



**UNIVERSIDAD
DE GRANADA**

Facultad de Ciencias

GRADO EN FÍSICA

Pedestrians synchronization and the wobbling of the Millenium Bridge.

Resumen

La inauguración del puente del Milenio en Londres trajo consigo la creación de distintos modelos que se utilizaron para explicar el fenómeno de 'Phase locking' que se observaba: la sincronización de la oscilación del puente con los peatones que ocurría al tener un número de peatones por encima de un número crítico. Hemos estudiado los modelos de Strogatz y Belykh ya que cada uno de ellos propone un enfoque diferente de la construcción del sistema. En ambos modelos hemos podido llegar a conclusiones muy parecidas obteniendo un número crítico de peatones en torno a $N_c \in [150; 170]$.

Autores

José Luis Pérez Barbero
Ismael Quirantes Bamohamed
Marta Guzmán Ferrer

Índice

A. Introducción	3
B. Modelo de Strogatz: Aplicación del modelo de Kuramoto	3
B.1. Descripción del modelo	3
B.2. Estudio analítico del número crítico de peatones	5
B.3. Simulación y resultados	9
C. Modelos de Belykh	11
C.1. Modelo del péndulo invertido	12
C.2. Modelo de van der-Pol	13
C.3. Estudio analítico del número crítico de peatones	13
C.4. Simulación y resultados	16
C.4.1. Modelo de van der-Pol para péndulos idénticos	16
C.4.2. Modelo de péndulos invertidos	18
D. Conclusiones	20
E. ANEXO: Códigos de Python	22
E.1. Modelo de Strogatz	22
E.2. Modelo de Van der-Pol (Peatones idénticos)	26
E.3. Modelo de los péndulos invertidos.	28

A. Introducción

Con la apertura del puente del Milenio en Londres, aparecieron una serie de fenómenos que abrieron la puerta a un estudio más profundo de las características de un puente de esta índole. Se dieron cuenta de que a partir de un número crítico de peatones, este experimentaba un movimiento para nada despreciable de bamboleo (oscilación lateral) peligroso tanto para los peatones como para la estructura del puente. Por ello, hicieron un modelo que permitiera calcular este número crítico de personas basado en el modelo de Kuramoto. Aunque posteriormente se hicieron modelos más sofisticados en los que se tuvieron en cuenta más parámetros biológicos como la influencia de la altura, edad o peso de la persona.

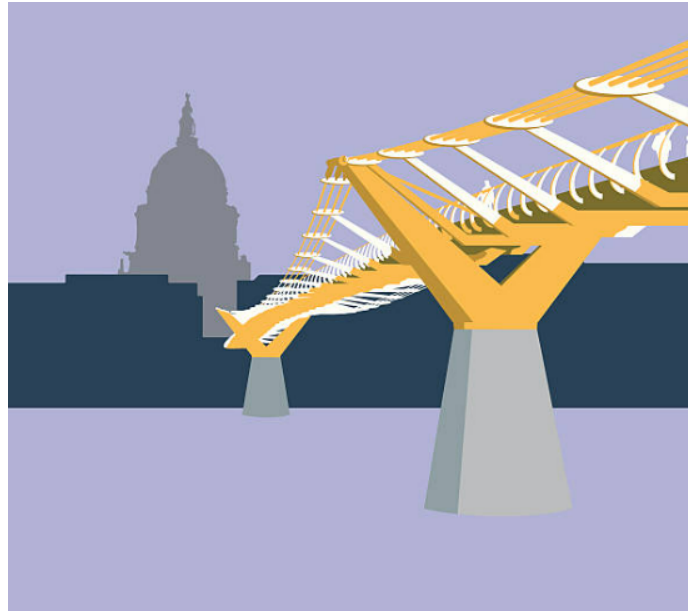


Figura 1: Ilustración Puente del Milenio

Lo que se presenta en este trabajo es replicar el modelo de Kuramoto propuesto por [4] resolviendo una serie de ecuaciones descriptivas de nuestro sistema, para luego calcular el número crítico de personas para el caso del puente del Milenio. Luego replicaremos el trabajo más sofisticado de [2], y se hará una comparación.

B. Modelo de Strogatz: Aplicación del modelo de Kuramoto

B.1. Descripción del modelo

El puente del milenio es modelado como un oscilador armónico débilmente amortiguado y forzado en este modelo tal y como muestra la siguiente ecuación:

$$M \frac{d^2 X}{dt^2} + B \frac{dX}{dt} + KX = G \sum_{i=1}^N \sin \theta_i \quad (1)$$

donde $X(t)$ es el desplazamiento lateral del puente, M su masa, B su amortiguación y K su constante de recuperación. Si consideramos N peatones sobre el puente, cada uno va a ejercer una fuerza lateral máxima G modulada por el seno de la fase de cada peatón. El ritmo del caminante se ve determinado por la función $\theta(t)$, que queda descrita por una ecuación que acopla con la anterior, obteniendo así un sistema descrito matemáticamente por el acoplamiento de $N+1$ ecuaciones diferenciales. La evolución individual de un peatón vendrá dada por la ecuación diferencial (2), que acopla al peatón i con la oscilación del puente:

$$\frac{d\theta_i}{dt} = \Omega_i + CA \sin(\Psi - \theta_i + \alpha) \quad (2)$$

Esta expresión proviene del modelo de Kuramoto [1], ecuación que describe la dinámica de un sistema considerando el acoplamiento de los diferentes componentes que lo conforman. En (2), Ω_i son frecuencias generadas y distribuidas normalmente, que representan las frecuencias naturales de los peatones al caminar, C es un parámetro que tiene en cuenta el grado de acoplamiento de la oscilación del puente con el peatón, donde A sería la amplitud de esta oscilación. Ψ es la fase promedio de las vibraciones tal que $X = A \sin(\Psi)$ y $\frac{\partial X}{\partial t} = \Omega_0 A \cos(\Psi)$, siendo $\Omega_0 = \sqrt{\frac{K}{M}}$ la frecuencia de resonancia del puente y α el desfase introducido entre el puente y los peatones.

La amplitud de las oscilaciones vendrá dada para un tiempo determinado por:

$$A = (X^2 + (\Omega_0^{-1} \partial X / \partial t)^2)^{1/2} \quad (3)$$

Y el parámetro de orden de Kuramoto por:

$$R = N^{-1} \left| \sum_{j=1}^N \exp(i\theta_j) \right| \quad (4)$$

La ecuación [2] está basada en el modelo de Kuramoto, modelo usado sobretodo para describir la sincronización de un gran conjunto de osciladores (químicos, biológicos...) acoplados y que permite describir el fenómeno de *phase locking*: un fenómeno observable en el que las partes del sistema optan de forma espontánea, por adquirir una determinada fase.

El modelado de peatones consiste en dotar de una fase θ_i a los peatones, que irá

evolucionando en el tiempo conforme a la ecuación (2). Esta fase es la encargada de modelar la fuerza lateral ejercida sobre el puente por el hecho de que existan peatones caminando sobre él. Inspeccionando la Figura 2, podemos interpretar que cuando la fase del peatón es $\pi/2$ el peatón está ejerciendo una fuerza máxima en una dirección (está apoyando uno de los pies completamente). Esta fase irá evolucionando hasta $3\pi/2$ donde ejercerá entonces esta fuerza máxima en el sentido contrario. Los ángulos intermedios a estos representarían los procesos de terminar de apoyar y levantar el pie. Las fases 0 y π representarían un apoyo simultáneo del peatón con los dos pies, cancelando cualquier tipo de contribución a la fuerza lateral ejercida sobre el puente.

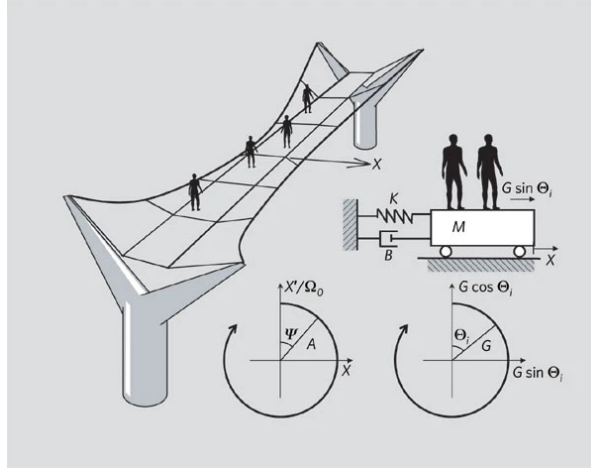


Figura 2: Modelado del puente y de los peatones. El puente es considerado como un oscilador armónico amortiguado y forzado. A los peatones se les asocia una fase que simula las pisadas izquierda y derecha. Figura de [4]

Como resumen, el sistema de ecuaciones diferenciales que describen la dinámica del sistema es:

$$\begin{cases} M \frac{d^2 X}{dt^2} + B \frac{dX}{dt} + KX = G \sum_{i=1}^N \sin \theta_i \\ \frac{d\theta_i}{dt} = \Omega_i + CA \sin(\Psi - \theta_i + \alpha) \end{cases} \quad (5)$$

Que deberá resolverse con diferentes métodos numéricos.

