

TEMA 1: Fundamentos del modelado de sistemas dinámicos

MODELOS COMPUTACIONALES Y SIMULACIÓN DE SISTEMAS
Curso 2025-2026 - Semanas 2 y 3

Índice

1. ¿Qué es un Sistema Dinámico?
 - a. Componentes de un sistema dinámico.
 - b. Ejemplos de sistemas dinámicos.
2. Solución analítica vs solución numérica
 - a. Ejemplo de solución analítica
 - b. Ejemplo de solución numérica
3. Clasificación de modelos dinámicos
4. Diagramas de fase
5. Representación matemática de sistemas dinámicos: Espacio de Estados.
 - a. Ventajas
 - b. Ejemplos

Índice

- 1. ¿Qué es un Sistema Dinámico?**
 - a. Componentes de un sistema dinámico.
 - b. Ejemplos de sistemas dinámicos.
2. Solución analítica vs solución numérica
 - a. Ejemplo de solución analítica
 - b. Ejemplo de solución numérica
3. Clasificación de modelos dinámicos
4. Diagramas de fase
5. Representación matemática de sistemas dinámicos: Espacio de Estados.
 - a. Ventajas
 - b. Ejemplos

1. ¿Qué es un Sistema Dinámico?

- Un sistema dinámico es un conjunto de elementos interrelacionados cuyo estado, definido por variables de estado, cambia a lo largo del tiempo, generalmente influenciado por entradas (inputs), condiciones iniciales y reglas de evolución.
- Un **sistema dinámico autónomo** se expresa como:

$$\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t)), \quad \mathbf{x}(t) \in \mathbb{R}^n, \quad \mathbf{x}(t=0) = x_0$$

1. ¿Qué es un Sistema Dinámico?

- Esta expresión representa un conjunto de n ecuaciones diferenciales acopladas:

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}, \quad f(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{bmatrix} \quad \left\{ \begin{array}{l} \frac{dx_1}{dt} = f_1(x_1, x_2, \dots, x_n) \\ \frac{dx_2}{dt} = f_2(x_1, x_2, \dots, x_n) \\ \vdots \\ \frac{dx_n}{dt} = f_n(x_1, x_2, \dots, x_n) \end{array} \right.$$

1.a. Componentes de un sistema dinámico

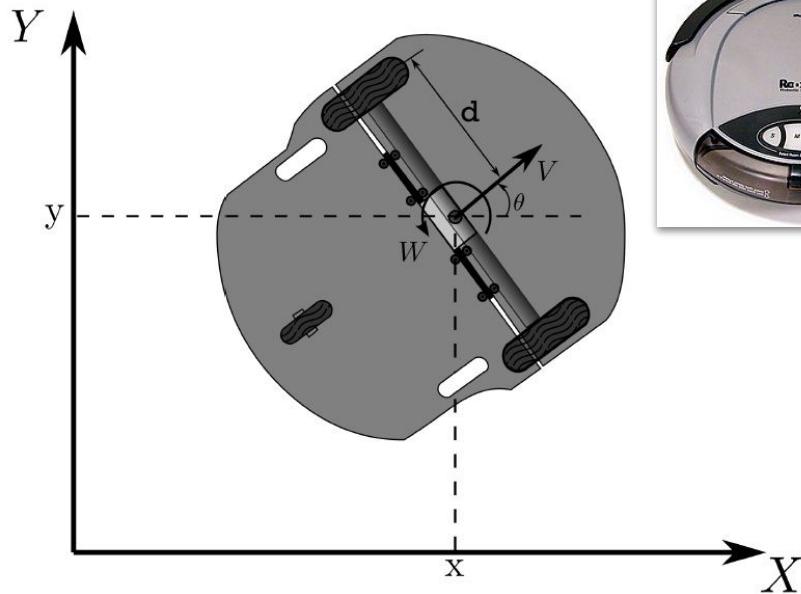
- **Variables de estado:** representan el estado interno del sistema $\mathbf{x}(t)$.
- **Entradas (inputs):** señales externas que afectan al sistema $\mathbf{u}(t)$.
- **Salidas (outputs):** resultados observables del sistema $\mathbf{y}(t)$.
- **Ley de evolución:** describe cómo las variables cambian $d\mathbf{x}(t)/dt$.

Cuando el sistema incluye entradas $\mathbf{u}(t)$, se le denomina
sistema dinámico NO autónomo:

$$\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t), \mathbf{u}(t)), \quad \mathbf{x}(t) \in \mathbb{R}^n, \quad \mathbf{u}(t) \in \mathbb{R}^m, \quad \mathbf{x}(t=0) = x_0$$

1.b. Ejemplos de sistemas dinámicos

- Robot Móvil

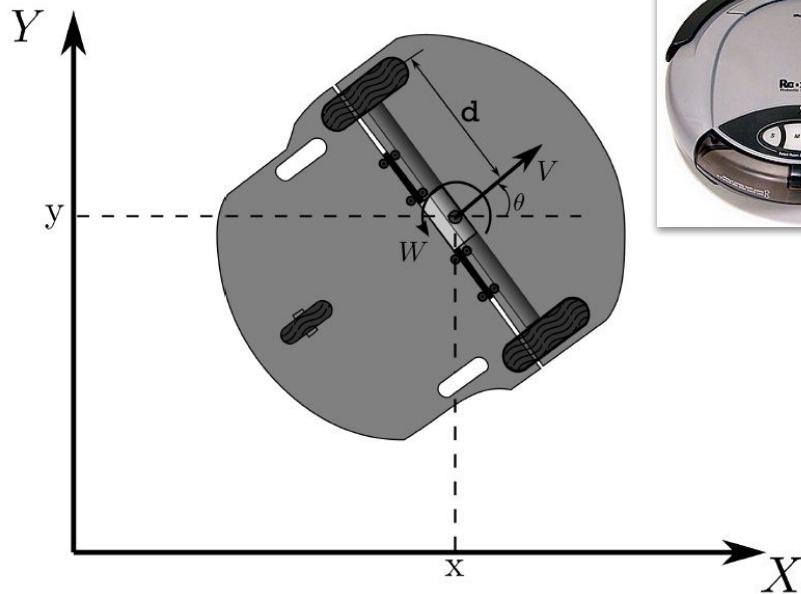


VARIABLES DE ESTADO:

$$\mathbf{x}(t) = \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix}$$

1.b. Ejemplos de sistemas dinámicos

- Robot Móvil

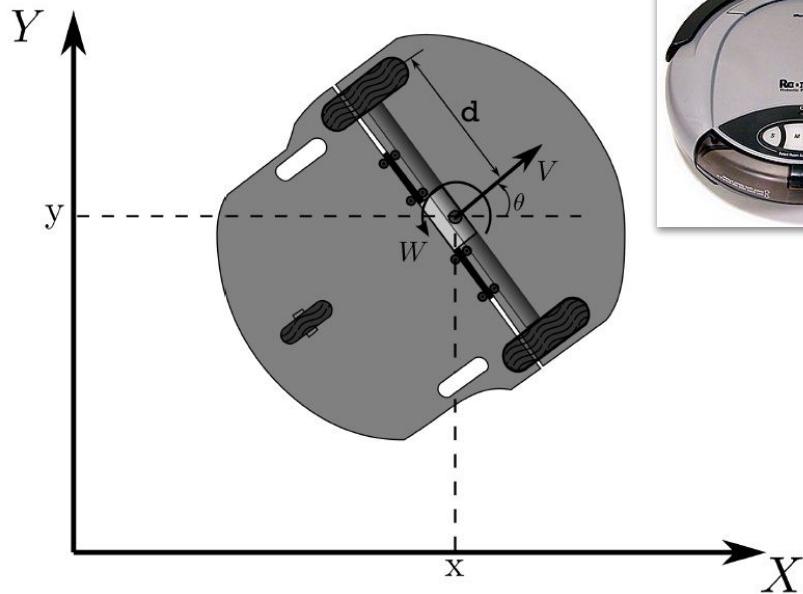


ENTRADAS DEL SISTEMA:

$$\mathbf{u}(t) = \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix}$$

1.b. Ejemplos de sistemas dinámicos

- Robot Móvil

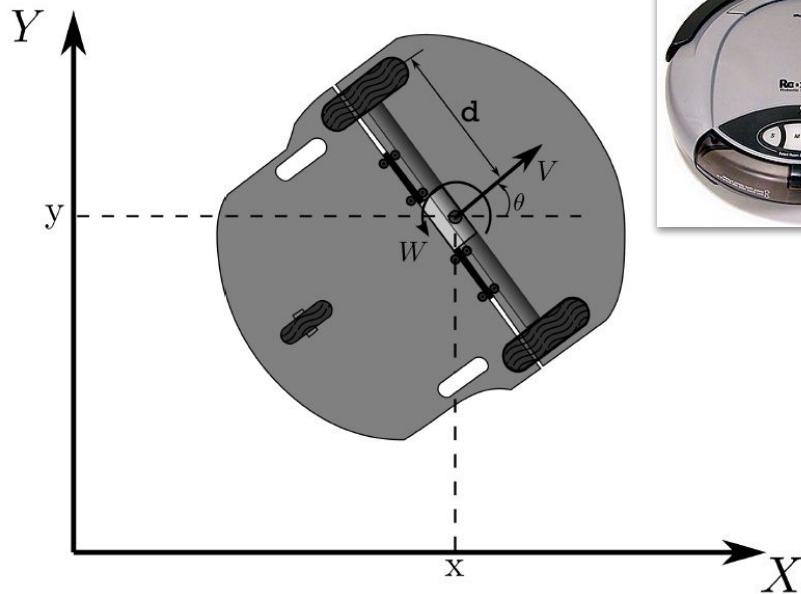


SALIDAS DEL SISTEMA:

$$\mathbf{y}(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}$$

1.b. Ejemplos de sistemas dinámicos

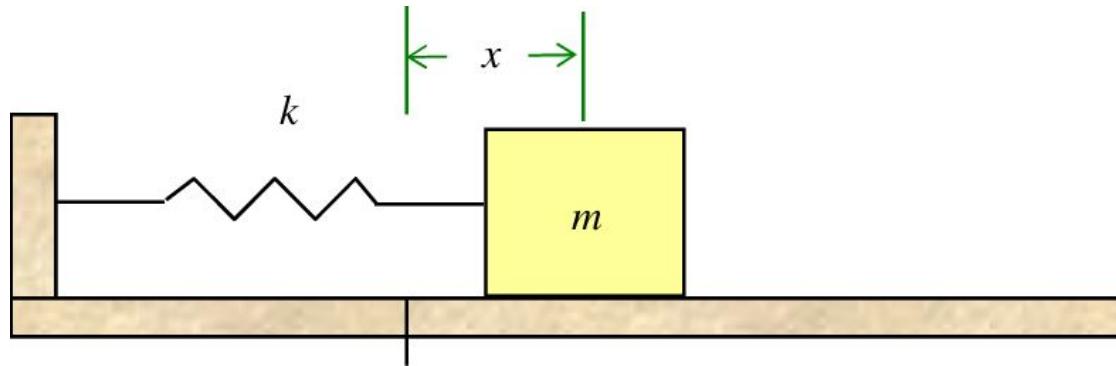
- Robot Móvil



LEY DE EVOLUCIÓN:

$$\frac{d}{dt} \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} = \begin{bmatrix} v(t) \cos(\theta(t)) \\ v(t) \sin(\theta(t)) \\ \omega(t) \end{bmatrix}$$

1.a. Ejemplos de sistemas dinámicos



Sistema	Estado	Entrada	Salida
Masa-resorte	Posición, velocidad	Fuerza $F(t)$	Posición $x(t)$
Población bacteriana	Población $P(t)$	Nutrientes $N(t)$	Tasa de crecimiento
Cuenta bancaria	Saldo x_t	Depósitos/retiros u_t	Saldo
Sistema económico	PIB $Y(t)$, Precios $P(t)$	Políticas fiscales $G(t), I(t)$	Producción, inflación

Índice

1. ¿Qué es un Sistema Dinámico?
 - a. Componentes de un sistema dinámico.
 - b. Ejemplos de sistemas dinámicos.
2. **Solución analítica vs solución numérica**
 - a. **Ejemplo de solución analítica**
 - b. **Ejemplo de solución numérica**
3. Clasificación de modelos dinámicos
4. Diagramas de fase
5. Representación matemática de sistemas dinámicos: Espacio de Estados.
 - a. Ventajas
 - b. Ejemplos

2. Solucion analitica vs solucion numerica

Dado un sistema dinámico en la forma: $\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t))$

Su solución consiste en obtener $\mathbf{x}(t)$.

- Solución analítica:
 - Si es integrable: $\mathbf{x}(t) = \Phi(t; \mathbf{x}_0)$
 - Para analizar el sistema, estabilidad, divergencia, etc (siguiente tema).
- Solución numérica (método de euler):
 - Permite seguir la evolución del sistema en tiempo real
 - Es la solución utilizada para simulación y control:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot f(\mathbf{x}_k)$$

2. Solucion analitica vs solucion numerica

Dado un sistema dinámico en la forma: $\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t))$

Su solución consiste en obtener $\mathbf{x}(t)$.

- Solución analítica:
 - Si es integrable: $\mathbf{x}(t) = \Phi(t; \mathbf{x}_0)$
 - Para analizar el sistema, estabilidad, divergencia, etc (siguiente tema).
- Solución numérica (método de euler):
 - Permite seguir la evolución del sistema en tiempo real
 - Es la solución utilizada para simulación:

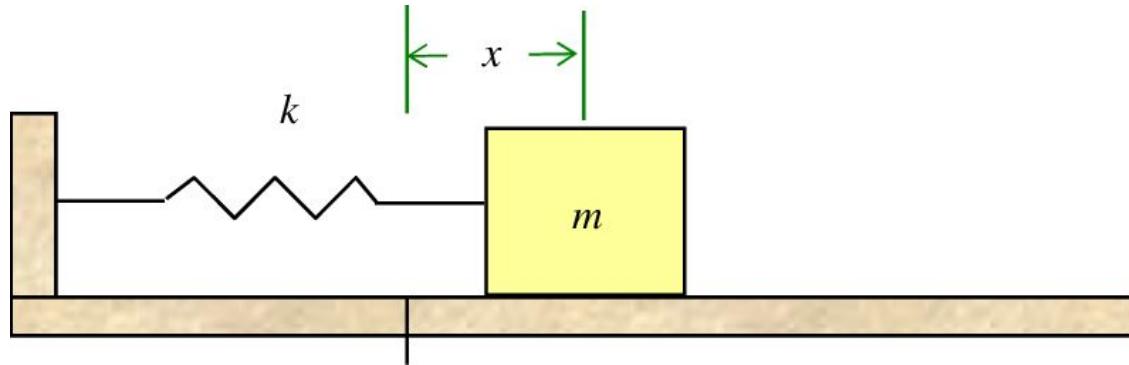
$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot f(\mathbf{x}_k)$$

Ejemplo Robot Móvil

$$\begin{cases} x_{k+1} = x_k + \Delta t \cdot v_k \cos(\theta_k) \\ y_{k+1} = y_k + \Delta t \cdot v_k \sin(\theta_k) \\ \theta_{k+1} = \theta_k + \Delta t \cdot \omega_k \end{cases}$$

Ejemplo: Masa-resorte

- Solución analítica:

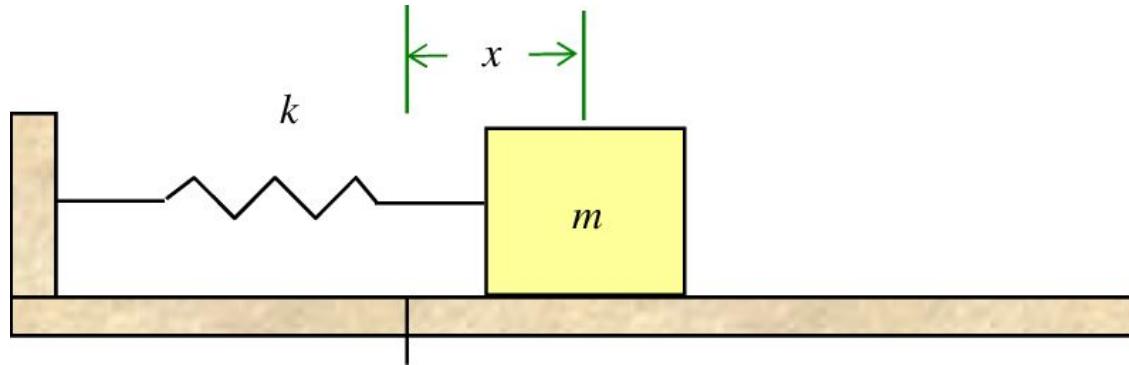


$$m\ddot{x}(t) + kx(t) = 0 \quad \rightarrow \quad \ddot{x}(t) = -\frac{k}{m}x(t)$$

