para establecer diferentes respuestas, cuanto más alto más bruscos son los autoprogresivos.

Autoprogresivos proporcionales flexibles corregidos

- Unos autoprogresivos, con la respuesta deseada en los niveles de asistencia altos, resultarán demasiado bruscos en los niveles de asistencia bajos.
- Para suavizar el comportamiento de los autoprogresivos en los niveles bajos de asistencia, el código donde se recupera el valor de asistencia se dejó finalmente en la forma:

```
if (pulsos >= cadencia) {
    if (contador_retardo_inicio_progresivo < retardo_inicio_progresivo && auto_progresivo){
        if (bkp_contador_retardo_aceleracion > retardo_aceleracion){
            bkp_contador_retardo_aceleracion = retardo_aceleracion;
        }
        contador_retardo_aceleracion = bkp_contador_retardo_aceleracion*
            (fac_a+fac_b*pow(contador_retardo_inicio_progresivo,fac_c))*v_crucero/voltaje_maximo;
        ...
    }
```