B.2. Estudio analítico del número crítico de peatones

En esta sección se procede con un desarrollo analítico del número crítico de peatones, el valor para el cual se produce la sincronización. De forma general, el desarrollo consiste en simplificar las ecuaciones diferenciales para realizar un estudio de forma cuantitativa, pareciéndose al **modelo de Kuramoto**, por lo que se puede realizar un estudio basándose en dicho modelo, siguiendo el desarrollo de Strogatz [4].

En primer lugar, se adimensionalizan las ecuaciones (5) a través de los siguientes cambios:

$$\begin{aligned}\Omega_0 &= \sqrt{\frac{K}{M}} & 2\varsigma &= B\frac{\Omega_0}{K} \\ L_1 &= N\frac{G}{K} & L_2 &= \frac{\Omega_0}{C} \\ L &= \sqrt{L_1 L_2} & \varepsilon &= \sqrt{\frac{L_1}{L_2}} = \sqrt{\frac{NGC}{K\Omega_0}}\end{aligned}$$

e introducimos las siguientes variables:

$$\begin{aligned}\tau &= \Omega_0 t \\ x &= X/L \\ a &= A/L\end{aligned}$$

Obteniéndose las ecuaciones diferenciales (6) y (7):

$$\frac{d^2 x}{d\tau^2} + 2\varsigma \frac{dx}{d\tau} + x = \varepsilon \langle \sin \Theta_i \rangle \quad (6)$$

$$\frac{d\Theta_i}{d\tau} = \frac{\Omega_i}{\Omega_0} + \varepsilon a \sin(\Psi - \Theta_i + \alpha) \quad i = 1, \dots, N. \quad (7)$$

y acabando con los últimos cambios

$$\begin{aligned}\varsigma &= \varepsilon b \\ \frac{\Omega_i}{\Omega_0} &= 1 + \varepsilon \omega_i \\ \theta_i &= \Theta_i - \tau \\ \psi &= \Psi - \tau\end{aligned}$$

obtenemos de (6) y (7) las ecuaciones (8) y (9).

$$\frac{d^2 x}{d\tau^2} + x = \varepsilon [\langle \sin \tau + \theta_i \rangle - 2b \frac{dx}{d\tau}] \quad (8)$$

$$\frac{d\theta_i}{d\tau} = \varepsilon [\omega_i + a \sin(\psi - \theta_i + \alpha)] \quad (9)$$

La ecuación diferencial (8) es de segundo orden, por lo que para simplificar el desarrollo, será transformada mediante cambios de variables, siguiendo los pasos de Guckenheimer [3]. El primer cambio que se realiza será girar un ángulo τ los ejes del

espacio de fases obteniendo las variables u y v a través de la matriz de transformación A ,

$$\begin{pmatrix} u \\ v \end{pmatrix} = A \begin{pmatrix} x \\ \frac{dx}{d\tau} \end{pmatrix}, \quad A = \begin{pmatrix} \cos(\tau) & -\sin(\tau) \\ -\sin(\tau) & -\cos(\tau) \end{pmatrix}$$

Posteriormente se cambia a coordenadas polares a y ψ .

Averaging Method

Así, una vez pasado a dos ecuaciones diferenciales de primer orden en coordenadas polares, se puede aplicar el llamado *Averaging Method* que es la aplicación del **Teorema del Promedio** que, como se explica en [3], nos indica que si tenemos una ecuación diferencial de la siguiente manera

$$\dot{x} = \varepsilon f(x, \dot{x}, \varepsilon) \quad (10)$$

y otra de la forma

$$\dot{y} = \frac{\varepsilon}{T} \int_0^T f(y, t, 0) dt \doteq \varepsilon \bar{f}(y), \quad (11)$$

donde f es una función periódica de periodo T , existe un cambio de variable $x = y + \varepsilon w(y, t, \varepsilon)$ tal que la ecuación (10) se transforma en la ecuación (12)

$$\dot{y} = \varepsilon \bar{f}(y) + \varepsilon^2 f_1(y, t, \varepsilon) \quad (12)$$

Por lo tanto, para $0 < \varepsilon \ll 1$ se puede aproximar la ecuación (10) a (11) para un tiempo $t \sim \frac{1}{\varepsilon}$ en orden a cumplir $|x(t) - y(t)| = \mathcal{O}(\varepsilon)$, es decir, las variables de las amplitudes y de las fases deben variar lentamente.

Al aplicar el método anterior, se promedia los términos sinusoidales, llegando a anularse algunos, y obtenemos las ecuaciones (13), (14) y (15) en una escala de $\tau \sim \frac{1}{\varepsilon}$.

$$\dot{a} = -ba - \frac{1}{2} \langle \sin(\psi - \theta_i) \rangle \quad (13)$$

$$a\dot{\psi} = -\frac{1}{2} \langle \cos(\psi - \theta_i) \rangle \quad (14)$$

$$\dot{\theta}_i = \omega_i + a \sin(\psi - \theta_i + \alpha) \quad i = 1, \dots, N, \quad (15)$$

A la vista de las ecuaciones diferenciales, cabe destacar que se presupone que $\varepsilon \ll 1$, que la derivación está respecto a $T = \varepsilon\tau$ y que estas ecuaciones promediadas describen como evolucionan las vibraciones del puente y la sincronización de la multitud, de hecho, se puede esperar que las magnitudes mencionadas tenderán a variar de

forma lenta, de tal manera que para notar una evolución significativa necesitaríamos decenas e incluso centenares de peatones.

Soluciones Estacionarias

Las ecuaciones (13)-(15) tienen un gran parecido al modelo de Kuramoto por lo que es adecuado tratar a éstas de forma similar. Teniendo en cuenta lo anterior, la simetría rotacional que poseen las ecuaciones y la suposición de que $\dot{\psi} = 0$ haciéndose constante ψ (en concreto le daremos el valor $\psi = \frac{\pi}{2}$), al realizar un estudio de las soluciones estacionarias aparecen (16), (17) y (18)

$$ba = \frac{1}{2} \langle \cos \theta_i \rangle \quad (16)$$

$$\langle \sin \theta_i \rangle = 0 \quad (17)$$

$$\sin \theta_i^* = \omega_i / a \quad i = 1, \dots, N. \quad (18)$$

La ecuación (18) se cumple para $|\omega_i| < a$, estas frecuencias ω se distribuyen con una función de densidad $p(\omega)$ y para el resto de casos se distribuyen según (19)

$$\rho(\theta, \omega) = \frac{1}{2\pi} \frac{\sqrt{\omega^2 - a^2}}{|\omega - a \sin \theta|}. \quad (19)$$

Así, una vez que sabemos la distribución, podemos desarrollar un poco más la ecuación (16) escribiéndolo como una suma de contribuciones de los peatones desincronizados con los que si que lo están, dando como resultado que la parte desincronizada ($|\omega| > a$) es nula y por lo tanto teniendo en cuenta (18) tenemos (20)

$$2ba = \int_{-\pi/2}^{\pi/2} \cos \theta p(a \sin \theta) a \cos \theta d\theta \quad (20)$$

y realizando el límite de $a \rightarrow 0^+$, calculando la integral y deshaciendo los cambios hechos para adimensionalizar las ecuaciones diferenciales originales, obtenemos el **Número de Peatones Crítico** (21).

$$N_c = \frac{4\zeta}{\pi} \left[\frac{K}{GCP(\Omega_0)} \right] \quad (21)$$

Podemos notar varios aspectos de la fórmula: en primer lugar, depende inversamente de la probabilidad de que un peatón camine oscilando a la frecuencia natural del puente Ω_0 por lo que es importante tener en cuenta la varianza de la distribución ya que si la varianza es menor la probabilidad de dicha frecuencia es mayor y el número de peatones crítico mayor, luego la sincronización del sistema se produce antes, y

al contrario, es decir, si la varianza aumenta, disminuye la probabilidad (el número crece y la sincronización ocurre más tarde). En segundo lugar, se puede observar que depende de algunos parámetros del puente tal como la amortiguación y la constante de recuperación, por lo que esta fórmula podría ser útil para el diseño de puentes en vista a la seguridad de los peatones.