Ejemplo: Masa-resorte

- Solución analítica:

$$\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t))$$



$$x_1(t) = x(t), \quad x_2(t) = \dot{x}(t)$$



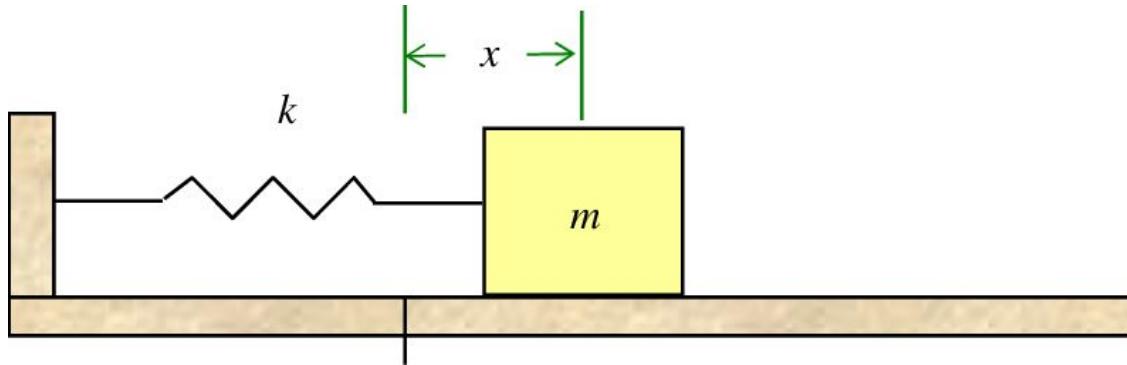
$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -\frac{k}{m}x_1$$

Ejemplo: Masa-resorte

- Solución analítica:

$$\omega = \sqrt{\frac{k}{m}}$$



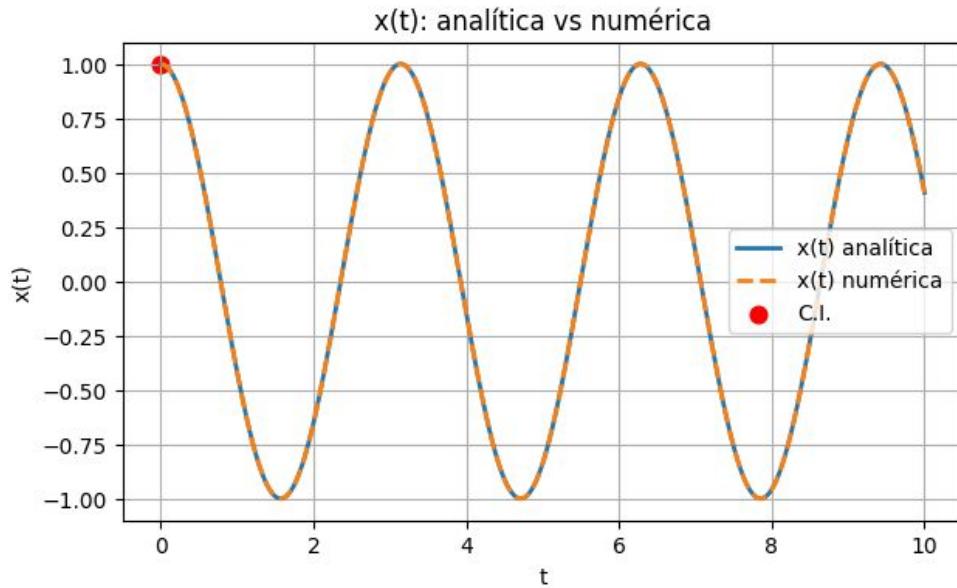
**Solución general
Oscilación armónica**



$$x(t) = A \cos(\omega t) + B \sin(\omega t)$$

Ejemplo: Masa-resorte

- Solución numérica:



$$\begin{cases} x_1^{k+1} = x_1^k + \Delta t \cdot x_2^k \\ x_2^{k+1} = x_2^k - \Delta t \cdot \omega^2 x_1^k \end{cases}$$

Ejemplo: Modelo logístico

- Solución analítica:

$$\frac{dx}{dt} = x(1 - x)$$

Integrando, nos queda:

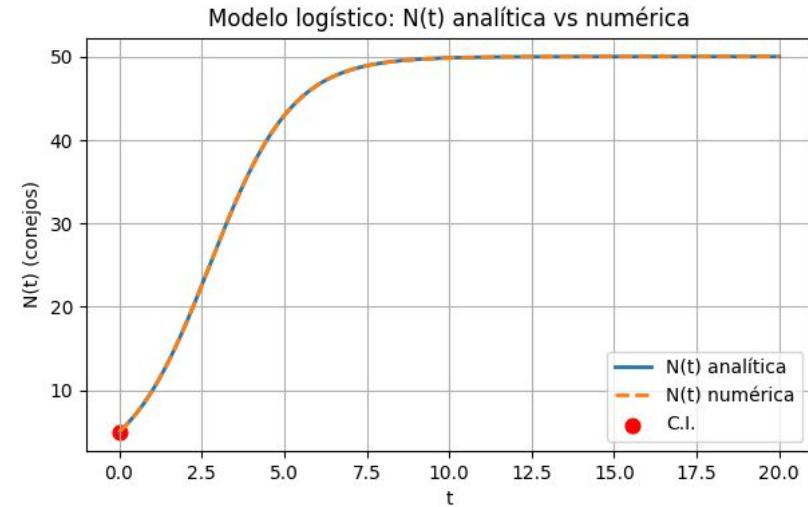
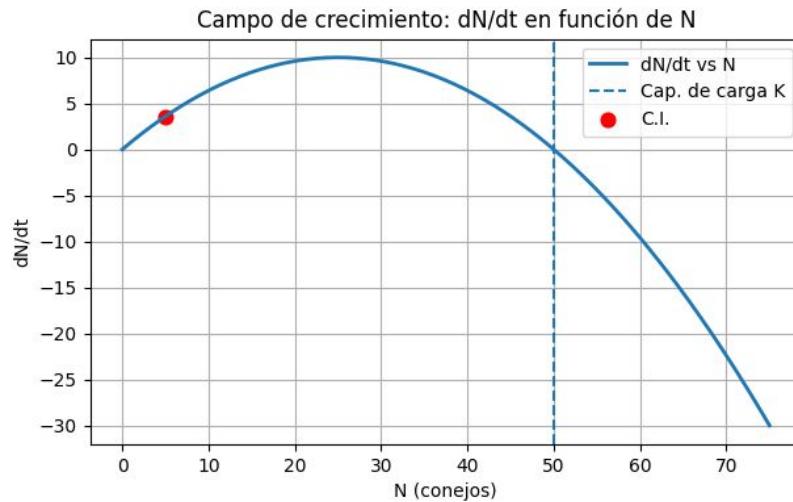
$$x(t) = \frac{Ke^t}{1 + Ke^t}$$

$x(0) = 0.01$

$$\frac{K}{1 + K} = 0,01 \Rightarrow K = \frac{0,01}{0,99} = \frac{1}{99}$$

Ejemplo: Modelo logístico

- Solución numérica: $x_{k+1} = x_k + \Delta t \cdot x_k(1 - x_k)$



Índice

1. ¿Qué es un Sistema Dinámico?
 - a. Componentes de un sistema dinámico.
 - b. Ejemplos de sistemas dinámicos.
2. Solución analítica vs solución numérica
 - a. Ejemplo de solución analítica
 - b. Ejemplo de solución numérica
3. **Clasificación de modelos dinámicos**
4. Diagramas de fase
5. Representación matemática de sistemas dinámicos: Espacio de Estados.
 - a. Ventajas
 - b. Ejemplos

3. Clasificación de modelos dinámicos

- ❖ Según el tipo de incertidumbre
- ❖ Según la naturaleza del tiempo
- ❖ Según la dependencia temporal
- ❖ Según la relación entrada-salida

3. Clasificación de modelos dinámicos

- **Según el tipo de incertidumbre**

- **Modelos deterministas:**

- No incorporan incertidumbre; el estado futuro está completamente determinado por el estado actual y la dinámica del sistema.

- Ejemplo:

$$\frac{dx}{dt} = ax$$

- **Modelos estocásticos:**

- Incluyen términos aleatorios o ruido en la dinámica. El comportamiento es probabilístico.

- Ejemplo:

$$\frac{dx}{dt} = ax + \sigma\xi(t)$$

3. Clasificación de modelos dinámicos

- **Según la naturaleza del tiempo**

- **Modelos en tiempo continuo:**

- El estado evoluciona en un dominio temporal continuo.
 - Ejemplo (masa-resorte):

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -\frac{k}{m}x_1$$

- **Modelos en tiempo discreto:**

- El estado se actualiza en pasos de tiempo separados.
 - Ejemplo (cuenta del banco):

$$x_{k+1} = x_k + u_k$$

3. Clasificación de modelos dinámicos

- Según la dependencia temporal

- Modelos autónomos:

- La dinámica del sistema depende únicamente del estado, no del tiempo explícitamente.
 - Ejemplo (masa-resorte):
$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -\frac{k}{m}x_1$$

- Modelos no autónomos:

- La dinámica depende explícitamente del tiempo o de entradas externas.
 - Ejemplo (robot móvil):

$$\frac{dx}{dt} = f(x, t) \quad \text{o} \quad \frac{dx}{dt} = f(x, u(t)) \quad \rightarrow$$

$$\frac{d}{dt} \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} = \begin{bmatrix} v(t) \cos(\theta(t)) \\ v(t) \sin(\theta(t)) \\ \omega(t) \end{bmatrix}$$

3. Clasificación de modelos dinámicos

- Según la relación entrada-salida

- Modelos lineales:

- La función dinámica es lineal en las variables de estado y entrada.
 - Ejemplo (masa-resorte):

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -\frac{k}{m}x_1$$

$$\frac{dx}{dt} = Ax + Bu$$

- Modelos no lineales:

- Involucran productos, funciones no lineales o términos no lineales en las variables.
 - Ejemplo (modelo logístico):

$$\frac{dx}{dt} = rx \left(1 - \frac{x}{K}\right)$$

Índice

1. ¿Qué es un Sistema Dinámico?
 - a. Componentes de un sistema dinámico.
 - b. Ejemplos de sistemas dinámicos.
2. Solución analítica vs solución numérica
 - a. Ejemplo de solución analítica
 - b. Ejemplo de solución numérica
3. Clasificación de modelos dinámicos
4. **Diagramas de fase**
5. Representación matemática de sistemas dinámicos: Espacio de Estados.
 - a. Ventajas
 - b. Ejemplos

4. Diagramas de fase

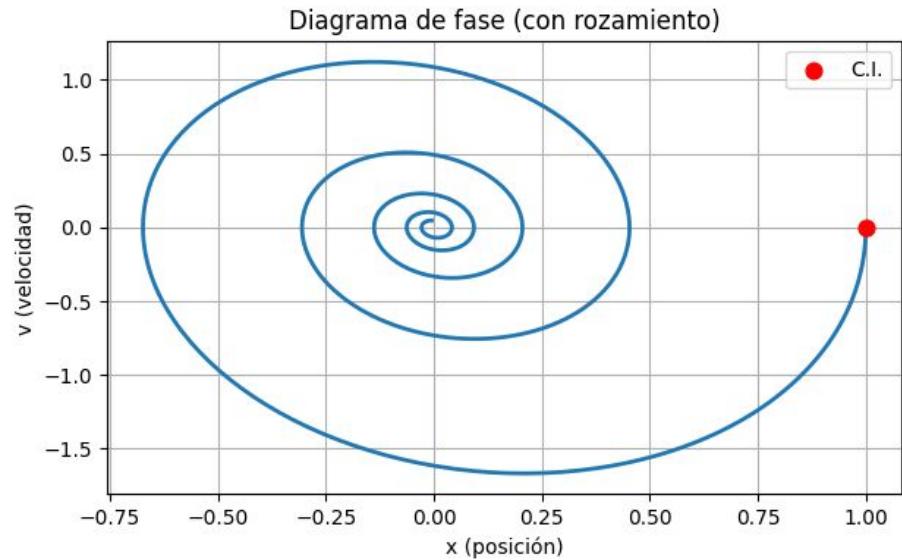
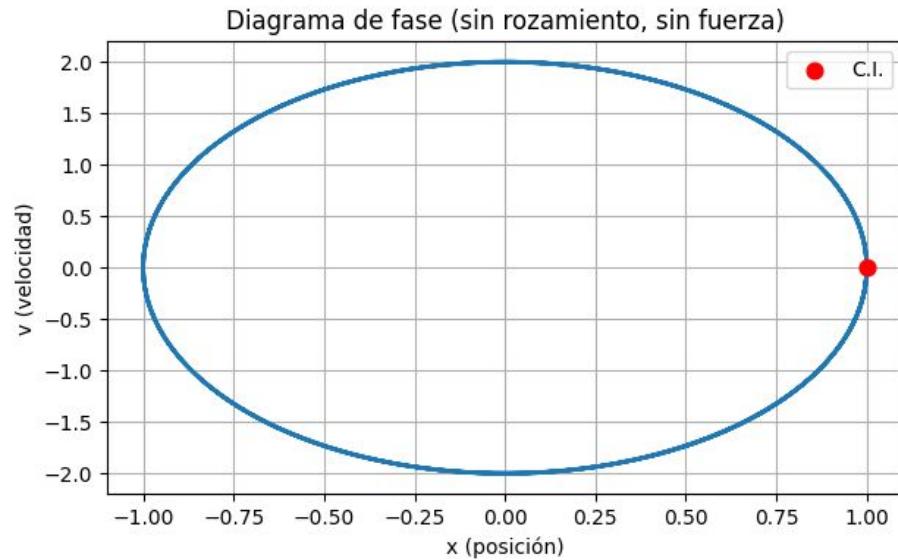
- Los diagramas de fase son una herramienta fundamental para analizar visualmente el comportamiento de sistemas dinámicos
 - Representa la evolución de las variables de estado desde sus condiciones iniciales.
- En un sistema dinámico continuo de dos variables de estado x_1 y x_2 , el diagrama de fase se construye en el plano (x_1, x_2) , donde las trayectorias indican cómo evoluciona el sistema a lo largo del tiempo.
- Ejemplo masa-resorte (siguiente transparencia):

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -\frac{k}{m}x_1$$

4. Diagramas de fase

- Solución: $x(t) = A \cos(\omega t) + B \sin(\omega t)$



Índice

1. ¿Qué es un Sistema Dinámico?
 - a. Componentes de un sistema dinámico.
 - b. Ejemplos de sistemas dinámicos.
2. Solución analítica vs solución numérica
 - a. Ejemplo de solución analítica
 - b. Ejemplo de solución numérica
3. Clasificación de modelos dinámicos
4. Diagramas de fase
5. **Representación matemática de sistemas dinámicos: Espacio de Estados.**
 - a. Ventajas
 - b. Ejemplos

5. Representación: Espacio de Estados.

- Hasta ahora hemos representado los sistema como:

$$\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t), \mathbf{u}(t)), \quad \mathbf{x}(t) \in \mathbb{R}^n, \quad \mathbf{u}(t) \in \mathbb{R}^m, \quad \mathbf{x}(t=0) = x_0$$

- La representación en espacio de estados permite modelar sistemas dinámicos de manera compacta, general y aplicable tanto a sistemas lineales como no lineales, continuos o discretos, y de una o múltiples dimensiones. Este enfoque considera explícitamente:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), t) \\ y(t) = g(x(t), u(t), t) \end{cases} \xrightarrow{\text{LTI}} \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

5.a. Espacio de Estados: Ventajas

- **Ventajas del enfoque en espacio de estados**
 - Permite trabajar con sistemas de orden arbitrario y con múltiples entradas y salidas.
 - Facilita el análisis de estabilidad, controlabilidad y observabilidad.
 - Es la base para técnicas modernas de control automático (control óptimo, control robusto, etc.).
 - Se adapta fácilmente a implementaciones computacionales.

5.b. Espacio de Estados: Ejemplo

- **Ejemplo masa-resorte**

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = -\frac{k}{m}x_1(t) \end{cases} \quad \text{Espacio de estados:} \quad \dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & 0 \end{bmatrix} x(t)$$

En el siguiente tema ...

- Concepto y análisis de estabilidad (puntos de equilibrio)
- Observabilidad y controlabilidad.

TEMA 1: Fundamentos del modelado de sistemas dinámicos

MODELOS COMPUTACIONALES Y SIMULACIÓN DE SISTEMAS
Curso 2025-2026 - Semanas 2 y 3

TEMA 2: Estabilidad, Controlabilidad y Observabilidad

MODELOS COMPUTACIONALES Y SIMULACIÓN DE SISTEMAS
Curso 2025-2026 - Semanas 4 y 5



UNIVERSITAT D'ALACANT
UNIVERSIDAD DE ALICANTE
Escola Politècnica Superior
Escuela Politécnica Superior



Índice

1. Definición de estabilidad en sistemas dinámicos
2. Métodos para analizar estabilidad
 - a. Linealización alrededor de un punto
 - b. Análisis de las raíces del sistema
 - c. Criterio de Routh-Hurwitz
 - d. Método de Lyapunov
3. Análisis del comportamiento
 - a. Análisis Monótono u oscilatorio
 - b. Análisis de amortiguamiento
4. Control
 - a. Controlabilidad
 - b. Observabilidad
 - c. Obtener señal de control $u(t)$

Índice

- 1. Definición de estabilidad en sistemas dinámicos**
2. Métodos para analizar estabilidad
 - a. Linealización alrededor de un punto
 - b. Análisis de las raíces del sistema
 - c. Criterio de Routh-Hurwitz
 - d. Método de Lyapunov
3. Análisis del comportamiento
 - a. Análisis Monótono u oscilatorio
 - b. Análisis de amortiguamiento
4. Control
 - a. Controlabilidad
 - b. Observabilidad
 - c. Obtener señal de control $u(t)$

1. Estabilidad: Puntos de equilibrio

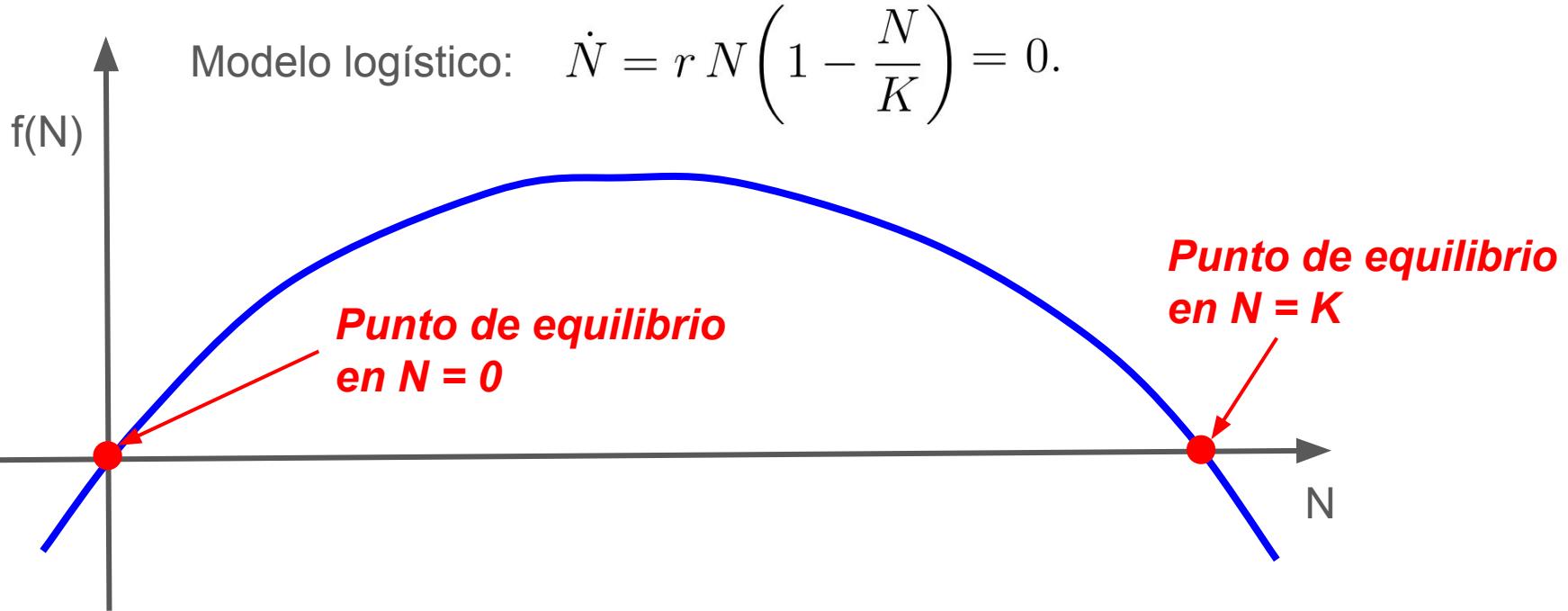
Dado un sistema dinámico:

$$\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t)), \quad \mathbf{x}(t) \in \mathbb{R}^n, \quad \mathbf{x}(t=0) = x_0$$

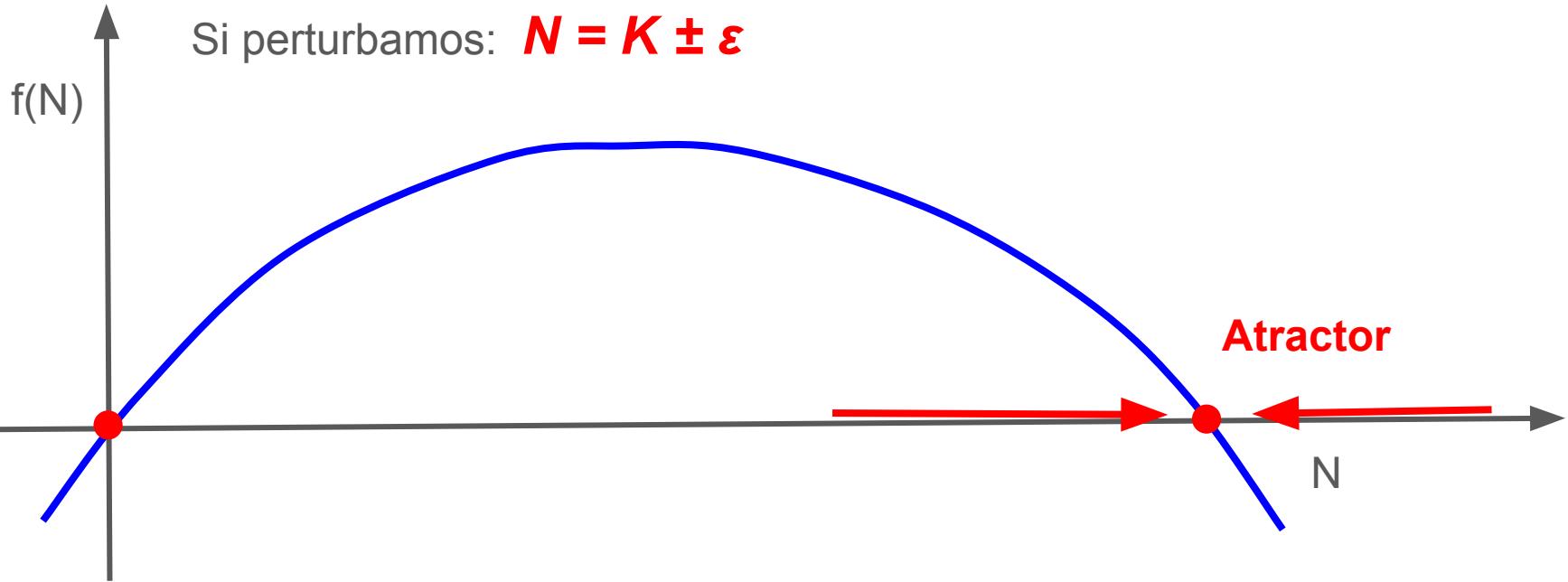
Un punto de equilibrio se define como:

$$f(x_e, u_e) = 0 \quad \rightarrow \quad \mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot f(\mathbf{x}_k)$$

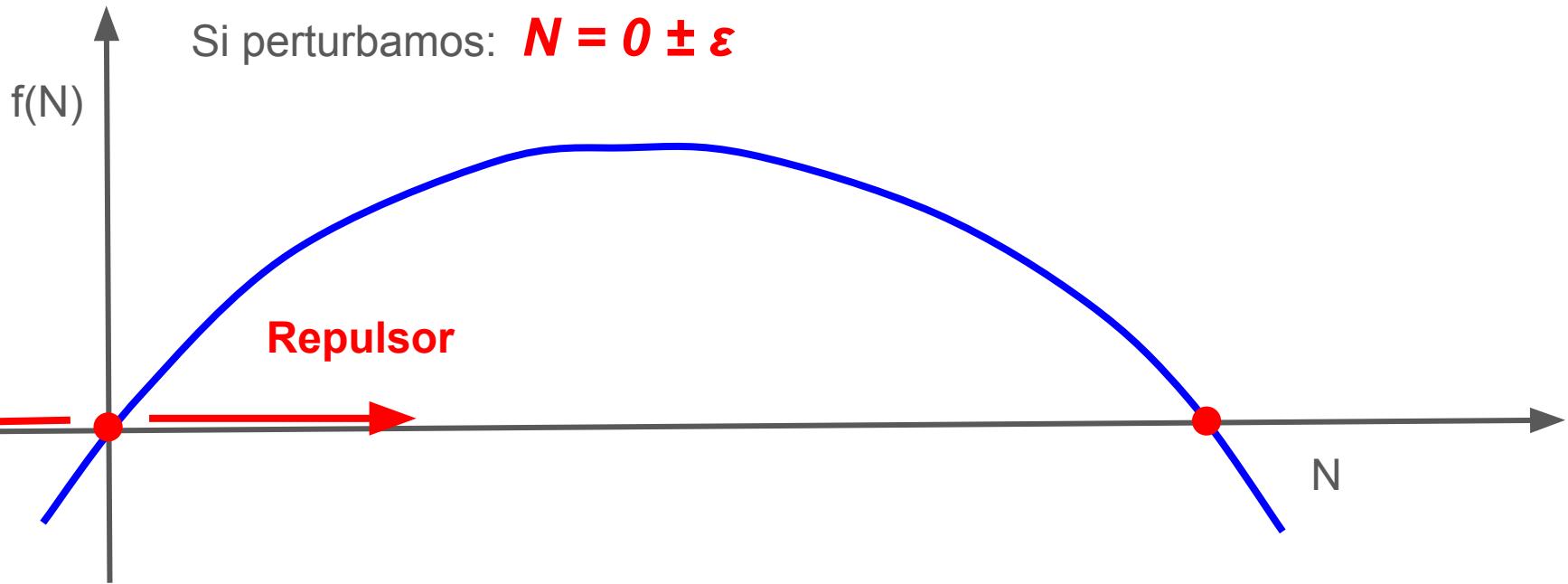
1. Estabilidad: Puntos de equilibrio



1. Estabilidad: Equilibrio estable



1. Estabilidad: Equilibrio inestable



1. Estabilidad: Tipos de estabilidad

- **Estabilidad de Lyapunov:** el sistema es estable si para cada $\epsilon > 0$ existe un $\delta > 0$ tal que si $\|x(0) - x_e\| < \delta$, entonces $\|x(t) - x_e\| < \epsilon$ para todo $t > 0$.
- **Estabilidad asintótica:** además de ser estable, se cumple que:

$$\lim_{t \rightarrow \infty} x(t) = x_e$$

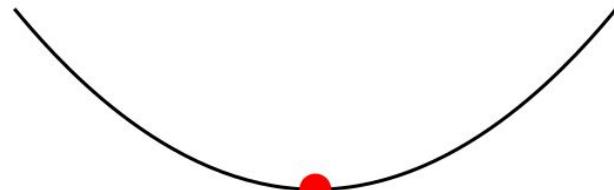
- **Estabilidad exponencial:** existe $M > 0$ y $\alpha > 0$ tal que:

$$\|x(t) - x_e\| \leq M e^{-\alpha t} \|x(0) - x_e\|$$

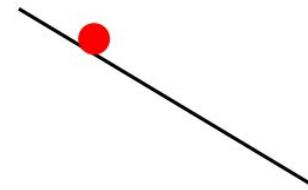
- **Inestabilidad:** si existen condiciones iniciales arbitrariamente cercanas a x_e cuyas trayectorias divergen, el sistema es inestable.

1. Estabilidad

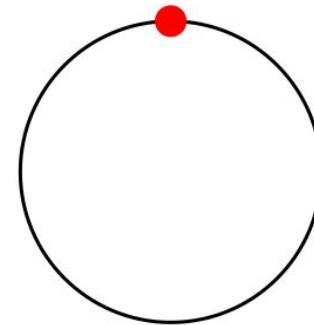
- Ejemplo ilustrativo:



Estable



Inestable



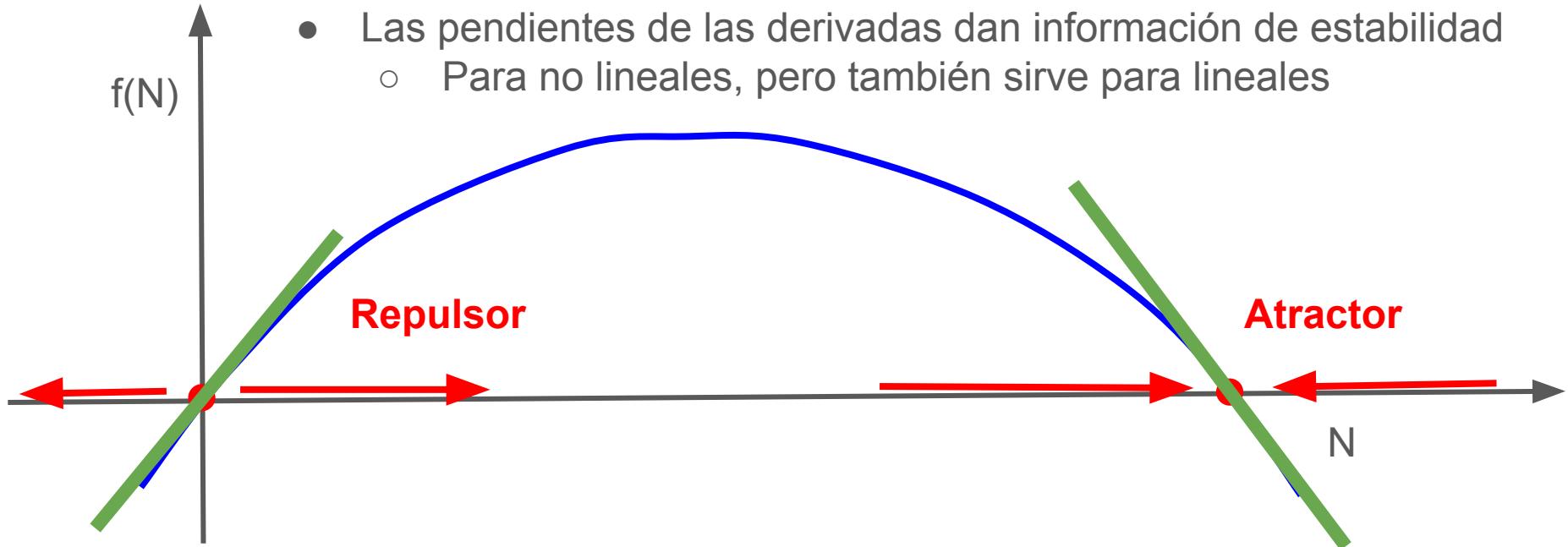
Marginal

Índice

1. Definición de estabilidad en sistemas dinámicos
2. **Métodos para analizar estabilidad**
 - a. Linealización alrededor de un punto
 - b. Análisis de las raíces del sistema
 - c. Criterio de Routh-Hurwitz
 - d. Método de Lyapunov
3. Análisis del comportamiento
 - a. Análisis Monótono u oscilatorio
 - b. Análisis de amortiguamiento
4. Control
 - a. Controlabilidad
 - b. Observabilidad
 - c. Obtener señal de control $u(t)$

2.a. Linealización alrededor de un punto

- Las pendientes de las derivadas dan información de estabilidad
 - Para no lineales, pero también sirve para lineales



2.a. Linealización alrededor de un punto

- Dado un sistema dinámico no lineal

$$\dot{x} = f(x, u, t), \quad y = g(x, u, t)$$

- Se definen variables de pequeña señal alrededor de $f(x_0, u_0) = 0$

$$\delta x := x - x_0, \quad \delta u := u - u_0, \quad \delta y := y - g(x_0, u_0, t)$$

2.a. Linealización alrededor de un punto

- Series de Taylor de primer orden

$$\dot{\delta x} = A \delta x + B \delta u + \mathcal{O}(\|(\delta x, \delta u)\|^2)$$

$$\delta y = C \delta x + D \delta u + \mathcal{O}(\|(\delta x, \delta u)\|^2)$$

- Se obtienen las jacobianas alrededor del punto

$$A = \left. \frac{\partial f}{\partial x} \right|_{(x_0, u_0, t)}, \quad B = \left. \frac{\partial f}{\partial u} \right|_{(x_0, u_0, t)}, \quad C = \left. \frac{\partial g}{\partial x} \right|_{(x_0, u_0, t)}, \quad D = \left. \frac{\partial g}{\partial u} \right|_{(x_0, u_0, t)}$$

2.a. Linealización alrededor de un punto

- Criterio de estabilidad local
 - Se obtienen los autovalores λ de la matriz **A**:
 - Si la parte real es negativa: **ESTABLE**
 - Si alguna parte real es positiva: **INESTABLE**
 - Si la parte real es 0: **MARGINAL**

2.a. Linealización: Ejemplo

- Dado el modelo logístico para el control de población

$$\dot{N} = r N \left(1 - \frac{N}{K} \right)$$

- Se pueden definir las variables de pequeña señal

$$\delta \dot{N} = A \delta N + B \delta u$$

2.a. Linealización: Ejemplo

- En este caso las matrices son 1x1 \Rightarrow

$$C = 1, \quad D = 0.$$

$$A = f'(N^*) = r \left(1 - \frac{2N^*}{K}\right)$$

$$\lambda = A = f'(N^*).$$

- En $N_1^* = 0$: $\lambda = f'(0) = r > 0 \Rightarrow$ autovalor positivo $\Rightarrow \text{INESTABLE}$
- En $N_2^* = K$: $\lambda = f'(K) = -r < 0 \Rightarrow$ autovalor negativo $\Rightarrow \text{ESTABLE}$