B.3. Simulación y resultados

El primer paso es caracterizar el comportamiento del puente, es decir, fijar los parámetros que determinan su masa, su constante elástica, su coeficiente de amortiguación y su frecuencia de resonancia. En la siguiente tabla se introducen los valores de estas constantes en el orden mencionado:

$\mathbf{M} \text{ (kg)}$	$\mathbf{K} \text{ (kg} \cdot \text{s}^{-2}\text{)}$	$\mathbf{B} \text{ (kg} \cdot \text{s}^{-1}\text{)}$	$\mathbf{\Omega_0 \text{ (rad} \cdot \text{s}^{-1}\text{)}}$
$1.13 \cdot 10^5$	$4.73 \cdot 10^6$	$1.10 \cdot 10^4$	6.47

Cuadro 1: Parámetros que caracterizan al puente: masa, constante recuperadora, constante de amortiguación y frecuencia de resonancia.

En segundo lugar, caracterizamos el comportamiento de los peatones:

1. *Acoplamiento peatón-puente*: El grado de este acoplamiento viene dado por la constante C . Para la simulación se le ha establecido un valor de $C = 10m^{-1}s^{-1}$
2. *Desfase*: Se introduce un desfase que actúa como retraso en la interacción peatón-puente y al que se le ha atribuido un valor $\alpha = \pi/2$
3. *Inicialización de peatones*: A las fases θ_i se les ha asignado un valor aleatorio al principio del algoritmo (entre 0 y 2π).
4. *Frecuencias de cada peatón*: Se han fijado las frecuencias de los peatones siguiendo una distribución normal con media $\Omega_0 = 6,47rad \cdot s^{-1}$ y desviación típica $0.63rad \cdot s^{-1}$.

Una vez fijados los parámetros del modelo, podemos llevar a cabo la simulación. Para ello, se ha utilizado el lenguaje de programación *python* y diferentes librerías como *scipy* para resolver el sistema (5).

La simulación se realizó de la siguiente manera: el puente se encuentra vacío los primeros 250s, tiempo en el que se introducen a continuación 50 personas. Se deja evolucionar el sistema durante otros 250s y a partir de entonces se van introduciendo 10 personas cada 100s. Durante toda la evolución del sistema se van registrando los valores de la amplitud de oscilación del puente y el del parámetro de orden de Kuramoto. Una vez alcanzado el número crítico de peatones, estas dos últimas métricas

del puente aumentan de forma abrupta con el número de personas: se ha alcanzado el estado de *phase lock*. Lo descrito en este párrafo se muestra en la siguiente figura:

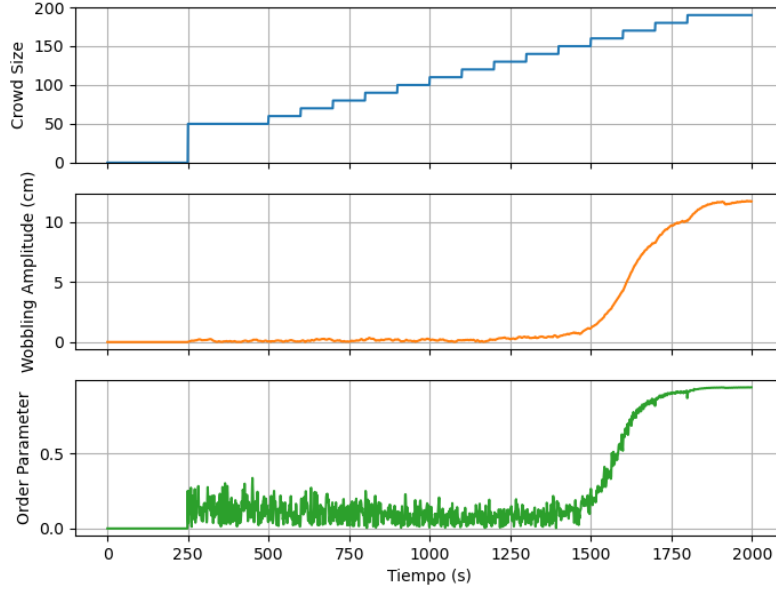


Figura 3: Representación del número de personas consideradas, amplitud de las oscilaciones y parámetro de orden en función del tiempo. A partir de aproximadamente 150 personas, la amplitud y el parámetro de orden aumentan de forma abrupta con el número de personas.

El phase lock puede visualizarse también en los registros del algoritmo. Si representamos el seno de las fases de los peatones en conjunto y en función del tiempo, puede diferenciarse claramente entre la fase coordinada y descoordinada, tal y como se muestra en las siguientes figuras:

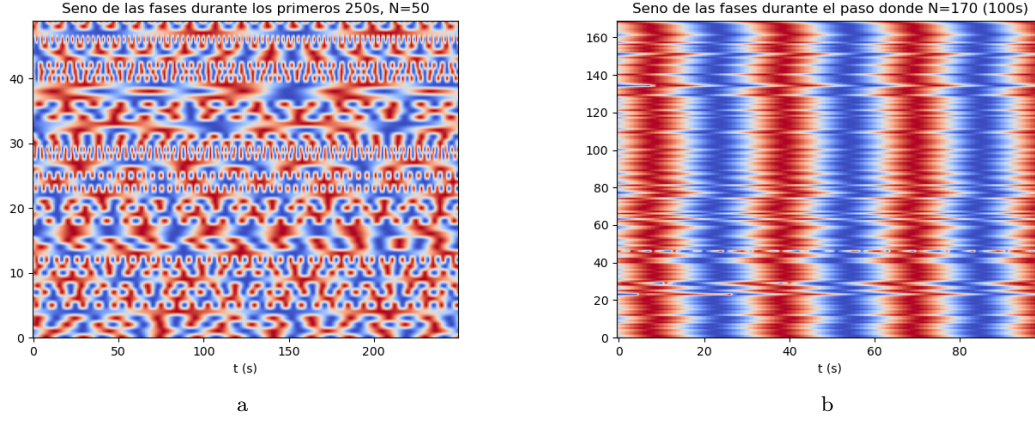


Figura 4: Mapas de calor del seno de las fases de los peatones en función del tiempo. (a): Región en la que encontramos a un número de peatones por debajo del número crítico (no hay sincronización por lo que el patrón de la figura es completamente aleatorio). (b): Región donde se ha sobrepasado el número crítico, los peatones se han coordinado (phase locking, el patrón de la figura refleja una clara coordinación entre los peatones en el tiempo).

C. Modelos de Belykh

Otros modelos descriptivos que se implementaron fueron los modelos de Belykh, ambos modelos parten de una ecuación general pero con distinta función de respuesta del peatón ante la oscilación. Estos modelos simulan el paso del peatón como una oscilación lateral.

$$\begin{cases} \ddot{x}_i + f(x_i, \dot{x}_i) = -\ddot{y} \\ \ddot{y} + 2h\dot{y} + \Omega_0^2 y = -r \sum_{i=1}^n \ddot{x}_i \end{cases} \quad (22)$$

La variable x hace referencia a la ecuación del movimiento del peatón mientras la variable y sería la ecuación del movimiento de oscilación del puente. La función $f(x_i, \dot{x}_i)$ es la función que representa el mecanismo de oscilación y la respuesta de cada peatón ante la vibración del puente. Considerando a cada peatón con una masa m , sin pérdida de generalidad porque consideraremos que cada peatón tiene una frecuencia de paso diferente, el parámetro $h = d/(2(M + nm))$ es el coeficiente de amortiguación normalizado y sin dimensiones, $\Omega_0 = \sqrt{K/(M + nm)}$ es la frecuencia natural del puente actualizada para los n peatones. Por último $r = m/(M + nm)$ es la fuerza del acoplamiento entre los peatones y el puente.

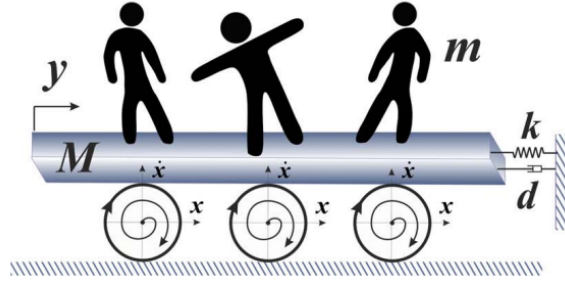


Figura 5: La plataforma de masa M representa la oscilación del puente con una fuerza recuperadora y amortiguada. Los peatones se modelan mediante osciladores auto-sostenidos, que representan el equilibrio lateral del caminante y son capaces de ajustar la posición de sus centros de masa. Figura de [2]

C.1. Modelo del péndulo invertido

Muchos estudios en biomecánica [2] han confirmado que un modelo de péndulo invertido puede aplicarse con éxito al análisis del equilibrio del cuerpo humano en su dirección lateral del paso.

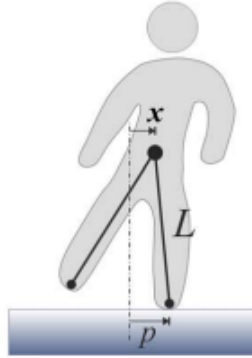


Figura 6: La variable x representa el desplazamiento con respecto al centro de masas y la variable p la posición lateral del centro de presión del pie, L es la longitud del péndulo invertido. Figura de [2]

La ecuación que describe nuestro modelo es (22), y para este caso:

$$f(x_i, \dot{x}_i) = \lambda (\dot{x}_i^2 - v^2 x_i^2 + v^2 a^2) \dot{x}_i - \omega_0^2 x_i \quad (23)$$

Con λ el parámetro de amortiguación, v el periodo del ciclo límite y a la amplitud del ciclo límite (desplazamiento lateral del paso). $\omega_0 = \sqrt{g/L}$.

C.2. Modelo de van der-Pol

El modelo de van der-Pol es descrito nuevamente por (22) pero en este caso:

$$f(x_i, \dot{x}_i) = \lambda_i (\dot{x}_i^2 + x_i^2 a^2) \dot{x}_i + \omega_i^2 x_i \quad (24)$$

donde λ_i es el parámetro de amortiguación de cada peaton, $\omega_i = \omega_0^i$ cada peatón tiene una frecuencia diferente y $\Omega = \Omega_0/v$.

El oscilador de van der-Pol es un oscilador con amortiguamiento no lineal y su evolución temporal obedece una ecuación diferencial de segundo orden, tal y como se mostraría el sistema (5) una vez caracterizada la f .

C.3. Estudio analítico del número crítico de peatones

Para predecir analíticamente el número crítico de peatones, [2] parte de la situación más sencilla: el modelo de Van der-Pol para osciladores (peatones) idénticos (aunque es extensible al caso de osciladores no idénticos). El primer paso es caracterizar el estado de phase lock. Esta fase correspondería a considerar las siguientes soluciones del sistema (5):

$$x_i(t) = B_i \sin(t + \varphi_i) \text{ y } y(t) = A \sin(t + \psi) \quad (25)$$

B_i y φ_i son la amplitud y la fase del oscilador i (peatón i). Como vamos a considerar peatones idénticos podemos fijar $B_i = B$ y $\varphi_i = \varphi$. Esta simplificación corresponde a una coordinación completa de los peatones tal que $x_1 = x_2 = \dots = x_n = B \sin(t + \varphi)$, la situación más favorable posible para hacer comenzar al puente sus oscilaciones. Sustituyendo (25) en la primera ecuación del sistema (22) caracterizando $f(x, \dot{x}, t)$ con (24):

$$B(\omega^2 - 1) \sin(t + \varphi) + \lambda B(B^2 - a^2) \cos(t + \varphi) = A \sin(t + \psi) \quad (26)$$

Si ahora definimos un ángulo β tal que:

$$\begin{aligned} \cos \beta &= \frac{\omega^2 - 1}{\sqrt{(\omega^2 - 1)^2 + \lambda^2 (B^2 - a^2)^2}} \\ \sin \beta &= \frac{\lambda (B^2 - a^2)}{\sqrt{(\omega^2 - 1)^2 + \lambda^2 (B^2 - a^2)^2}} \end{aligned} \quad (27)$$

Con (26) y (27) obtenemos que:

$$B\sqrt{(\omega^2 - 1)^2 + \lambda^2 (B^2 - a^2)^2} [\cos \beta \sin(t + \varphi) + \sin \beta \cos(t + \varphi)] = A \sin(t + \psi) \quad (28)$$

Si ahora tenemos en cuenta que $\cos \beta \sin(t + \varphi) + \sin \beta \cos(t + \varphi) = \sin(t + \varphi + \beta)$:

$$B\sqrt{(\omega^2 - 1)^2 + \lambda^2 (B^2 - a^2)^2} \sin(t + \varphi + \beta) = A \sin(t + \psi) \quad (29)$$

La ecuación (29) se cumplirá si:

$$A = B\sqrt{(\omega^2 - 1)^2 + \lambda^2 (B^2 - a^2)^2}, \text{ y si } \beta = \psi - \varphi \quad (30)$$

Teniendo en cuenta ahora la ecuación (27) obtenemos las siguientes ecuaciones que determinan la amplitud de las oscilaciones del puente y el desfase entre osciladores y el puente:

$$A = \sqrt{B^2 (\omega^2 - 1)^2 + \lambda^2 (B^2 - a^2)^2} B^2 \quad (31)$$

$$\tan(\varphi - \psi) = -\lambda \frac{B^2 - a^2}{\omega^2 - 1} \quad (32)$$

Llegados a este punto, nos damos cuenta de que necesitamos más ecuaciones que nos permitan obtener condiciones sobre las cuatro constantes A, B, φ y ψ . Para ello, sustituiremos las soluciones del phase lock (25) en la segunda ecuación del sistema (22):

$$A = \frac{rn}{\sqrt{\Delta}} B, \text{ con } \Delta = (\Omega^2 - 1)^2 + 4h^2 \quad (33)$$

Comparando (33) y (31) podemos llegar a que:

$$B^2 = a^2 + \frac{1}{\lambda} \sqrt{\frac{(rn)^2}{\Delta} - (\omega^2 - 1)^2} \quad (34)$$

Esta última ecuación aporta una condición clave para la existencia de B^2 . En concreto, debemos imponer que el radicando sea positivo para que todo el desarrollo anterior sea válido (asumiendo soluciones correspondientes a una fase de coordinación (phase lock)). En concreto, esta condición establece que:

$$\frac{(rn)^2}{\Delta} \geq (\omega^2 - 1)^2 \quad (35)$$

Que a su vez nos da cuenta de un número crítico n_c a partir del cual, la fase de phase lock es compatible con nuestras suposiciones. Gráficamente, podemos calcular este número crítico si representamos juntas las curvas (31) y (33) en función de B :

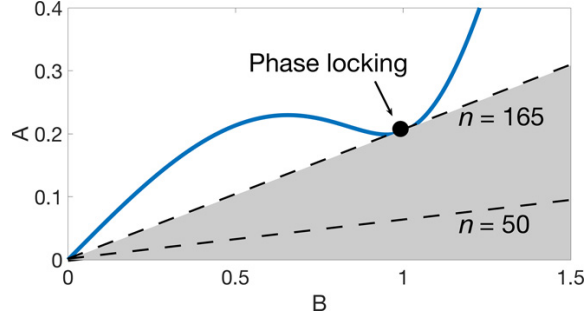


Figura 7: Cuando la curva azul (ecuación 31) es capaz de intersecar con la línea punteada (ecuación 33) encontrar estados compatibles con phase lock. Los parámetros elegidos para el modelo arrojan un valor de peatones crítico de $n_c = 165$. Figura extraída de [2].