2.a. Linealización: Ejemplo

- Dado el modelo logístico para el control de población con variable de control

$$\dot{N} = rN \left(1 - \frac{N}{K}\right) - u(t)$$

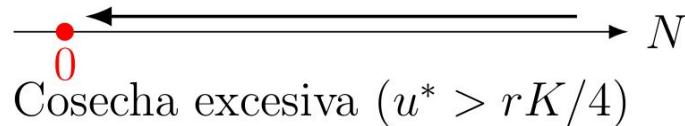
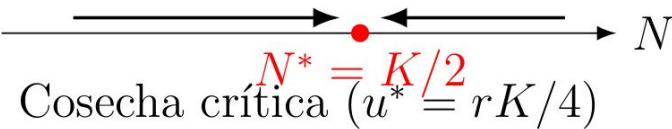
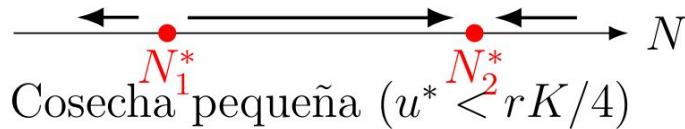
- Se pueden definir las variables de pequeña señal, en este caso **B = 1**

$$\delta\dot{N} = A\delta N + B\delta u$$

2.a. Linealización: Ejemplo

- Los puntos de equilibrio dependen de u

$$N_{1,2}^* = \frac{K}{2} \left(1 \pm \sqrt{1 - \frac{4u^*}{rK}} \right)$$



Índice

1. Definición de estabilidad en sistemas dinámicos
2. **Métodos para analizar estabilidad**
 - a. Linealización alrededor de un punto
 - b. Análisis de las raíces del sistema**
 - c. Criterio de Routh-Hurwitz
 - d. Método de Lyapunov
3. Análisis del comportamiento
 - a. Análisis Monótono u oscilatorio
 - b. Análisis de amortiguamiento
4. Control
 - a. Controlabilidad
 - b. Observabilidad
 - c. Obtener señal de control $u(t)$

2.b. Análisis de las raíces del sistema

- Dada la **ecuación diferencial característica** de un sistema:

$$\ddot{y} + 3\dot{y} + 2y = 0$$

- Aplicando Transformada de Laplace, se obtiene una ecuación algebraica:
 - A eso de le llama polinomio característico

$$P(s) = s^2 + 3s + 2$$

2.b. Análisis de las raíces del sistema

- Si partimos de la **dinámica interna** del sistema:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

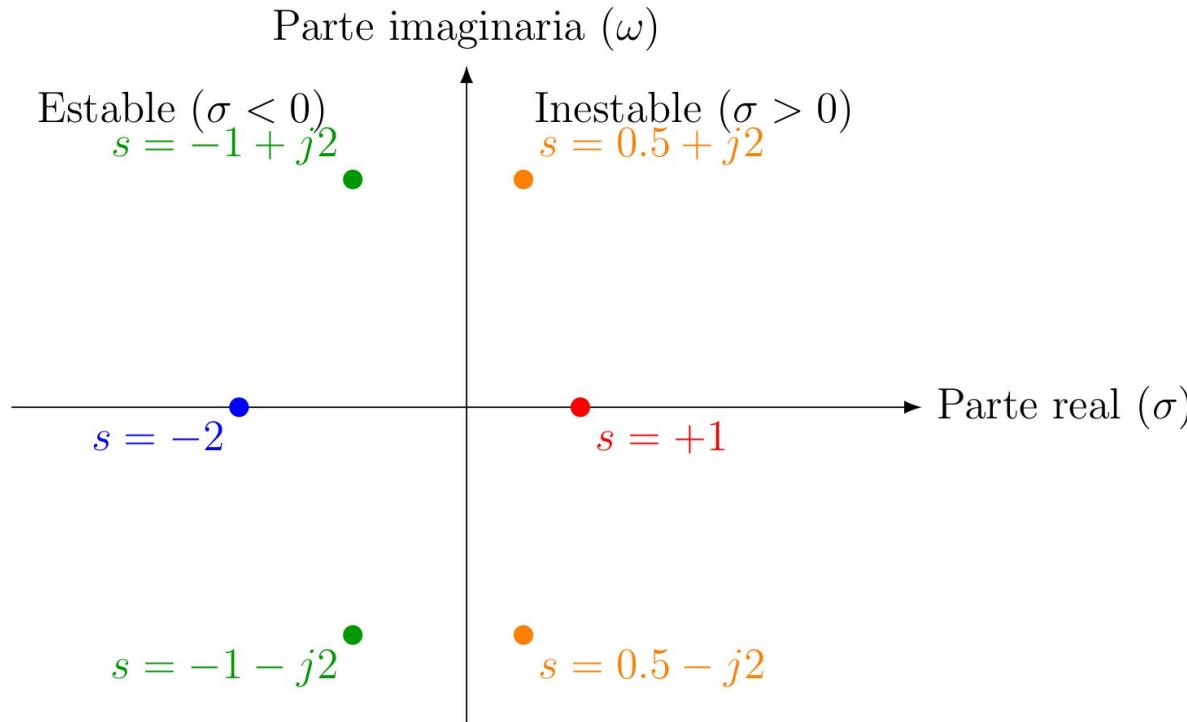
- Aplicando Transformada de Laplace, el polinomio sería:

$$P(s) = \det(sI - A)$$

2.b. Análisis de las raíces del sistema

- La variable s en el dominio de Laplace es una variable compleja (parte real e imaginaria) $\Rightarrow s = \sigma + j\omega$
- Las raíces (autovalores) nos dicen si las soluciones:
 - **decaen** (parte real negativa \rightarrow estable),
 - **crecen** (parte real positiva \rightarrow inestable),
 - **oscilan sin decaer** (parte real cero \rightarrow marginal).

2.b. Análisis de las raíces del sistema



Índice

1. Definición de estabilidad en sistemas dinámicos
2. **Métodos para analizar estabilidad**
 - a. Linealización alrededor de un punto
 - b. Análisis de las raíces del sistema
 - c. **Criterio de Routh-Hurwitz**
 - d. Método de Lyapunov
3. Análisis del comportamiento
 - a. Análisis Monótono u oscilatorio
 - b. Análisis de amortiguamiento
4. Control
 - a. Controlabilidad
 - b. Observabilidad
 - c. Obtener señal de control $u(t)$

2.c. Criterio de Routh-Hurwitz

- Método para analizar la estabilidad de un **sistema lineal**
- Encontrar las raíces para polinomios $n \geq 3$ es complicado

$$P(s) = a_0 s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n$$

2.c. Criterio de Routh-Hurwitz

- Construcción de la tabla de Routh Se organizan los coeficientes del polinomio característico en filas sucesivas:

s^n	a_0	a_2	a_4
s^{n-1}	a_1	a_3	a_5
s^{n-2}	b_1	b_2	b_3
:	:	:	:

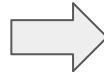
$$b_1 = \frac{a_1 a_2 - a_0 a_3}{a_1}$$

2.c. Criterio de Routh-Hurwitz (Ejemplo)

- Ejemplo masa-resorte:

$$P(s) = s^2 + \frac{b}{m}s + \frac{k}{m}$$

s^2	1	$\frac{k}{m}$
s^1	$\frac{b}{m}$	0
s^0	$\frac{k}{m}$	

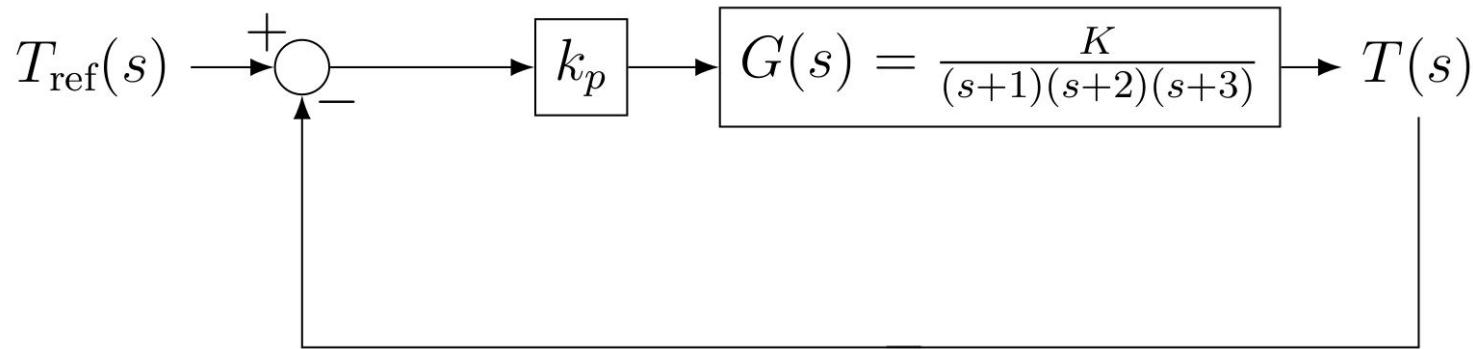


$$1 > 0, \quad \frac{b}{m} > 0, \quad \frac{k}{m} > 0$$

ESTABLE

2.c. Criterio de Routh-Hurwitz (Ejemplo)

- **Ejemplo horno:**



$$T(s) = \frac{K k_p}{(s + 1)(s + 2)(s + 3) + K k_p}$$

2.c. Criterio de Routh-Hurwitz (Ejemplo)

- Ejemplo horno:

$$P(s) = s^3 + 6s^2 + 11s + (6 + Kk_p).$$

s^3	1	11
s^2	6	$6 + Kk_p$
s^1	$\frac{6 \cdot 11 - 1(6 + Kk_p)}{6}$	0
s^0	$6 + Kk_p$	

2.c. Criterio de Routh-Hurwitz (Ejemplo)

Ejemplo horno:

- $1 > 0$ (siempre cierto).
- $6 > 0$ (siempre cierto).
- $\frac{66 - (6 + Kk_p)}{6} > 0 \Rightarrow Kk_p < 60.$
- $6 + Kk_p > 0 \Rightarrow Kk_p > -6$ (siempre cierto si $Kk_p > 0$)

ESTABLE si: $0 < Kk_p < 60$

Índice

1. Definición de estabilidad en sistemas dinámicos
2. **Métodos para analizar estabilidad**
 - a. Linealización alrededor de un punto
 - b. Análisis de las raíces del sistema
 - c. Criterio de Routh-Hurwitz
 - d. Método de Lyapunov**
3. Análisis del comportamiento
 - a. Análisis Monótono u oscilatorio
 - b. Análisis de amortiguamiento
4. Control
 - a. Controlabilidad
 - b. Observabilidad
 - c. Obtener señal de control $u(t)$

2.d. Método de Lyapunov

Definición Sea el sistema no lineal autónomo:

$$\dot{x} = f(x), \quad x \in \mathbb{R}^n.$$

Un equilibrio se encuentra en $x = 0$ (sin pérdida de generalidad, mediante un cambio de coordenadas). Una función $V : \mathbb{R}^n \rightarrow \mathbb{R}$ es candidata de Lyapunov si cumple:

1. $V(x) > 0$ para todo $x \neq 0$ y $V(0) = 0$ (positiva definida).
2. Su derivada temporal a lo largo de trayectorias del sistema satisface:

$$\dot{V}(x) = \nabla V(x) \cdot f(x) \leq 0.$$

Conclusiones

- Si $\dot{V}(x) \leq 0$, el equilibrio es **estable en el sentido de Lyapunov**.
- Si además $\dot{V}(x) < 0$ estrictamente, el equilibrio es **asintóticamente estable**.
- Si $\dot{V}(x)$ puede ser positiva en alguna región, no se garantiza la estabilidad.

2.d. Método de Lyapunov

Interpretación física Este método suele interpretarse como el análisis de la **energía** de un sistema:

- $V(x)$ representa la energía total (cinética + potencial).
- Si la energía nunca aumenta y tiende a disiparse, el sistema debe converger al equilibrio.

Índice

1. Definición de estabilidad en sistemas dinámicos
2. Métodos para analizar estabilidad
 - a. Linealización alrededor de un punto
 - b. Análisis de las raíces del sistema
 - c. Criterio de Routh-Hurwitz
 - d. Método de Lyapunov
- 3. Análisis del comportamiento**
 - a. Análisis Monótono u oscilatorio
 - b. Análisis de amortiguamiento
4. Control
 - a. Controlabilidad
 - b. Observabilidad
 - c. Obtener señal de control $u(t)$

3.a. Análisis Monótono u oscilatorio

a) **Usando autovalores de la linealización** Dado un sistema no lineal, linealizado alrededor de un equilibrio:

$$\delta \dot{x} = A\delta x,$$

los autovalores de la matriz A determinan el comportamiento local:

- Autovalores reales \Rightarrow respuesta monótona (crecimiento o decaimiento exponencial).
- Autovalores complejos conjugados \Rightarrow respuesta oscilatoria (con frecuencia de oscilación dada por la parte imaginaria).

3.a. Análisis Monótono u oscilatorio

b) Usando las raíces del polinomio característico Dada una ecuación diferencial lineal de coeficientes constantes:

$$a_n \frac{d^n y}{dt^n} + \cdots + a_1 \frac{dy}{dt} + a_0 y = 0,$$

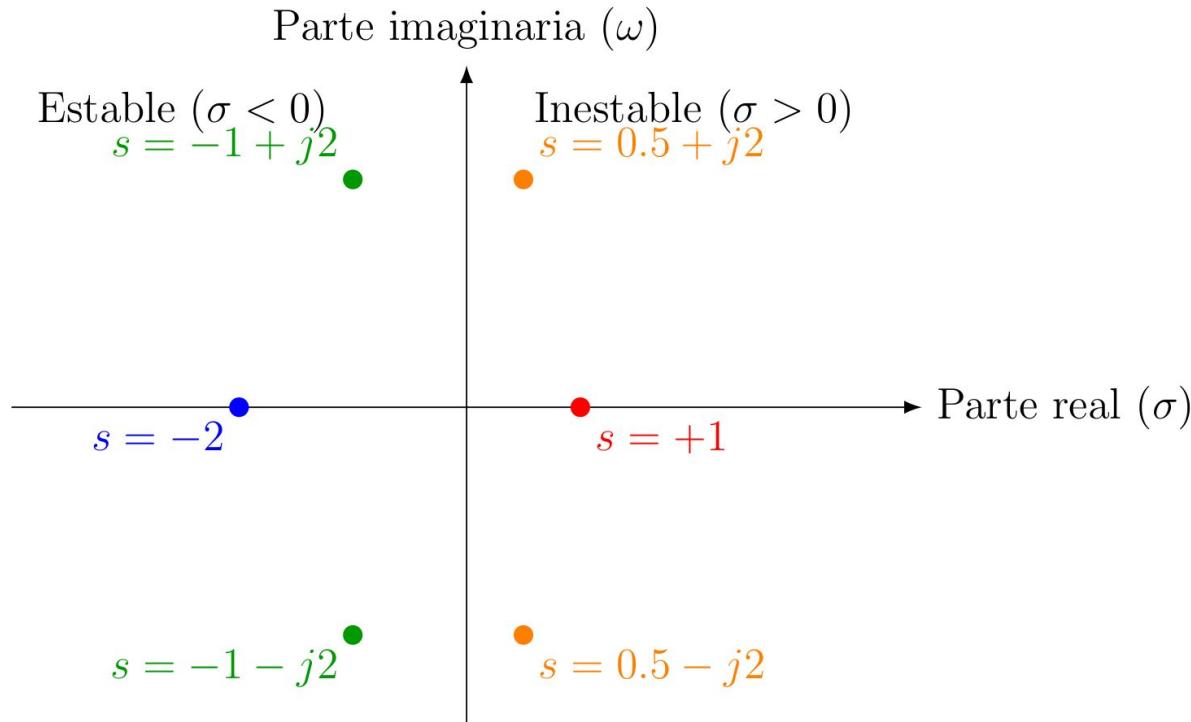
su polinomio característico es:

$$P(s) = a_n s^n + \cdots + a_1 s + a_0.$$

Las raíces de $P(s)$ coinciden con los autovalores de A , y la interpretación es la misma:

- Raíces reales \Rightarrow comportamiento monótono.
- Raíces complejas \Rightarrow comportamiento oscilatorio.

3.a. Análisis Monótono u oscilatorio



3.a. Análisis Monótono u oscilatorio

Ejemplo comparativo: masa–resorte con y sin rozamiento

- Sin rozamiento:

$$m\ddot{x} + kx = 0, \quad P(s) = ms^2 + k.$$

Raíces: $s = \pm j\sqrt{\frac{k}{m}}$. Son puramente imaginarias \Rightarrow oscilaciones persistentes (movimiento armónico simple). El sistema es **marginalmente estable y oscilatorio**.