Finalmente, cabe mencionar en esta sección la predicción analítica realizada en [2], muy similar a lo anterior pero para el caso de osciladores no idénticos. Para ello vamos a considerar que existe un rango de frecuencias $[\omega_-, \omega_+]$ en la que la frecuencia del peatón i puede estar. A continuación, llevando a cabo los mismos pasos que anteriormente llegamos a las siguientes ecuaciones:

$$A^2 = B_i^2 (\omega_i^2 - 1)^2 + \lambda^2 (B_i^2 - a^2)^2 B_i^2 \quad (36)$$

$$\tan(\varphi_i - \psi) = -\lambda \frac{B_i^2 - a^2}{\omega_i^2 - 1} \quad (37)$$

$$A^2 = \frac{(rn)^2}{\Delta} \bar{B}^2 \quad (38)$$

$$\tan(\bar{\varphi} - \psi) = \frac{2h}{1 - \Omega^2} \quad (39)$$

Donde $\bar{B} = \frac{1}{n} \sum_{i=1}^n B_i$ y $\bar{\varphi} = \frac{1}{n} \sum_{i=1}^n \varphi_i$ son la amplitud y fase media de los peatones respectivamente. Manipulando las ecuaciones (36, 37, 38) llegamos a:

$$B_i^2 (\omega_i^2 - 1)^2 + \lambda^2 (B_i^2 - a^2)^2 B_i^2 = \frac{(rn)^2}{\Delta} \bar{B}^2 \quad (40)$$

El último paso es encontrar las amplitudes B_i y \bar{B} que permiten la fase de coor-

dinación. Podemos encontrar estos valores si representamos por separado la parte izquierda y derecha de la ecuación (40). Para la parte izquierda, vamos a representar dos curvas considerando las cotas inferior y superior del intervalo $[\omega_-, \omega_+]$. La región donde se corten estas dos curvas con la que representa la parte derecha de dicha ecuación, será donde el estado de phase lock puede darse. Esto puede verse en la siguiente figura:

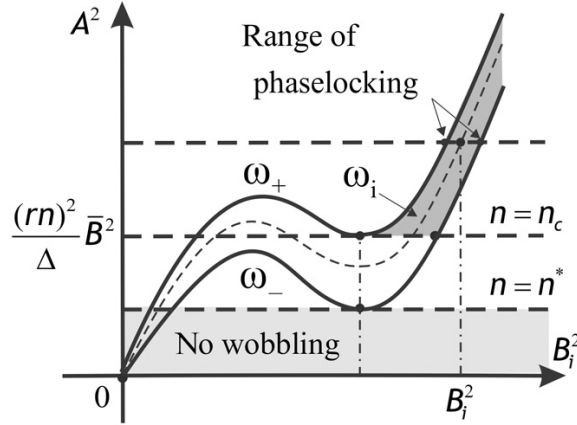


Figura 8: En la intersección de las tres curvas mencionadas es compatible un estado de phase locking. Figura extraída de [2].

Con los mismos parámetros, este nuevo análisis arroja el mismo número crítico que considerando peatones idénticos: $n_c = 165$.

C.4. Simulación y resultados

C.4.1. Modelo de van der-Pol para péndulos idénticos

En primer lugar, fijamos los parámetros del puente como hicimos con el modelo de Strogatz, que son: la frecuencia natural del puente, un parámetro de amortiguación y su masa. En la siguiente tabla se introducen estos valores en el orden mencionado:

Ω_0	h	M
1,2	0,5	113000

Cuadro 2: Parámetros que caracterizan al puente: frecuencia natural, constante de amortiguación y masa.

Los parámetros que caracterizan a los peatones y la inicialización del sistema fueron establecidos de la siguiente manera:

1. La masa de los peatones (m) se estableció en $m=70$.
2. Las frecuencias ω_i (o en este caso simplemente ω) no se corresponderán realmente con la frecuencia de los peatones, esta estará controlada por los parámetros

- λ y a . El valor de estas constantes se estableció en $\lambda = 0,5$ y $a = 1$.
3. La posición lateral inicial del puente se estableció como $y(0) = -1$.
 4. El grado de acoplamiento entre peatones se describe con la siguiente constante:

$$r = \frac{m}{M + nm} \quad (41)$$

5. Por último se les asignó a los valores $x_i(0)$ un valor aleatorio perteneciente al intervalo $[-1,1]$.

Los resultados de la simulación los observamos en los mapas de calor de la figura (9), dónde hemos representado primero el seno de las fases de los peatones para un número de 50 peatones (muy por debajo del número crítico) donde vemos que efectivamente no hay ningún tipo de sincronización. Luego representamos estos mismos senos para un número de 170 peatones dónde hemos sobrepasado el número crítico, por lo que observamos una clara coordinación entre ellos, verificando así el modelo estudiado.

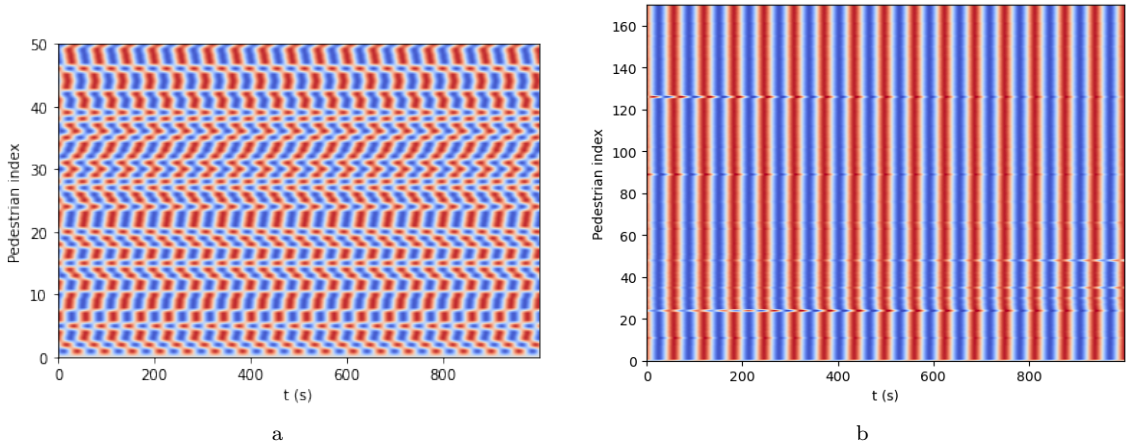


Figura 9: Mapas de calor del seno de las fases de los peatones en función del tiempo. (a): Región en la que encontramos a un número de peatones por debajo del número crítico (no hay sincronización por lo que el patrón de la figura es completamente aleatorio). (b): Región donde se ha sobrepasado el número crítico, los peatones se han coordinado (phase locking, el patrón de la figura refleja una clara coordinación entre los peatones en el tiempo).

Otra manera de observar el phase locking es representar la amplitud máxima del puente en función del número de peatones y observamos un crecimiento brusco a partir del número crítico. Figura (12):

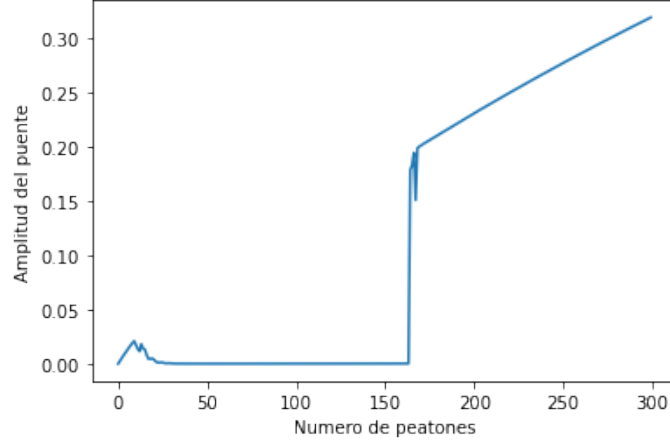


Figura 10: Representación de la amplitud del puente en función del número de peatones para el modelo de van der-Pol con peatones idénticos

C.4.2. Modelo de péndulos invertidos

Por último llevamos a cabo una simulación del último modelo presentado por [2]. Esta nueva simulación podía llevarse acabo simplemente caracterizando nuevamente $f(x, \dot{x}, t)$ de la ecuación (5) para el caso de péndulos invertidos, (23), además de ajustar otros parámetros.

Cabe mencionar aquí que el artículo no facilita para nada la recreación de este modelo ya que establecer las condiciones iniciales es una tarea muy ardua, tanto, que los tutores de este mismo trabajo tuvieron que contactar en su día con los autores del artículo para poder recrearlo. Igualmente, la caracterización del puente se hizo con los siguientes parámetros: (tienen el mismo significado físico que para el caso de Van der-Pol)

Ω_0	h	M	v
1,21	0,05	113000	0.66

Cuadro 3: Parámetros que caracterizan al puente: frecuencia natural, constante de amortiguación, masa y v . Este último parámetro controla el periodo de los ciclos límite.

Además, los siguientes cálculos e inicializaciones fueron necesarias:

1. Los peatones y puente en el instante cero ($x_i(0)$ y $y(0)$) fueron inicializados según la ecuación (25). Se generaron uniformemente fases aleatorias φ_i de 0 a 2π . Para las constantes B_i se utilizó la ecuación (34), y para la constante A se utilizó la ecuación (38) y finalmente, para ψ a partir de (39) se puede llegar a:

$$\psi = \bar{\varphi} - \arctan \frac{2h}{1 - \Omega^2} \quad (42)$$

2. A los peatones se les asignó esta vez las siguientes constantes:

m	a	λ	p
70	1.0	2.8	2.0

Cuadro 4: Parámetros que caracterizan a los peatones. Las constantes tienen el mismo significado físico que en el modelo de Van der-Pol. La nueva constante p introducida, establece la posición lateral del centro de masas del péndulo (peatón) tal y como describe la Figura (6).