- Con rozamiento:

$$m\ddot{x} + b\dot{x} + kx = 0, \quad P(s) = ms^2 + bs + k.$$

Según el valor de b , las raíces pueden ser reales o complejas:

- $b^2 > 4mk$: raíces reales negativas \Rightarrow decaimiento monótono.
- $b^2 < 4mk$: raíces complejas conjugadas con parte real negativa \Rightarrow oscilaciones amortiguadas.

Índice

1. Definición de estabilidad en sistemas dinámicos
2. Métodos para analizar estabilidad
 - a. Linealización alrededor de un punto
 - b. Análisis de las raíces del sistema
 - c. Criterio de Routh-Hurwitz
 - d. Método de Lyapunov
- 3. Análisis del comportamiento**
 - a. Análisis Monótono u oscilatorio
 - b. Análisis de amortiguamiento**
4. Control
 - a. Controlabilidad
 - b. Observabilidad
 - c. Obtener señal de control $u(t)$

3.b. Análisis de amortiguamiento

El polinomio característico se escribe como:

$$P(s) = s^2 + 2\zeta\omega_n s + \omega_n^2.$$

Masa-resorte

$$P(s) = s^2 + \frac{b}{m}s + \frac{k}{m}$$

En sistemas de segundo orden, se define la frecuencia natural y el coeficiente de amortiguamiento:

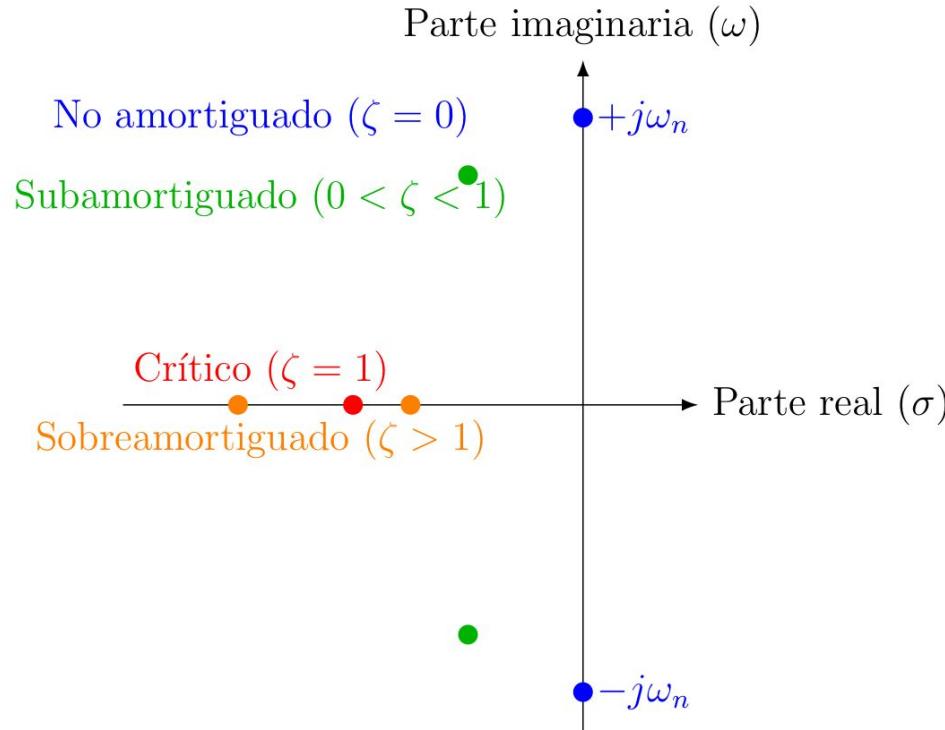
$$\omega_n = \sqrt{\frac{k}{m}}, \quad \zeta = \frac{b}{2\sqrt{mk}}.$$

3.b. Análisis de amortiguamiento

Según el valor de ζ , se distinguen cuatro casos:

- **No amortiguado ($\zeta = 0$):** raíces $s = \pm j\omega_n$. Oscilaciones persistentes (sin pérdida de energía).
- **Subamortiguado ($0 < \zeta < 1$):** raíces $s = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2}$. Oscilaciones amortiguadas (frecuencia menor que ω_n).
- **Amortiguamiento crítico ($\zeta = 1$):** raíz doble $s = -\omega_n$. Decaimiento monótono más rápido posible sin oscilaciones.
- **Sobreamortiguado ($\zeta > 1$):** raíces reales y distintas, ambas negativas. Decaimiento monótono pero más lento que en el caso crítico.

3.b. Análisis de amortiguamiento



Índice

1. Definición de estabilidad en sistemas dinámicos
2. Métodos para analizar estabilidad
 - a. Linealización alrededor de un punto
 - b. Análisis de las raíces del sistema
 - c. Criterio de Routh-Hurwitz
 - d. Método de Lyapunov
3. Análisis del comportamiento
 - a. Análisis Monótono u oscilatorio
 - b. Análisis de amortiguamiento
- 4. Control**
 - a. **Controlabilidad**
 - b. Observabilidad
 - c. Obtener señal de control $u(t)$

4.a. Controlabilidad

Definición general Un sistema dinámico es **controlable** si, partiendo de cualquier estado inicial $x(0)$, es posible encontrar una señal de control $u(t)$ que lo lleve a cualquier estado final x_f en un tiempo finito.

Para un sistema lineal en espacio de estados:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x \in \mathbb{R}^n,$$

la **matriz de controlabilidad** se define como:

$$\mathcal{C} = [B \ AB \ A^2B \ \dots \ A^{n-1}B],$$

donde A es la matriz de dinámica ($n \times n$) y B la matriz de entrada ($n \times m$).

4.a. Controlabilidad

Criterio de Kalman El sistema es **completamente controlable** si y sólo si la matriz \mathcal{C} tiene *rango completo*, es decir:

$$\text{rango}(\mathcal{C}) = n.$$

Interpretación El rango de una matriz indica el número de direcciones linealmente independientes que abarcan sus columnas.

- Si $\text{rango}(\mathcal{C}) = n$, las columnas de \mathcal{C} abarcan todo el espacio de estados \mathbb{R}^n . Esto significa que las entradas, aplicadas en el tiempo, pueden mover al sistema en cualquier dirección del espacio de estados.
- Si $\text{rango}(\mathcal{C}) < n$, existen direcciones del espacio de estados inaccesibles: no se puede forzar al sistema a moverse hacia ellas mediante las entradas. El sistema es entonces **no controlable**.

Índice

1. Definición de estabilidad en sistemas dinámicos
2. Métodos para analizar estabilidad
 - a. Linealización alrededor de un punto
 - b. Análisis de las raíces del sistema
 - c. Criterio de Routh-Hurwitz
 - d. Método de Lyapunov
3. Análisis del comportamiento
 - a. Análisis Monótono u oscilatorio
 - b. Análisis de amortiguamiento
- 4. Control**
 - a. Controlabilidad
 - b. Observabilidad**
 - c. Obtener señal de control $u(t)$

4.b. Observabilidad

Definición general

Un sistema es **observable** si, partiendo de la salida $y(t)$ (junto con la entrada $u(t)$ conocida), es posible determinar el estado inicial $x(0)$ en un tiempo finito. En otras palabras, no hay "información oculta": las mediciones contienen toda la información necesaria para reconstruir el vector de estados.

4.b. Observabilidad

Consideremos un sistema lineal invariante en el tiempo (LTI):

$$\dot{x} = Ax + Bu, \quad y = Cx,$$

donde $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$.

El **criterio de Kalman** establece que el sistema es observable si y sólo si la *matriz de observabilidad*:

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

tiene *rango completo*, es decir:

$$\text{rango}(\mathcal{O}) = n.$$

4.b. Observabilidad

Interpretación

- Las filas de \mathcal{O} representan cómo las salidas y sus derivadas dependen de los estados internos.
- Si el rango es n , significa que la información contenida en las salidas y sus derivadas abarca todo el espacio de estados. Por tanto, es posible reconstruir $x(t)$ a partir de $y(t)$.
- Si el rango es menor que n , existen componentes del estado que **no afectan a la salida**, por lo que son invisibles al observador. El sistema entonces es **no observable**.

Índice

1. Definición de estabilidad en sistemas dinámicos
2. Métodos para analizar estabilidad
 - a. Linealización alrededor de un punto
 - b. Análisis de las raíces del sistema
 - c. Criterio de Routh-Hurwitz
 - d. Método de Lyapunov
3. Análisis del comportamiento
 - a. Análisis Monótono u oscilatorio
 - b. Análisis de amortiguamiento
- 4. Control**
 - a. Controlabilidad
 - b. Observabilidad
 - c. Obtener señal de control $u(t)$**

4.c. Obtener señal de control $u(t)$

Definición general

En secciones anteriores vimos cómo analizar estabilidad y controlabilidad. El siguiente paso en diseño de control consiste en determinar **qué señal de entrada $u(t)$ aplicar** para llevar el sistema desde un estado inicial $x(0)$ hasta un estado deseado x_d .

4.c. Obtener señal de control $u(t)$

Dado un sistema lineal:

$$\dot{x}(t) = Ax(t) + Bu(t),$$

con estado deseado x_d , existen distintas estrategias de control:

- **Realimentación de estado:**

$$u(t) = -K(x(t) - x_d),$$

donde K es una matriz de ganancias elegida de modo que $A - BK$ tenga autovalores en la región deseada (por ejemplo, con parte real negativa). Este método estabiliza el error $e(t) = x(t) - x_d$.

4.c. Obtener señal de control $u(t)$

- **Control estacionario:** si se quiere que x_d sea un equilibrio, se puede resolver

$$0 = Ax_d + Bu_d,$$

para encontrar la entrada constante u_d que mantiene al sistema en x_d .

- **Óptimo (LQR, MPC, etc.):** en problemas más complejos, se plantea un costo y se obtiene $u(t)$ optimizando una función objetivo.

4.c. Obtener señal de control $u(t)$: Ejemplo

El modelo cinemático del robot diferencial es:

$$\dot{x} = v \cos \theta, \quad \dot{y} = v \sin \theta, \quad \dot{\theta} = \omega,$$

donde $u(t) = (v, \omega)$ son las entradas.

Estado deseado Queremos llevar al robot desde (x, y, θ) hasta un estado deseado (x_d, y_d, θ_d) .

4.c. Obtener señal de control $u(t)$: Ejemplo

Control de estabilización al punto Una ley de control común es:

$$\rho = \sqrt{(x_d - x)^2 + (y_d - y)^2}, \quad \alpha = \arctan 2(y_d - y, x_d - x) - \theta, \quad \beta = -\theta - \alpha + \theta_d.$$

El control se diseña como:

$$v = k_\rho \rho, \quad \omega = k_\alpha \alpha + k_\beta \beta,$$

con ganancias $k_\rho, k_\alpha, k_\beta > 0$ elegidas adecuadamente.

Interpretación - v avanza proporcionalmente a la distancia ρ al objetivo. - ω corrige la orientación del robot hacia el punto deseado y asegura que finalice con orientación θ_d . - Con este control, el robot es capaz de alcanzar cualquier configuración deseada (x_d, y_d, θ_d) desde cualquier posición inicial.

En el siguiente tema ...

- Paradigmas de simulación
 - Nos centraremos en simulación de sistemas discretos.

TEMA 2: Estabilidad, Controlabilidad y Observabilidad

MODELOS COMPUTACIONALES Y SIMULACIÓN DE SISTEMAS
Curso 2025-2026 - Semanas 4 y 5



UNIVERSITAT D'ALACANT
UNIVERSIDAD DE ALICANTE
Escola Politècnica Superior
Escuela Politécnica Superior



TEMA 3: Paradigmas de simulación

MODELOS COMPUTACIONALES Y SIMULACIÓN DE SISTEMAS
Curso 2025-2026



Índice

1. Introducción
2. Simulación continua basada en ecuaciones diferenciales
3. Simulación basada en eventos discretos (**DES**, Discrete Event Simulation)
 - a. Modelo **M/M/c** y la fórmula de **Erlang C**
4. Simulación mediante el método Monte Carlo (**MC**)
 - a. Estimación de Pi
 - b. Propagación de incertidumbre
5. Simulación multiagente y basada en redes
6. Software de simulación

Índice

- 1. Introducción**
2. Simulación continua basada en ecuaciones diferenciales
3. Simulación basada en eventos discretos (**DES**, Discrete Event Simulation)
 - a. Modelo **M/M/c** y la fórmula de **Erlang C**
4. Simulación mediante el método Monte Carlo (**MC**)
 - a. Estimación de Pi
 - b. Propagación de incertidumbre
5. Simulación multiagente y basada en redes
6. Software de simulación

1. Introducción

- El estudio de los **paradigmas de simulación** constituye uno de los pilares fundamentales en la modelización computacional de sistemas.
- Un paradigma de simulación es un enfoque o marco conceptual que define **cómo se representan, procesan y analizan los sistemas reales** dentro de un entorno computacional.
- En esencia, cada paradigma define una **forma diferente de concebir el tiempo, los eventos, las interacciones entre componentes y la evolución del sistema**.

1. Introducción

Existen múltiples enfoques según la **naturaleza del sistema** que se desea simular: sistemas discretos, continuos, estocásticos o híbridos:

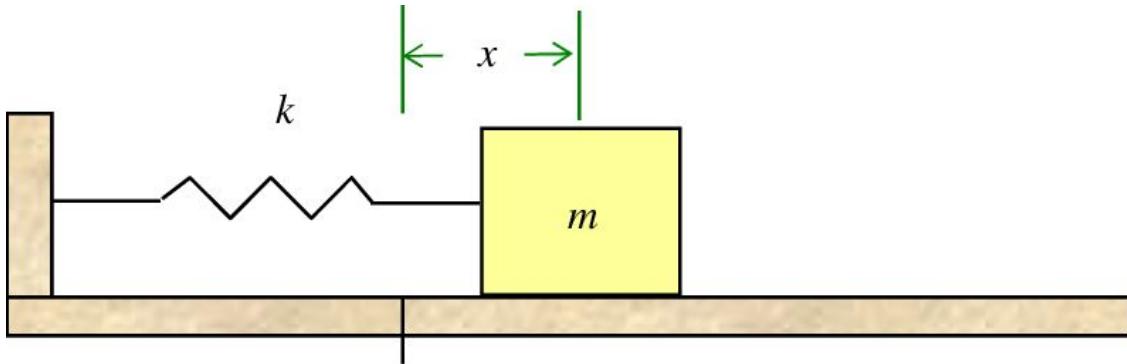
- Simulación continua basada en ecuaciones diferenciales.
- Simulación basada en eventos discretos (**DES**, Discrete Event Simulation).
- Simulación mediante el método Monte Carlo (**MC**).
- Simulación multiagente y basada en redes.

Índice

1. Introducción
2. **Simulación continua basada en ecuaciones diferenciales**
3. Simulación basada en eventos discretos (**DES**, Discrete Event Simulation)
 - a. Modelo **M/M/c** y la fórmula de **Erlang C**
4. Simulación mediante el método Monte Carlo (**MC**)
 - a. Estimación de Pi
 - b. Propagación de incertidumbre
5. Simulación multiagente y basada en redes
6. Software de simulación

2. Simulación continua basada en EDO

- Solución numérica (método de euler): $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot f(\mathbf{x}_k)$



Índice

1. Introducción
2. Simulación continua basada en ecuaciones diferenciales
3. **Simulación basada en eventos discretos (DES, Discrete Event Simulation)**
 - a. Modelo M/M/c y la fórmula de Erlang C
4. Simulación mediante el método Monte Carlo (**MC**)
 - a. Estimación de Pi
 - b. Propagación de incertidumbre
5. Simulación multiagente y basada en redes
6. Software de simulación

3. Simulación basada en eventos discretos

- La simulación basada en eventos discretos (DES) es un enfoque en el cual el tiempo avanza de un evento a otro.
 - El estado del sistema solo cambia en los instantes en los que ocurre un evento.
 - Entre un evento y otro, el sistema permanece en un estado constante.
- Ejemplos típicos incluyen:
 - Sistemas de colas en bancos o aeropuertos, procesos de fabricación, redes de comunicación o sistemas de atención al cliente.

3. Simulación basada en eventos discretos

Un modelo típico de simulación discreta se compone de los siguientes elementos:

- **Reloj de simulación:** variable que representa el tiempo actual de la simulación. No avanza de manera continua, sino que salta de un evento a otro.
- **Lista de eventos futuros (LEF):** estructura que contiene los eventos programados junto con el tiempo en que deben ocurrir. La simulación siempre ejecuta el evento más próximo en el tiempo.
- **Eventos:** sucesos que causan un cambio de estado en el sistema. Ejemplos: llegada de un cliente, inicio o fin de servicio, avería de una máquina.
- **Entidades:** elementos que fluyen a través del sistema (clientes, productos, mensajes, etc.).
- **Recursos:** elementos que atienden o procesan las entidades (empleados, servidores, máquinas...).
- **Colas:** estructuras que almacenan las entidades que esperan ser atendidas.

3. Simulación basada en eventos discretos

Formalización del avance temporal

Si designamos t_e como el tiempo del próximo evento, el reloj de simulación avanza de t a t_e y el estado del sistema $S(t)$ se actualiza mediante una función de transición discreta:

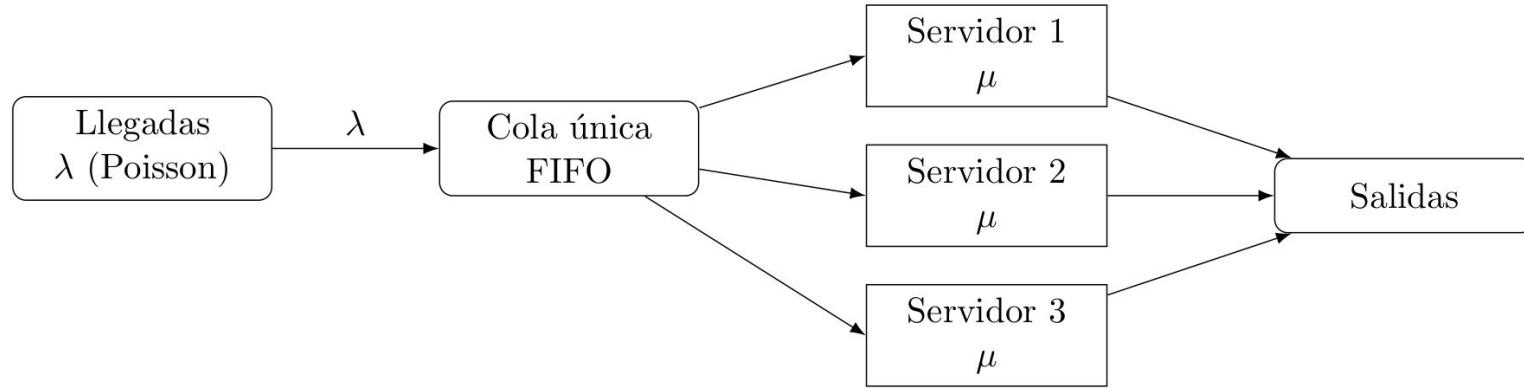
$$S(t_e^+) = f(S(t_e^-), e),$$

donde $S(t_e^-)$ es el estado inmediatamente antes del evento, $S(t_e^+)$ el estado después del evento y e representa el tipo de evento ocurrido.

Índice

1. Introducción
2. Simulación continua basada en ecuaciones diferenciales
3. **Simulación basada en eventos discretos (DES, Discrete Event Simulation)**
 - a. **Modelo M/M/c y la fórmula de Erlang C**
4. Simulación mediante el método Monte Carlo (**MC**)
 - a. Estimación de Pi
 - b. Propagación de incertidumbre
5. Simulación multiagente y basada en redes
6. Software de simulación

3.a. Modelo M/M/c y la fórmula de Erlang C



Erlang C (modelo analítico)

$$P_0 = \left[\sum_{n=0}^{c-1} \frac{a^n}{n!} + \frac{a^c}{c!(1-\rho)} \right]^{-1}$$

$$P_w = \frac{a^c}{c!(1-\rho)} P_0, \quad W_q = \frac{P_w}{c\mu - \lambda}$$

$$W = W_q + \frac{1}{\mu}, \quad L_q = \lambda W_q, \quad L = \lambda W$$

3.a. Modelo M/M/c y la fórmula de Erlang C

El modelo **M/M/c** es uno de los más clásicos en la teoría de colas y sirve como referencia para comparar resultados de simulación con soluciones analíticas. Se trata de un sistema con:

- Llegadas al sistema que siguen un proceso de Poisson con tasa λ (tiempos entre llegadas distribuidos exponencialmente).
- Tiempos de servicio también exponenciales con tasa μ .
- c servidores en paralelo que atienden una cola común con disciplina *FIFO* (First In, First Out).
- Capacidad ilimitada y población infinita de clientes.

3.a. Modelo M/M/c y la fórmula de Erlang C

Relación entre la tasa de llegadas λ y el tiempo medio entre llegadas. En un modelo de colas tipo $M/M/c$, las llegadas siguen un proceso de Poisson con tasa λ (llegadas por unidad de tiempo).

Los *tiempos entre llegadas* siguen entonces una distribución exponencial con media:

$$\mathbb{E}[T_{\text{entre}}] = \frac{1}{\lambda}.$$

Por lo tanto, si en la simulación se define el parámetro `mean_interarrival` como el tiempo medio entre llegadas, la tasa de llegadas utilizada en las fórmulas analíticas debe calcularse como:

$$\lambda = \frac{1}{\text{mean_interarrival}}.$$

Por ejemplo, si en promedio llega un cliente cada 1.8 minutos, entonces:

$$\lambda = \frac{1}{1.8} \approx 0.5556 \text{ llegadas/minuto.}$$

3.a. Modelo M/M/c y la fórmula de Erlang C

El comportamiento del sistema depende de la **intensidad de tráfico**:

$$\rho = \frac{\lambda}{c \mu}$$

que representa el grado de utilización promedio de los servidores. Para que el sistema sea estable, debe cumplirse $\rho < 1$.

3.a. Modelo M/M/c y la fórmula de Erlang C

Fórmula de Erlang C. La *fórmula de Erlang C* calcula la probabilidad de que un cliente tenga que esperar antes de ser atendido, es decir, que todos los servidores estén ocupados cuando llega un nuevo cliente. Primero se define la probabilidad de que el sistema esté vacío:

$$P_0 = \left[\sum_{n=0}^{c-1} \frac{a^n}{n!} + \frac{a^c}{c! (1 - \rho)} \right]^{-1}, \quad \text{donde } a = \frac{\lambda}{\mu}.$$

A partir de P_0 , la probabilidad de que un cliente deba esperar en la cola (todos los servidores ocupados) es:

$$P_w = \frac{a^c}{c! (1 - \rho)} P_0.$$

3.a. Modelo M/M/c y la fórmula de Erlang C

La espera media en cola (W_q) y en el sistema completo (W) se calculan mediante:

$$W_q = \frac{P_w}{c\mu - \lambda}, \quad W = W_q + \frac{1}{\mu}.$$

Finalmente, los números medios de clientes en cola y en el sistema son:

$$L_q = \lambda W_q, \quad L = \lambda W.$$

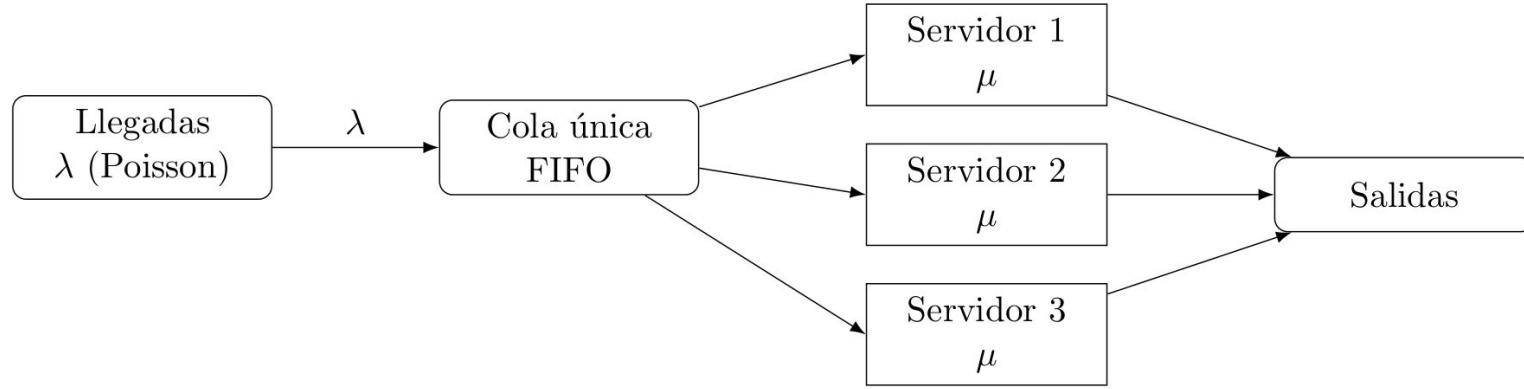
3.a. Modelo M/M/c y la fórmula de Erlang C

Interpretación:

- W_q mide el tiempo medio que un cliente pasa esperando antes de ser atendido.
- W incluye tanto el tiempo de espera como el tiempo de servicio.
- L_q y L son las longitudes medias de la cola y del sistema, respectivamente.
- La fórmula de Erlang C permite **predecir el rendimiento del sistema** sin necesidad de simulación, sirviendo como *modelo analítico de referencia*.

En la práctica, se utiliza para comparar los resultados de simulaciones de colas (por ejemplo, mediante SimPy o AnyLogic) con las métricas teóricas derivadas del modelo M/M/c.

3.a. Modelo M/M/c y la fórmula de Erlang C



Erlang C (modelo analítico)

$$P_0 = \left[\sum_{n=0}^{c-1} \frac{a^n}{n!} + \frac{a^c}{c!(1-\rho)} \right]^{-1}$$

$$P_w = \frac{a^c}{c!(1-\rho)} P_0, \quad W_q = \frac{P_w}{c\mu - \lambda}$$

$$W = W_q + \frac{1}{\mu}, \quad L_q = \lambda W_q, \quad L = \lambda W$$

3.a. Modelo M/M/c y la fórmula de Erlang C

- Para eventos discretos:

<https://simpy.readthedocs.io/en/latest/>



SimPy es una biblioteca de Python para *simulación basada en eventos discretos* (DES). Su idea central es modelar **procesos** como *corutinas* (funciones generadoras) que van *cediendo* el control al motor de simulación mediante **yield** de **eventos** como **timeout** (paso del tiempo), **request/release** (uso de recursos), o señales de sincronización (**Event**, **AllOf**, **AnyOf**). El **reloj de simulación** avanza al tiempo del *próximo evento*.

3.a. Modelo M/M/c y la fórmula de Erlang C

Métricas principales. Entre las medidas de desempeño más habituales se encuentran:

- **Utilización de los servidores:** proporción de tiempo en que los servidores están ocupados, $\rho = \frac{\lambda}{c\mu}$.
- **Tiempo medio en cola:** W_q , tiempo promedio que los clientes esperan antes de ser atendidos.
- **Tiempo total en el sistema:** $W = W_q + \frac{1}{\mu}$.
- **Número medio en cola:** $L_q = \lambda W_q$.
- **Número medio en el sistema:** $L = \lambda W$.

Índice

1. Introducción
2. Simulación continua basada en ecuaciones diferenciales
3. Simulación basada en eventos discretos (**DES**, Discrete Event Simulation)
 - a. Modelo **M/M/c** y la fórmula de **Erlang C**
- 4. Simulación mediante el método Monte Carlo (MC)**
 - a. Estimación de Pi
 - b. Propagación de incertidumbre
5. Simulación multiagente y basada en redes
6. Software de simulación

4. Simulación con el método Monte Carlo

- La simulación Monte Carlo (**MC**) es un paradigma de simulación basado en la generación de **números aleatorios** para modelar fenómenos inciertos o complejos.
 - El nombre proviene del famoso casino de Monte Carlo, debido al papel central del azar en el método.
- En esencia, consiste en **repetir muchas veces un experimento aleatorio** para aproximar un valor esperado, una probabilidad, o una integral difícil de resolver analíticamente.

4. Simulación con el método Monte Carlo

Idea básica. Si queremos estimar un valor esperado de una función $f(X)$ de una variable aleatoria X , se puede aproximar por el promedio muestral de N simulaciones independientes:

$$\mathbb{E}[f(X)] \approx \frac{1}{N} \sum_{i=1}^N f(X_i),$$

donde X_i son valores generados al azar según la distribución de X .

Cuando N es grande, por el *teorema de los grandes números*, la media muestral converge al valor esperado real.

4. Simulación con el método Monte Carlo

Ventajas del método Monte Carlo.

- Permite resolver problemas sin solución analítica cerrada.
- Es flexible y aplicable a cualquier tipo de distribución o sistema estocástico.
- La precisión mejora con el número de simulaciones (error $\sim 1/\sqrt{N}$).
- Es muy útil en física, finanzas, fiabilidad, optimización y estadística.

4. Simulación con el método Monte Carlo

Limitaciones.

- Requiere muchas simulaciones para obtener alta precisión.
- No siempre ofrece información determinista: sólo estimaciones probabilísticas.
- Los resultados dependen de la calidad del generador de números aleatorios.

Índice

1. Introducción
2. Simulación continua basada en ecuaciones diferenciales
3. Simulación basada en eventos discretos (**DES**, Discrete Event Simulation)
 - a. Modelo **M/M/c** y la fórmula de **Erlang C**
4. **Simulación mediante el método Monte Carlo (MC)**
 - a. **Estimación de Pi**
 - b. Propagación de incertidumbre
5. Simulación multiagente y basada en redes
6. Software de simulación

4.a. Estimación de Pi

Uno de los ejemplos clásicos de Monte Carlo es la estimación de π mediante muestreo aleatorio.

Se considera un cuadrado de lado 2 que encierra un círculo de radio 1. Si se generan puntos aleatorios (x, y) uniformemente distribuidos en el cuadrado, la probabilidad de que un punto caiga dentro del círculo es igual al cociente de sus áreas:

$$P = \frac{\text{Área del círculo}}{\text{Área del cuadrado}} = \frac{\pi r^2}{(2r)^2} = \frac{\pi}{4}.$$

Entonces:

$$\pi \approx 4 \times \frac{n_{\text{dentro}}}{N},$$

donde n_{dentro} es el número de puntos que satisfacen $x^2 + y^2 \leq 1$.

4.a. Estimación de Pi

Ejemplo numérico. Supongamos que se generan $N = 100,000$ puntos aleatorios y 78,520 caen dentro del círculo:

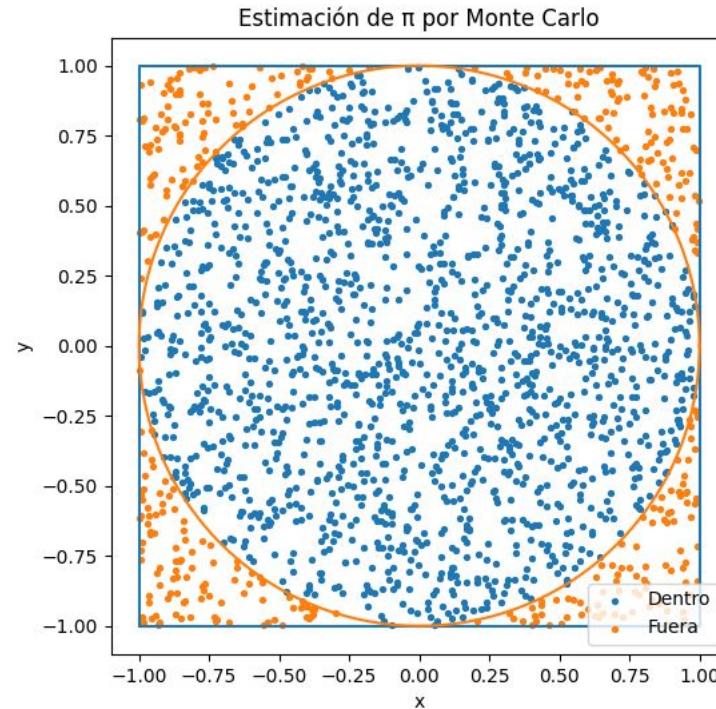
$$\pi \approx 4 \times \frac{78,520}{100,000} = 3.1408.$$

Este valor ya es una buena aproximación de $\pi \approx 3.1416$.