Además se generaron para estos unas frecuencias ω_i uniformemente aleatorias comprendidas en el siguiente intervalo: $[0.6935, 0.7665]$.

3. Finalmente, se volvió a definir el grado de acoplamiento entre peatones según la ecuación (41).

Una vez inicializado el sistema, obtuvimos los resultados que se predijeron en el desarrollo analítico: obteniendo una sincronización (aunque no tan perfecta en nuestro caso como en los modelos anteriores) a partir del número crítico $n_c = 165$:

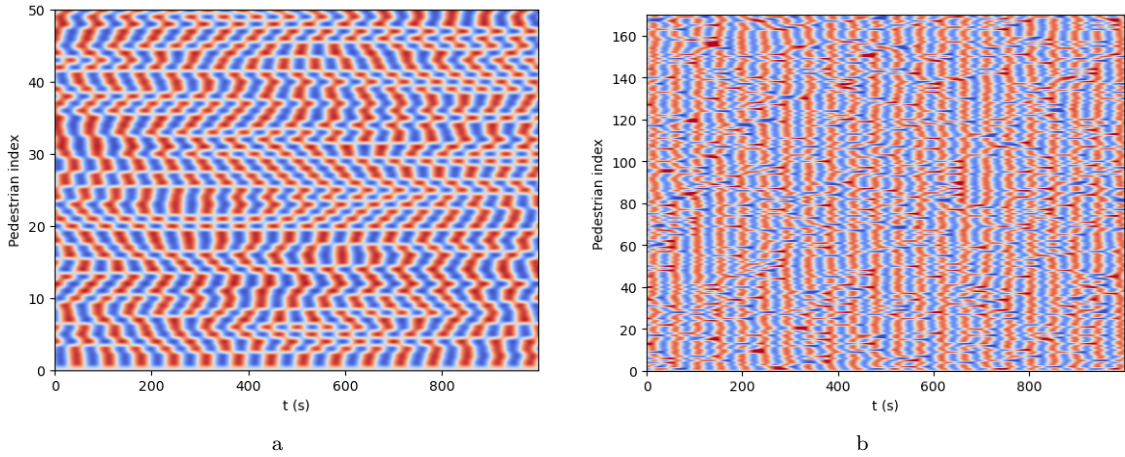


Figura 11: Mapas de calor del seno de las fases de los peatones en función del tiempo. (a): Región en la que encontramos a un número de peatones por debajo del número crítico (no hay sincronización por lo que el patrón de la figura es completamente aleatorio). (b): Región donde se ha sobrepasado el número crítico, los peatones se han coordinado (phase locking, el patrón de la figura refleja una clara coordinación entre los peatones en el tiempo).

Finalmente, como en el caso anterior llevamos a cabo 300 simulaciones, cada una de ellas con un número distinto de peatones para comprobar el cambio de fase de coordinado a no coordinado. En efecto, encontramos que la amplitud máxima del puente en función del número de peatones sufre un aumento muy abrupto cuando sobrepasamos el número crítico, tal y como se muestra en la siguiente figura:

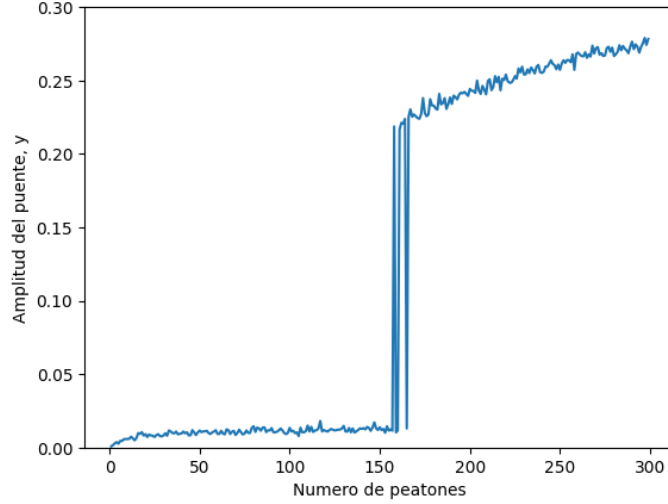


Figura 12: Representación de la amplitud del puente en función del número de peatones para el modelo de péndulos invertidos con peatones no-idénticos.

D. Conclusiones

En este trabajo hemos tenido la oportunidad de descubrir la importancia de modelar un sistema tan crítico como lo es un puente, sobretodo para la prevención de accidentes. Desde hace muchos años se es consciente de que un gran número de peatones caminando sobre puentes a un mismo ritmo es algo peligroso, de ahí el famoso dicho 'rompan filas' de los soldados debido a que provocaban una oscilación considerable cada vez que desfilaban por los puentes. Pero de lo que no se era tan consciente es de la capacidad que tiene el sistema formado por los peatones y el puente de sincronizarse a un cierto número de caminantes sobre él, sin importar que el comienzo de su marcha sea aleatorio y no estén inicialmente sincrónicos. Es algo bello de observar bajo nuestro punto de vista, como un sistema aleatorio se llega a sincronizar al estar acoplado con la superficie sobre la que se apoya. En el caso del Millennium Bridge, la interacción puente-peatón juega un papel crucial en la aparición de estas grandes amplitudes de oscilación. En condiciones de phase locking, un fenómeno colectivo, las partes de un sistema "se ponen de acuerdo" para adquirir en común una fase determinada, lo que se traduce en un fenómeno de sincronización.

El primer modelo estudiado ha sido el de Strogatz y nos permitió calcular mediante una simulación en python de las ecuaciones diferenciales acopladas descriptivas del sistema, el número crítico de peatones obteniendo $N_c \approx 150$. Es un modelo muy intuitivo y sencillo de implementar, la mayor dificultad que nos hemos encontrado ha sido la discusión de unidades de las variables, que no consideramos que se ilustrasen con claridad absoluta en ([4]) y ([1]). En este documento también resumimos los cálculos analíticos realizados por Strogatz para la obtención del número crítico y

nos encontramos con que este depende de la probabilidad de que un peatón camine oscilando a la frecuencia natural del puente.

Con los modelos de Belykh vimos otra manera de enfocar el sistema, más detallado con respecto a considerar parámetros biológicos ya que se centraba más en conseguir un modelaje más realista del caminar de los peatones. Se estudian los modelos de van der-Pol y el modelo del péndulo invertido, ambos modelos se consiguen unificar en una sola ecuación genérica en la que cada uno de los modelos presenta una función descriptiva de la reacción de los peatones ante el puente distinta en cada caso. Con este modelo nos hemos decepcionado un poco, ya que a la hora de implementarlo vimos que no eramos capaces de conseguir el phase locking y el motivo de este problema fue que era necesario considerar en las condiciones iniciales del puente una oscilación ya existente. Esto es en realidad algo muy poco realista. A pesar de esto, se obtienen unos resultados en ambos casos de $N_c = 165$ en las simulaciones, algo predecido previamente por un desarrollo analítico. Mencionar que en el caso del modelo de van der-Pol distingue entre caminantes idénticos y caminantes no idénticos, pero aún siendo más realista el segundo caso, los resultados obtenidos son los mismos y esto es debido a que finalmente los caminantes no idénticos se compensan sus contribuciones dispares llegando a una media aproximada del colectivo siendo similar a los caminantes idénticos.

En resumen, los dos modelos estudiados han sido modelos muy completos pero consideramos que es de destacar el modelo de Strogatz ante los modelos de Belykh ya que estos segundos son mucho más complicados de analizar y por un lado intentaron ser más realistas pero con las condiciones iniciales de las que es necesario partir, se pierde el realismo conseguido. Mientras que Strogatz es un modelo muy intuitivo y sencillo de implementar. Además, la forma de implementar el algoritmo es más realista: en el primer modelo el número de peatones va aumentando progresivamente, tal y como sucedería en un puente. En el segundo modelo solo podemos considerar que los peatones aparecen de forma abrupta. Durante las simulaciones se ha intentado añadir peatones *a lo Strogatz* no obteniendo los deseados estados de phase locking.

Por último, como crítica constructiva a [2], nos gustaría expresar la dificultad que hemos encontrado a la hora de caracterizar el sistema (inicialización de constantes y variables), ya que sin la ayuda de los tutores de la asignatura, quienes previamente se pusieron en contacto con el autor para clarificar nuestras mismas dudas, no hubiese sido posible llevar a cabo una gran parte de las simulaciones de este documento.

E. ANEXO: Códigos de Python

E.1. Modelo de Strogatz

```
%matplotlib notebook
import numpy as np
import pandas as pd
from scipy.integrate import solve_ivp
from numba import njit
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
from matplotlib.colors import Normalize
from matplotlib.animation import PillowWriter

@njit
def sistema(t, X, omega, omega_0, G, M, B, K, C, alpha):

    #X es un array con todas las variables del sistema

    x = X[0]
    V = X[1]
    theta = X[2:]

    #defino las listas y variables necesarias y que son dependientes del numero de personas (N)

    A = np.sqrt(x**2 + (V/omega_0)**2)
    Psi = np.arctan2(x, V/omega_0)

    #definimos el sistema de ecuaciones diferenciales acopladas
    dx_dt = V
    dV_dt = (G/M)*np.sum(np.sin(theta))-B*V/M-K*x/M
    dtheta_dt = omega + C*A*np.sin(Psi - theta + alpha)

    #preparamos una lista que devolver la X,V y las thetas
    return_list = [dx_dt, dV_dt]

    return return_list + list(dtheta_dt)

aux_complex = 1j
#CONSTANTES DEL PUENTE

M = 1.13e5 #kg
B = 1.10e4 #kg/s
K = 4.73e6 #kg/s^2
G = 30 #kg*m/s^2
C = 16 #n^-1 s^-1
```

```

alpha = np.pi/2 #rad
omega_0 = np.sqrt(K/M) #rad/s
std.omega = 0.63 #rad/s

#CONSTANTES DEL ALGORITMO

t_max = 2000 #tiempo considerado
t = np.arange(0, t_max-1, 1) #array con todos los tiempos
Nt = np.zeros(t_max-1) #array tal que N[t] es el numero de personas en t

t_start_50 = 250 #(s) #a adimos las 50 personas en t=250
t_stop_50 = 500 #(s) #empiezan los incrementos de 10 personas en t=500
t_step = 100 #(s) #el tiempo entre incrementos de 10 personas
t_final = t_max-200 #(s) #tiempo q dejamos al final de la grafica con N=cte

#llenamos el array Nt desde t_start_50 a t_stop_50 con 50 personas
for i in range(t_stop_50-t_start_50):
    Nt[i+t_start_50] = 50

#llenamos Nt con todas las N en cada uno de los t. Sumando diez en cada paso
for i in range(int((t_max-t_stop_50)/t_step)-1):
    for j in range(t_step):
        Nt[t_stop_50+i*t_step+j] += (i+1)*10 + 50

#dejamos como en el paper un numero cte de N los ultimos segundos
for i in range(t_final, t_max-1):
    Nt[i] = 50 + ((t_final-t_stop_50)/t_step+1)*10

N_max = int(Nt[-1])
#array con los resultados a plotear
X_results = []
R_results = []
V_results = []

#***** REGION 1: N=0 desde 0 a t_start_50
#en esta regi n no hay ni gente ni balanceo, metemos ceros en resultados

X_results.extend(np.zeros(t_start_50-1))
V_results.extend(np.zeros(t_start_50-1))
R_results.extend(np.zeros(t_start_50-1))
tetha_results=np.zeros((N_max, t_max))

#***** REGION 2: N=50 desde t_start_50 a t_stop_50

```

```

N=50
initial_conditions = 2*np.pi*np.random.rand(N+2)
initial_conditions[0] = 0
initial_conditions[1] = 0          #puente en reposo

#inicializamos las omegas de cada peaton
omega = np.random.normal(omega_0, std_omega, N)

#resolvemos la ecuación diferencial
args = [omega, omega_0, G, M, B, K, C, alpha]
t_eval = np.arange(t_start_50, t_stop_50, 1)
sol = solve_ivp(sistema, (t_start_50, t_stop_50-1),
                initial_conditions, t_eval=t_eval, args=args)

X_results.extend(sol.y[0])
V_results.extend(sol.y[1])
for k in range(N_max):
    if k < N:
        tetha_results[k][t_start_50:t_stop_50] = sol.y[k+2]

#calculamos el par metro de orden en cada t
for tt in range(len(sol.y[0])):
    aux = 0.0
    for i in range(N):
        aux += np.exp(sol.y[i+2][tt]*aux_complex)
    R_results.append(abs(aux/N))

***** REGION 3: Aumentamos N en 10 en cada paso de 100s

for i in range(int((t_final-t_stop_50)/t_step)):

    #numero de personas en el paso i
    N = int(50 + 10*(i+1))

    #inicializamos el estado inicial con el ultimo valor de cada variable
    #del paso anterior
    initial_conditions = np.concatenate((initial_conditions,
                                         2*np.pi*np.random.rand(10)))

    initial_conditions[0] = sol.y[0][-1]
    initial_conditions[1] = sol.y[1][-1]
    for k in range(int(N-10)):
        initial_conditions[k+2] = sol.y[k+2][-1]

    #a adimos nuevas frecuencias asociadas a los nuevos peatones
    omega = np.concatenate((omega, np.random.normal(omega_0, std_omega, 10)))
    args = [omega, omega_0, G, M, B, K, C, alpha]

```



```

t_eval = np.arange(t_stop_50+i*t_step, t_stop_50+i*t_step+t_step, 1)
sol = solve_ivp(sistema, (t_stop_50+i*t_step,
                           t_stop_50+i*t_step+t_step-1),
               initial_conditions, t_eval=t_eval, args=args)

X_results = np.concatenate((X_results, sol.y[0]))
V_results = np.concatenate((V_results, sol.y[1]))
for k in range(N_max):
    if k < N:
        tetha_results[k][t_stop_50+i*t_step:t_stop_50+
                           i*t_step+t_step] = sol.y[k+2]

#calculamos el par metro de orden en cada t
for tt in range(len(sol.y[0])):
    aux = 0.0
    for i in range(N):
        aux += np.exp(sol.y[i+2][tt]*aux_complex)
    R_results.append(abs(aux/N))

# ***** REGION 4: Mantenemos N constante desde t_final a t_max

#a adimos las ultimas diez personas
N += 10
omega = np.concatenate((omega, np.random.normal(omega_0, std_omega, 10)))
args = [omega, omega_0, G, M, B, K, C, alpha]

#inicializamos por ultima vez las condiciones iniciales
initial_conditions = np.concatenate((initial_conditions, 2*np.pi*np.random.rand(10)))
initial_conditions[0] = sol.y[0][-1]
initial_conditions[1] = sol.y[1][-1]
for i in range(int(N-10)):
    initial_conditions[i+2] = sol.y[i+2][-1]

#ultima integracion
t_eval = np.arange(t_final, t_max, 1)
sol = solve_ivp(sistema, (t_final, t_max-1), initial_conditions,
               t_eval=t_eval, args=args)

X_results = np.concatenate((X_results, sol.y[0]))
V_results = np.concatenate((V_results, sol.y[1]))
for k in range(N_max):
    if k < N:
        tetha_results[k][t_final:t_max] = sol.y[k+2]

#calculamos el par metro de orden en cada t

```

```

for tt in range(len(sol.y[0])):
    aux = 0.0
    for i in range(N):
        aux += np.exp(sol.y[i+2][tt]*aux_complex)
    R_results.append(abs(aux/N))

***** RESULTADOS

A = np.sqrt(X_results**2+(V_results/omega_0)**2)

#Creamos un dataframe para visualizar todos los resultados
df_data = {"Tiempo_(s)": t, "Personas": Nt, "Amplitud_de_oscilacion": A*100,
           "Parametro_de_orden": R_results}
df = pd.DataFrame(df_data)

axes = df.plot(0, [1, 2, 3],subplots=True, grid=True, figsize=(8, 6), legend=False)
axes[0].set_ylabel('Crowd_Size')
axes[1].set_ylabel('Wobbling_Amplitude_(cm)')
axes[2].set_ylabel('Order_Parameter')
axes[0].set_ylim(0,200)

plt.show()

tetha_results_sin = np.sin(tetha_results)