A medida que se incrementa N , el error medio cuadrático disminuye aproximadamente como $1/\sqrt{N}$.

$$\text{Error} \propto \frac{1}{\sqrt{N}}.$$

4.a. Estimación de Pi



Índice

1. Introducción
2. Simulación continua basada en ecuaciones diferenciales
3. Simulación basada en eventos discretos (**DES**, Discrete Event Simulation)
 - a. Modelo **M/M/c** y la fórmula de **Erlang C**
- 4. Simulación mediante el método Monte Carlo (MC)**
 - a. Estimación de Pi
 - b. Propagación de incertidumbre**
5. Simulación multiagente y basada en redes
6. Software de simulación

4.b. Propagación de incertidumbre

- Modelo cinemático robot móvil:

$$x_{k+1} = x_k + v_k \cos(\theta_k) \Delta t,$$

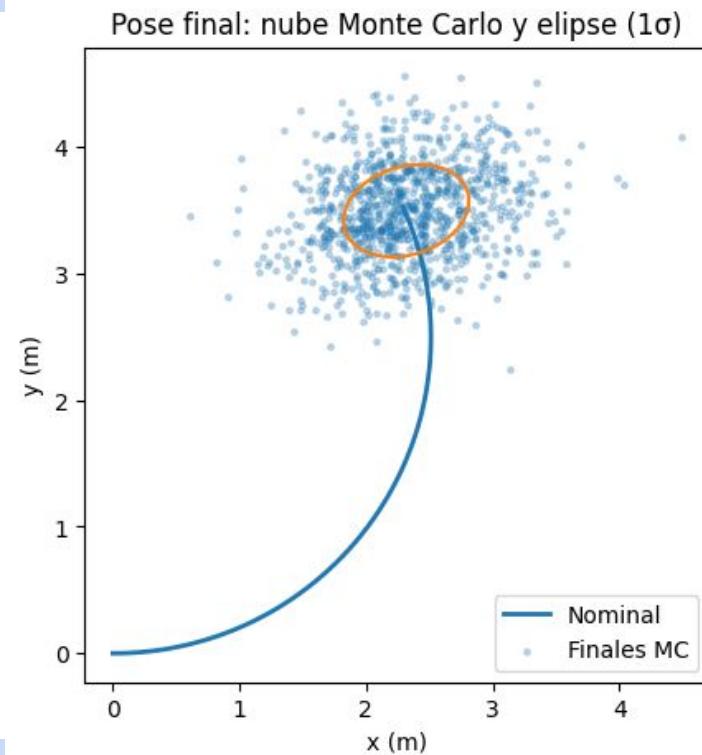
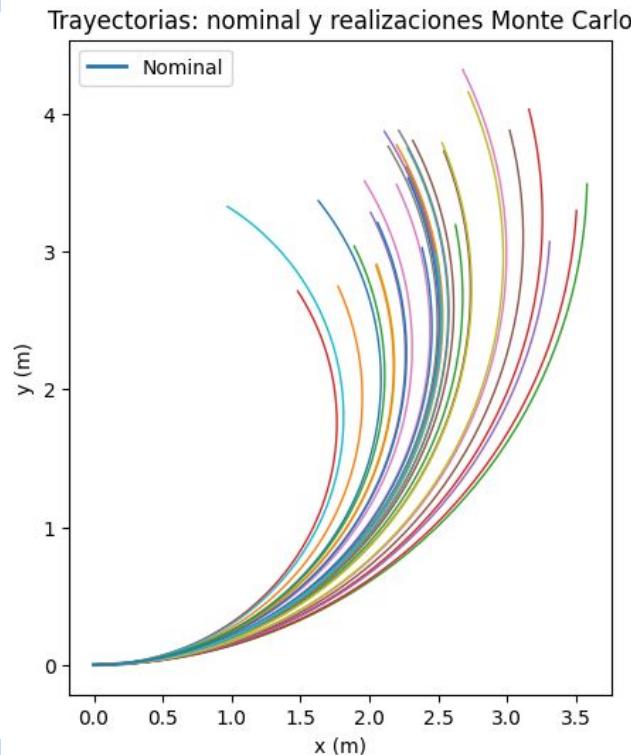
$$y_{k+1} = y_k + v_k \sin(\theta_k) \Delta t,$$

$$\theta_{k+1} = \theta_k + \omega_k \Delta t.$$

- Con incertidumbre modelada como:

$$v_k \sim \mathcal{N}(\bar{v}, \sigma_v^2), \quad \omega_k \sim \mathcal{N}(\bar{\omega}, \sigma_\omega^2).$$

4.b. Propagación de incertidumbre



Índice

1. Introducción
2. Simulación continua basada en ecuaciones diferenciales
3. Simulación basada en eventos discretos (**DES**, Discrete Event Simulation)
 - a. Modelo **M/M/c** y la fórmula de **Erlang C**
4. Simulación mediante el método Monte Carlo (**MC**)
 - a. Estimación de Pi
 - b. Propagación de incertidumbre
- 5. Simulación multiagente y basada en redes**
6. Software de simulación

5. Simulación multiagente y basada en redes



UNIVERSITAT D'ALACANT
UNIVERSIDAD DE ALICANTE
Escola Politècnica Superior
Escuela Politécnica Superior

Estructura básica de un modelo multiagente

- Conjunto de agentes $A = \{a_1, a_2, \dots, a_N\}$.
- Entorno E en el que se mueven o interactúan.
- Funciones de percepción, decisión y acción: $a_i : E \rightarrow A_i$.
- Reglas de interacción entre agentes y entorno.

Índice

1. Introducción
2. Simulación continua basada en ecuaciones diferenciales
3. Simulación basada en eventos discretos (**DES**, Discrete Event Simulation)
 - a. Modelo **M/M/c** y la fórmula de **Erlang C**
4. Simulación mediante el método Monte Carlo (**MC**)
 - a. Estimación de Pi
 - b. Propagación de incertidumbre
5. Simulación multiagente y basada en redes
- 6. Software de simulación**

6. Software de simulación

Entorno	Paradigma principal
Simulink (MATLAB)	Continuo / híbrido
AnyLogic	Multi-paradigma (eventos, agentes, continuo)
NetLogo	Basado en agentes
Arena / SimEvents	Eventos discretos

6. Software de simulación

Paradigma

Eventos discretos

Continuo / ecuaciones diferenciales

Monte Carlo

Basado en agentes

Basado en redes

Bibliotecas recomendadas

`simpy`

`scipy.integrate`, `numpy`, `matplotlib`

`numpy`, `random`, `pandas`

`mesa`, `networkx`

`networkx`, `igraph`

TEMA 3: Paradigmas de simulación

MODELOS COMPUTACIONALES Y SIMULACIÓN DE SISTEMAS
Curso 2025-2026



Tema 4: Lenguajes formales para modelos conceptuales

MODELOS COMPUTACIONALES Y SIMULACIÓN DE SISTEMAS
Curso 2025-2026



UNIVERSITAT D'ALACANT
UNIVERSIDAD DE ALICANTE
Escola Politècnica Superior
Escuela Politécnica Superior



Temario

B1 - Sistemas Dinámicos

T1: Fundamentos del modelado de sistemas dinámicos

T2: Estabilidad, controlabilidad y observabilidad

T3: Paradigmas de simulación

B2 - Sistemas Complejos

T4: Lenguajes formales para modelos conceptuales

T5: Redes complejas y modelado estructural

B3 - Modelado con IA

T6: Identificación de sistemas

T7: Inteligencia artificial aplicada al modelado de sistemas

Índice

1. Introducción
2. UML para modelado de sistemas
3. SysML para modelado de sistemas físicos y ciber-físicos
4. Formalismo DEVS
5. Modelado basado en agentes

Índice

- 1. Introducción**
2. UML para modelado de sistemas
3. SysML para modelado de sistemas físicos y ciber-físicos
4. Formalismo DEVS
5. Modelado basado en agentes

1. Introducción

- El **modelado de sistemas** complejos requiere herramientas que permitan expresar la estructura y el comportamiento de los sistemas de manera precisa, **sin ambigüedades**.
- Los lenguajes formales cumplen esta función, al **proporcionar una sintaxis y semántica** bien definidas que posibilitan la comunicación rigurosa entre l@s diseñador@s, ingenier@s y simuladores.
- Estos lenguajes se utilizan para describir la **estructura, el comportamiento y las restricciones** de un sistema.
- Entre los más utilizados se encuentran **UML, SysML y DEVS**, además de los lenguajes específicos de dominio (**DSLs**) y los enfoques de modelado basado en agentes.

1. Introducción

Un lenguaje formal permite definir de manera estructurada:

- Elementos de modelado: entidades, clases, bloques, agentes o componentes.
 - Relaciones: asociación, composición, flujo o dependencia.
 - Restricciones: condiciones algebraicas o lógicas sobre el sistema.
- El uso de diagramas facilita la visualización de las diferentes **vistas** de un sistema: estructural, de comportamiento, de requisitos y física.

Índice

1. Introducción
2. **UML para modelado de sistemas**
3. SysML para modelado de sistemas físicos y ciber-físicos
4. Formalismo DEVS
5. Modelado basado en agentes

2. UML para modelado de sistemas

UML (**Unified Modeling Language**) es un lenguaje gráfico semi-formal. Tiene una sintaxis bastante rígida (cuadros, flechas, tipos de relaciones ...) pero su semántica puede adaptarse al dominio: software, colas, robots, etc.

- **Describir la estructura** del sistema (qué entidades hay y cómo se relacionan).
- **Describir el comportamiento** (cómo cambia con el tiempo, qué eventos hay).
- Servir como puente entre el **modelo conceptual** y la **implementación / simulación**.



<https://www.omg.org/uml/>

2. UML para modelado de sistemas

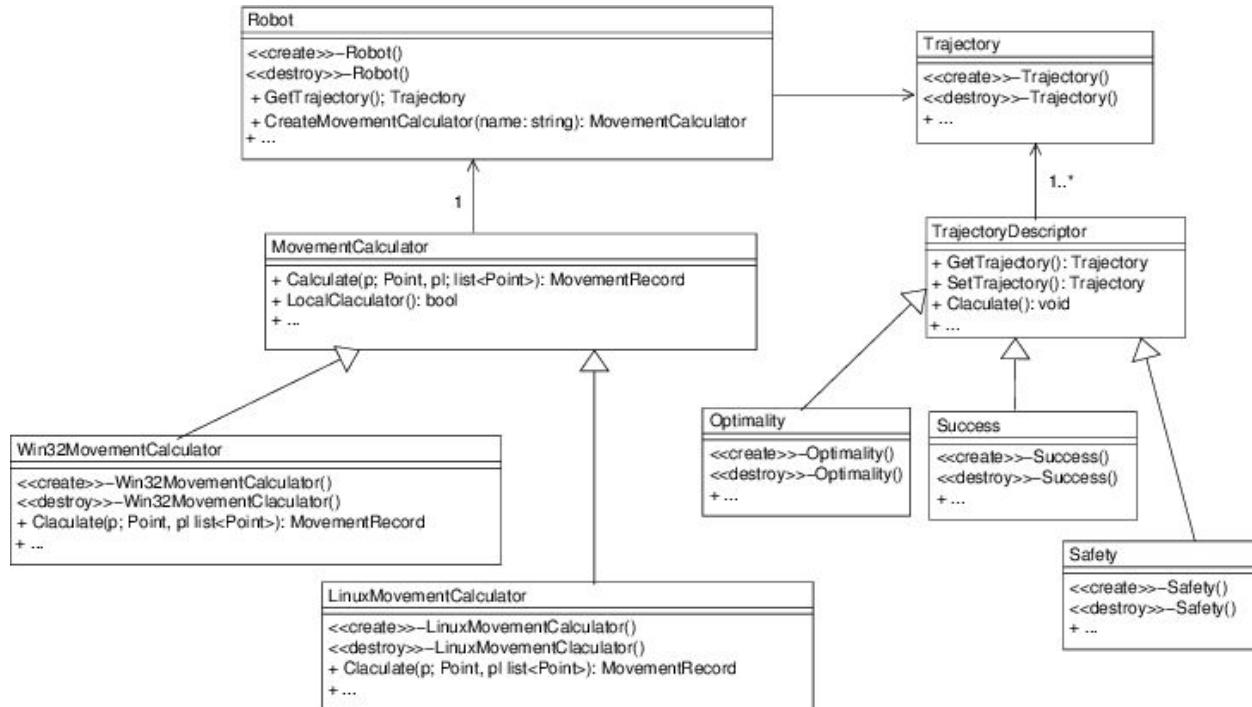
- **Clases** (rectángulos):
 - Nombre de la clase (parte superior).
 - Atributos (parte central).
 - Operaciones/métodos (parte inferior).
- **Atributos:** nombre : Tipo
 - Opcionalmente, visibilidad: + público, - privado, # protegido.
- **Operaciones:** operación(parámetros) : TipoRetorno

2. UML para modelado de sistemas

Relaciones:

- **Asociación:** línea simple entre clases. Puede tener:
 - *Multiplicidad* ($1, 0..1, *, 1..*$) en los extremos.
 - *Rol* (nombre del papel: **cliente**, **servidor**, etc.).
- **Agregación:** rombo blanco → “tiene-un” pero débil.
- **Composición:** rombo negro → “forma-parte-de” fuerte.
- **Herencia (generalización):** triángulo blanco hacia la superclase.
- **Dependencia:** línea discontinua con flecha, indica uso de una clase por otra.

2. UML para modelado de sistemas



Índice

1. Introducción
2. UML para modelado de sistemas
- 3. SysML para modelado de sistemas físicos y ciber-físicos**
4. Formalismo DEVS
5. Modelado basado en agentes

3. SysML para modelado de sistemas físicos

El lenguaje SysML (**S**ystems **M**odeling **L**anguage) es una extensión y especialización de UML diseñada para la **ingeniería de sistemas**. Su propósito es integrar en un mismo modelo los aspectos estructurales, funcionales, físicos y de requisitos de sistemas que combinan **hardware, software, control, comunicación y procesos humanos**.

Mientras que UML está orientado a la ingeniería de software, SysML amplía sus capacidades para representar:

- Relaciones entre subsistemas mecánicos, eléctricos, electrónicos y de control.
- Restricciones físicas y ecuaciones de comportamiento dinámico.
- Requisitos del sistema y su trazabilidad hacia componentes concretos.
- Integración de hardware y software dentro de un mismo modelo conceptual.



3. SysML para modelado de sistemas físicos

Los modelos SysML son muy utilizados en el diseño de:

- **Sistemas ciber–físicos**, como robots, vehículos autónomos, drones, etc.
- **Sistemas embebidos** con sensores, actuadores y software de control.
- **Sistemas industriales** (plantas de producción, líneas automatizadas).
- **Sistemas aeroespaciales**

3. SysML para modelado de sistemas físicos

Elementos y tipos de diagramas en SysML:

SysML conserva la estructura modular de UML, pero introduce nuevos tipos de diagramas que permiten una descripción multidisciplinar del sistema. Los diagramas más relevantes para la simulación de sistemas son los siguientes:

1. Block Definition Diagram (BDD)
2. Internal Block Diagram (IBD)
3. Requirement Diagram
4. Parametric Diagram
5. Activity y State Machine Diagrams

3. SysML para modelado de sistemas físicos

1. **BDD:** bloques Robot, Controlador, Sensor, Motor, Batería, Entorno.
2. **IBD:** conexiones internas entre sensores, controlador y motores.
3. **Requirement diagram:** requisitos R1–R3 (velocidad, detección, autonomía).
4. **Parametric diagram:** ecuaciones dinámicas para calcular movimiento y consumo de energía.
5. **State Machine:** comportamiento del robot (parado, avanzando, evitando).

3. SysML para modelado de sistemas físicos

Ventajas de SysML

- Permite modelar **estructuras, requisitos y comportamiento** en un mismo lenguaje.
- Facilita la integración de diferentes disciplinas (mecánica, eléctrica, software, control).
- Es compatible con herramientas de simulación (por ejemplo, exportación a Modelica).
- Favorece la **trazabilidad**: los requisitos se relacionan directamente con los bloques del diseño.

En resumen, SysML constituye un lenguaje unificado que conecta el **modelo conceptual**, las **restricciones físicas** y el **comportamiento dinámico**, facilitando la simulación integral de sistemas ciber–físicos complejos.

Índice

1. Introducción
2. UML para modelado de sistemas
3. SysML para modelado de sistemas físicos y ciber-físicos
- 4. Formalismo DEVS**
5. Modelado basado en agentes

4. Formalismo DEVS

El formalismo DEVS (**Discrete Event System Specification**) es un marco matemático para modelar y simular **sistemas de eventos discretos**. Proporciona una definición precisa de los componentes del sistema (modelos atómicos y acoplados) y de su comportamiento en el tiempo a través de **transiciones de estado** provocadas por eventos.

La gran ventaja de DEVS es que:

- Separa claramente la **especificación del modelo** de la **mecánica de la simulación**.
- Permite composicionalidad: modelos simples se pueden combinar en sistemas complejos.
- Tiene una semántica formal bien definida, lo que facilita el análisis.

TUTORIAL: <https://hosting.cs.vt.edu/hpcs2008/DEVSTutorial.pdf>

4. Formalismo DEVS

4.1. Modelo atómico DEVS

Un **modelo atómico** DEVS describe el comportamiento de un componente elemental de un sistema de eventos discretos. Se define como una tupla:

$$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$$

4. Formalismo DEVS

- X : conjunto de **entradas** (eventos de entrada). Cada entrada se asocia a un puerto y un valor.
- S : conjunto de **estados internos** del modelo.
- Y : conjunto de **salidas** (eventos de salida).
- $\delta_{int} : S \rightarrow S$: **función de transición interna**. Describe cómo cambia el estado cuando vence el tiempo de permanencia del estado actual (sin eventos externos).

4. Formalismo DEVS

- $\delta_{ext} : S \times X \times \mathbb{R}^+ \rightarrow S$: **función de transición externa.** Describe cómo cambia el estado cuando llega un evento externo, antes de que venza el tiempo interno. El parámetro de tiempo indica cuánto tiempo ha transcurrido desde la última transición (interna o externa).
- $\lambda : S \rightarrow Y$: **función de salida.** Determina la salida generada justo antes de que se produzca una transición interna.
- $ta : S \rightarrow \mathbb{R}_0^+$: **función de tiempo de avance** (*time advance*). Indica cuánto tiempo permanecerá el sistema en el estado actual si no hay eventos externos.