```

E.2. Modelo de Van der-Pol (Peatones idénticos)

```

import numpy as np
# import pandas as pd
from scipy.integrate import solve_ivp
from numba import njit
import matplotlib.pyplot as plt
# from matplotlib.animation import FuncAnimation
from matplotlib.colors import Normalize
# from matplotlib.animation import PillowWriter

@njit
def RK4(N, x0, h, tmax):
    # Rk4 method with parameters:
    # N->Dimensionality of the system
    # x0->Initial condition
    # h->Time step
    # tmax->Maximum time for the simulation
    # params->Parameters for the function of evaluation

    T=int(tmax/h)+1
    t=np.linspace(0,tmax,T)
    X=np.zeros((N,T))

```

```

X[:,0]=x0
h2=h/2
h6=h/6

for tt in range(1,T): #loop on times, from 1 to T (the time 0 has already been set)

    #Time dependent constants

    P=int (N/2-2)
    if P <= 165:
        omega = 1.097
    else:
        omega = np.sqrt(1 + (m*P)/
                        ((OMEGA**2-1+4*hh**2/(OMEGA**2-1))*(m*P+M)))

    r=m/(M+m*P)

    #Params to pass
    params = [N_max, r, omega, lamb, OMEGA, a, P]

    k1=func(X[:,tt-1], t[tt-1], params)
    k2=func(X[:,tt-1]+h2*k1, t[tt-1]+h2, params)
    k3=func(X[:,tt-1]+h2*k2, t[tt-1]+h2, params)
    k4=func(X[:,tt-1]+h*k3, t[tt-1]+h, params)
    X[:,tt]=X[:,tt-1]+h6*(k1+2*k2+2*k3+k4) #update the vector X according to the method

return t,X

#@njit
def func(X, t, params):
    # X[0]=y
    # X[1]=vy
    # X[2:n+2]=xi
    # X[n+2:]=vxi

    #n is the number of max pedestrians, P is the number of pedestrian in the
    #current iteration
    n, r, omega, lamb, OMEGA, a, P = params

    fVP=lamb*(X[n+2:]**2+X[2:n+2]**2-a**2)*X[n+2:]+omega**2*X[2:n+2]
    suma=-np.sum(fVP)
    suma=(suma+P*(2*hh*X[1]+OMEGA**2*X[0]))/(1-P*r)

    F=np.zeros(len(X))

    F[0] = X[1]
    F[2:n+2] = X[n+2:]
    F[1] = -r*suma-2*hh*X[1]-OMEGA**2*X[0]
    F[n+2:] = -F[1]-fVP

```

```

    return F

#CONSTANTES DEL PUENTE
OMEGA = 1.2
lamb = 0.5
hh= 0.05
m = 70
M = 113000
a = 1

#CONSTANTES DEL ALGORITMO
t_max = 3000
h=0.1

N_max = 50 #Numero de personas a considerar en el puente

#LAS CONDICIONES INICIALES
initial_conditions = np.zeros(2*N_max+2)
initial_conditions[0] = -1 #y(0)
#inicializamos las x de -1 a 1
initial_conditions[2:2+N_max] = 2*np.random.rand(N_max)-1

t,X = RK4(2*N_max+2, initial_conditions , h, t_max)

first_map = X[1:, 20000:21000]
fig, ax = plt.subplots()
im = ax.imshow(first_map , cmap='coolwarm' , aspect='auto' ,
               norm=Normalize(vmin=-1, vmax=1))
ax.set_xlabel('t(s)')
ax.set_ylabel('Pedestrian_index')
ax.set_ylim(0, N_max)

fig2, ax2 = plt.subplots()
plt.plot(t, X[0])
ax2.set_ylabel("y")
ax2.set_xlabel("t")

plt.show()

```

E.3. Modelo de los péndulos invertidos.

```

import numpy as np
from scipy.integrate import solve_ivp
from numba import njit
import matplotlib.pyplot as plt

```

```
from matplotlib.colors import Normalize
```

```
def RK4(N, x0, h, tmax):
```

```
    T=int(tmax/h)+1
```

```
    t=np.linspace(0,tmax,T)
```

```
    X=np.zeros((N,T))
```

```
    X[:,0]=x0
```

```
    h2=h/2
```

```
    h6=h/6
```

```
    for tt in range(1,T):
```

```
        #Params to pass
```

```
        params = [N_max, r, omega, lamb, OMEGA, a, v]
```

```
        k1=func(X[:,tt-1], t[tt-1], params)
```

```
        k2=func(X[:,tt-1]+h2*k1, t[tt-1]+h2, params)
```

```
        k3=func(X[:,tt-1]+h2*k2, t[tt-1]+h2, params)
```

```
        k4=func(X[:,tt-1]+h*k3, t[tt-1]+h, params)
```

```
        X[:,tt]=X[:,tt-1]+h6*(k1+2*k2+2*k3+k4) #update the vector X according to the method
```

```
    return t,X
```

```
def func(X, t, params):
```

```
    # X[0]=y
```

```
    # X[1]=vy
```

```
    # X[2:n+2]=xi
```

```
    # X[n+2:]=vxi
```

```
    n, r, omega, lamb, OMEGA, a, v = params
```

```
    fIV = np.zeros(n)
```

```
    for ii in range(n):
```

```
        if X[2+ii] >= 0:
```

```
            fIV[ii] = -omega[ii]**2*(X[2+ii]-p)+
```

```
            lamb*(X[n+2+ii]**2-v**2*(X[2+ii]-p)**2+v**2*a**2)*X[n+2+ii]
```

```
        elif X[2+ii] < 0:
```

```
            fIV[ii] = -omega[ii]**2*(X[2+ii]+p)
```

```
            +lamb*(X[n+2+ii]**2-v**2*(X[2+ii]+p)**2+v**2*a**2)*X[n+2+ii]
```

```
    suma=-np.sum(fIV)
```

```
    suma=(suma+n*(2*hh*X[1]+OMEGA**2*X[0]))/(1-n*r)
```

```
    F=np.zeros(len(X))
```

```

F[0] = X[1]
F[2:n+2] = X[n+2:]
F[1] = -r*suma-2*hh*X[1]-OMEGA**2*X[0]
F[n+2:] = -F[1]-fIV

    return F

#CONSTANTES DEL PUENTE
OMEGA = 1.21
lamb = 2.8
hh= 0.05
m = 70
M = 113000
a = 1.0
p = 2.0
v = 0.66

#CONSTANTES DEL ALGORITMO
t_max = 3000
h=0.1

#CONSTANTES DE LOS PEATONES
N_max = 170

omega = np.random.rand(N_max)*0.073+0.6935
B=np.sqrt(a**2+(2*hh*(omega**2-1))/(lamb*(OMEGA**2-1)))
phi=2*np.pi*np.random.rand(N_max)
r=m/(M+m*N_max)

#LAS CONDICIONES INICIALES
Ampt=r*N_max*np.mean(B)/np.sqrt((OMEGA**2-1)**2+4*hh**2) #A
PSI = np.mean(phi)-np.arctan(-2*hh/(1-OMEGA**2)) #fase del puente
initial_conditions = np.zeros(2*N_max+2)
initial_conditions[0] = Ampt*np.cos(PSI) #y(0)
initial_conditions[2:2+N_max] = B*np.sin(phi)
initial_conditions[2+N_max:] = B*np.cos(phi)

t,X = RK4(2*N_max+2, initial_conditions , h, t_max)

first_map = X[1:, 20000:21000]
fig, ax = plt.subplots()
im = ax.imshow(first_map , cmap='coolwarm' , aspect='auto' , norm=Normalize(vmin=-1, vmax=1))
ax.set_xlabel('t_(s)')
ax.set_ylabel('Pedestrian_index')
ax.set_ylim(0, N_max)

```

```
fig2 , ax2 = plt.subplots()  
plt.plot(t, X[0])  
ax2.set_ylabel("y")  
ax2.set_xlabel("t")  
  
plt.show()
```

Referencias

- [1] Juan A. Acebrón, L. L. Bonilla, Conrad J. Pérez Vicente, Félix Ritort, and Renato Spigler. The kuramoto model: A simple paradigm for synchronization phenomena. *Rev. Mod. Phys.*, 77:137–185, Apr 2005.
- [2] Igor Belykh, Russell Jeter, and Vladimir Belykh. Foot force models of crowd dynamics on a wobbly bridge. *Science Advances*, 3(11):e1701512, 2017.
- [3] John Guckenheimer and Philip Holmes. *Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields*. 42^a edition, 1983.
- [4] Steven Strogatz, Daniel Abrams, F. Mcrobie, Bruno Eckhardt, and Edward Ott. Crowd synchrony on the millennium bridge. *Nature*, 438:43–4, 12 2005.