4. Formalismo DEVS

4.3. Modelo acoplado DEVS

Los **modelos acoplados** (o *coupled models*) permiten construir sistemas complejos combinando varios modelos atómicos y/o acoplados. Un modelo acoplado DEVS se define como:

$$N = \langle X_N, Y_N, D, \{M_d\}_{d \in D}, EIC, EOC, IC \rangle$$

4. Formalismo DEVS

- X_N : conjunto de **entradas** del modelo acoplado.
- Y_N : conjunto de **salidas** del modelo acoplado.
- D : conjunto de **componentes** (índices de los submodelos).
- M_d : modelo DEVS (atómico o acoplado) para cada $d \in D$.

4. Formalismo DEVS

- *EIC (External Input Coupling)*: acoplamiento de las entradas externas X_N con las entradas de los submodelos.
- *EOC (External Output Coupling)*: acoplamiento de las salidas de los submodelos con las salidas externas Y_N .
- *IC (Internal Coupling)*: acoplamiento interno entre las salidas de unos submodelos y las entradas de otros.

Índice

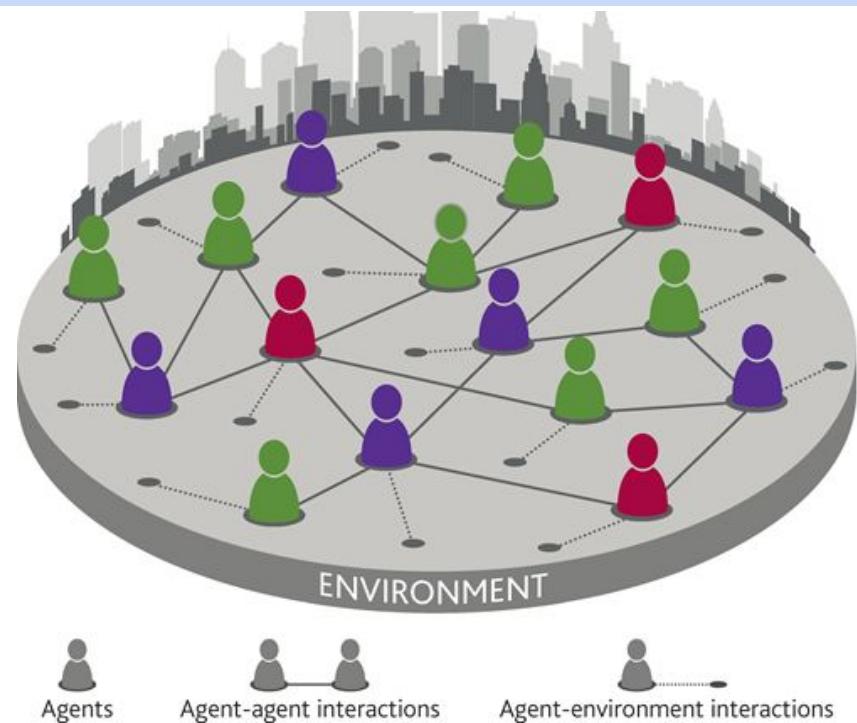
1. Introducción
2. UML para modelado de sistemas
3. SysML para modelado de sistemas físicos y ciber-físicos
4. Formalismo DEVS
- 5. Modelado basado en agentes**

5. Modelado basado en agentes

Un modelo basado en agentes se compone de tres elementos principales:

- **Agentes**: entidades autónomas que poseen un conjunto de estados internos, reglas de comportamiento y capacidad para percibir y actuar sobre su entorno.
- **Entorno**: espacio (físico o abstracto) en el que los agentes interactúan, que puede contener recursos, obstáculos o información compartida.
- **Interacciones**: mecanismos mediante los cuales los agentes comunican información o modifican su entorno.

$$A_i = \langle S_i, P_i, R_i, f_i \rangle$$



5. Modelado basado en agentes

- S_i : conjunto de estados internos del agente i .
- P_i : conjunto de percepciones que el agente obtiene del entorno o de otros agentes.
- R_i : conjunto de reglas o políticas de decisión.
- f_i : función de comportamiento que define cómo el agente cambia de estado o actúa en respuesta a P_i .

El sistema completo se describe como un conjunto de agentes:

$$\mathcal{A} = \{A_1, A_2, \dots, A_N\}$$

que interactúan dentro de un entorno E definido por sus propias dinámicas.

Tema 4: Lenguajes formales para modelos conceptuales

MODELOS COMPUTACIONALES Y SIMULACIÓN DE SISTEMAS
Curso 2025-2026



UNIVERSITAT D'ALACANT
UNIVERSIDAD DE ALICANTE
Escola Politècnica Superior
Escuela Politécnica Superior



Tema 5: Redes complejas y modelado estructural

MODELOS COMPUTACIONALES Y SIMULACIÓN DE SISTEMAS
Curso 2025-2026

Índice

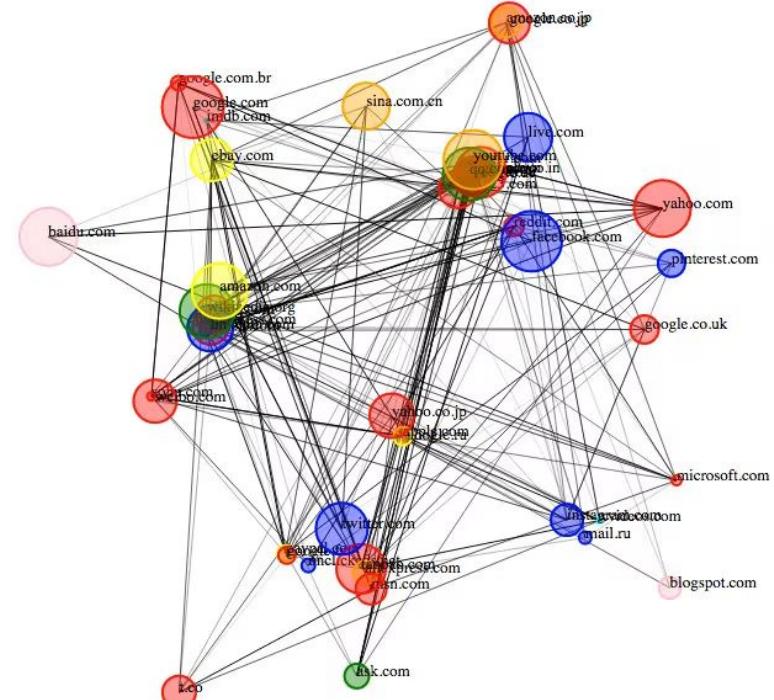
1. Introducción
2. Definición y propiedades fundamentales
3. Topologías típicas de redes complejas
 - a. Redes aleatorias (Erdős-Rényi)
 - b. Redes libres de escala (Barabási-Albert)
 - c. Redes de mundo pequeño (Watts-Strogatz)
 - d. Comparativa entre modelos de redes
4. Dinámica en redes
 - a. Difusión en redes
 - b. Sincronización de redes
 - c. Modelos epidémicos en redes
5. Estabilidad en dinámica de redes
6. Relación entre redes complejas y modelos basados en agentes

Índice

- 1. Introducción**
2. Definición y propiedades fundamentales
3. Topologías típicas de redes complejas
 - a. Redes aleatorias (Erdős-Rényi)
 - b. Redes libres de escala (Barabási-Albert)
 - c. Redes de mundo pequeño (Watts-Strogatz)
 - d. Comparativa entre modelos de redes
4. Dinámica en redes
 - a. Difusión en redes
 - b. Sincronización de redes
 - c. Modelos epidémicos en redes
5. Estabilidad en dinámica de redes
6. Relación entre redes complejas y modelos basados en agentes

1. Introducción

- Las **redes complejas** son una herramienta fundamental para representar y analizar sistemas formados por un gran número de elementos interconectados.
- **Topología no trivial**, con patrones de conexión que emergen de manera natural en sistemas biológicos, sociales, tecnológicos y físicos.

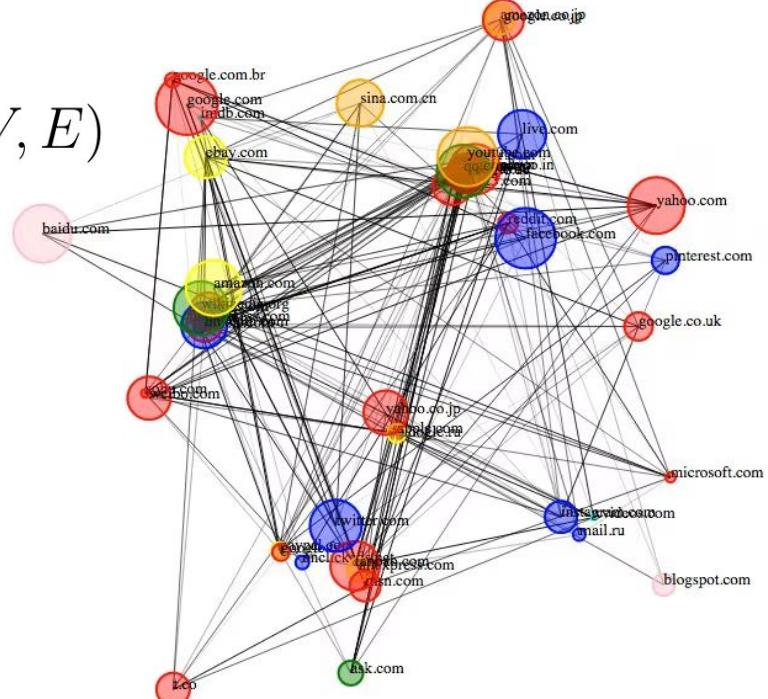


1. Introducción

- Matemáticamente, una red puede representarse mediante un grafo: $G = (V, E)$

$$V = \{v_1, v_2, \dots, v_N\}, \quad E \subseteq V \times V$$

- Permite entender **fenómenos globales** a partir de las **propiedades locales** de las conexiones.



Índice

1. Introducción
- 2. Definición y propiedades fundamentales**
3. Topologías típicas de redes complejas
 - a. Redes aleatorias (Erdős-Rényi)
 - b. Redes libres de escala (Barabási-Albert)
 - c. Redes de mundo pequeño (Watts-Strogatz)
 - d. Comparativa entre modelos de redes
4. Dinámica en redes
 - a. Difusión en redes
 - b. Sincronización de redes
 - c. Modelos epidémicos en redes
5. Estabilidad en dinámica de redes
6. Relación entre redes complejas y modelos basados en agentes

2. Definición y propiedades fundamentales

- **Representación mediante matrices**

Una red puede representarse mediante su **matriz de adyacencia** $A = [a_{ij}]$, definida como:

$$a_{ij} = \begin{cases} 1 & \text{si existe un enlace entre } i \text{ y } j, \\ 0 & \text{en otro caso.} \end{cases}$$

En redes ponderadas, el valor a_{ij} puede ser un peso $w_{ij} \in \mathbb{R}$, indicando la fuerza o intensidad de la conexión.

2. Definición y propiedades fundamentales

- **Representación mediante matrices**

La **matriz de grado** D es diagonal, con elementos $D_{ii} = k_i$, donde k_i es el grado del nodo i , es decir, el número de enlaces que posee:

$$k_i = \sum_j a_{ij}$$

A partir de estas matrices, se define la **matriz laplaciana**:

$$L = D - A$$

que juega un papel clave en el análisis dinámico sobre redes (por ejemplo, en sincronización o difusión).

2. Definición y propiedades fundamentales

- **Propiedades estructurales**
 - Algunas propiedades estadísticas básicas son:
- **Distribución de grados:** $P(k)$ indica la probabilidad de que un nodo tenga grado k . Es un indicador de la heterogeneidad de la red.

2. Definición y propiedades fundamentales

- **Propiedades estructurales**
 - Algunas propiedades estadísticas básicas son:
- **Coeficiente de clustering:** mide la tendencia de los nodos a formar triángulos o grupos cerrados. Para un nodo i :

$$C_i = \frac{2e_i}{k_i(k_i - 1)}$$

donde e_i es el número de enlaces entre los vecinos de i .

2. Definición y propiedades fundamentales

- **Propiedades estructurales**
 - Algunas propiedades estadísticas básicas son:
- Longitud promedio de camino:

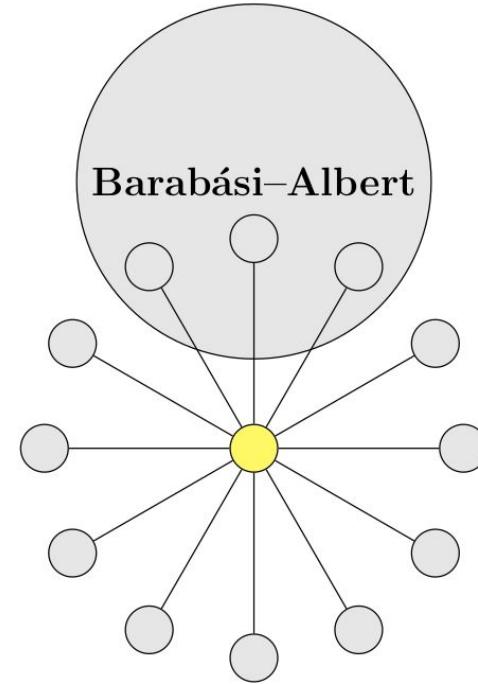
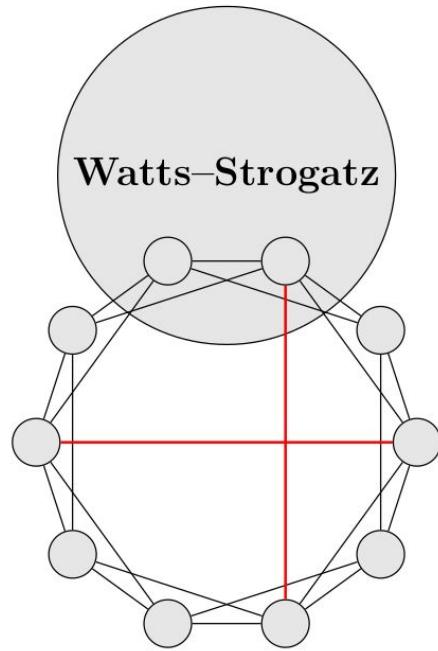
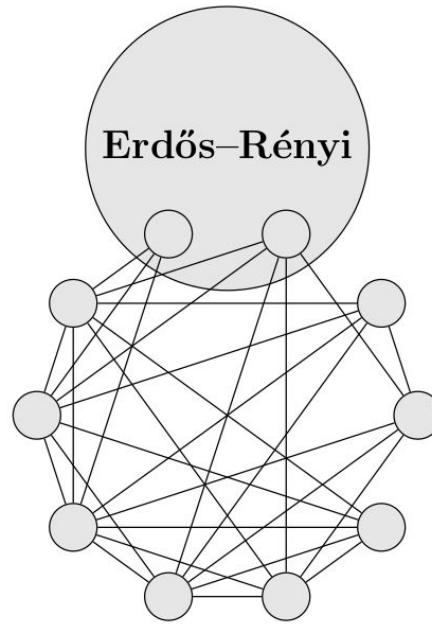
$$L = \frac{1}{N(N - 1)} \sum_{i \neq j} d(i, j)$$

donde $d(i, j)$ es la distancia más corta entre los nodos i y j .

Índice

1. Introducción
2. Definición y propiedades fundamentales
- 3. Topologías típicas de redes complejas**
 - a. Redes aleatorias (Erdős-Rényi)
 - b. Redes libres de escala (Barabási-Albert)
 - c. Redes de mundo pequeño (Watts-Strogatz)
 - d. Comparativa entre modelos de redes
4. Dinámica en redes
 - a. Difusión en redes
 - b. Sincronización de redes
 - c. Modelos epidémicos en redes
5. Estabilidad en dinámica de redes
6. Relación entre redes complejas y modelos basados en agentes

3. Topologías típicas de redes complejas

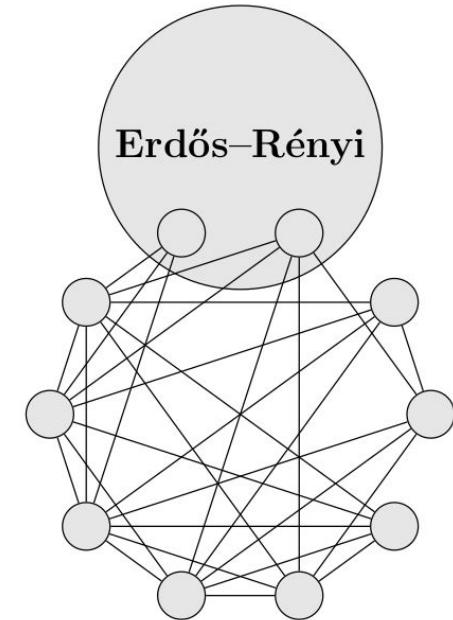


Índice

1. Introducción
2. Definición y propiedades fundamentales
3. **Topologías típicas de redes complejas**
 - a. **Redes aleatorias (Erdős-Rényi)**
 - b. Redes libres de escala (Barabási-Albert)
 - c. Redes de mundo pequeño (Watts-Strogatz)
 - d. Comparativa entre modelos de redes
4. Dinámica en redes
 - a. Difusión en redes
 - b. Sincronización de redes
 - c. Modelos epidémicos en redes
5. Estabilidad en dinámica de redes
6. Relación entre redes complejas y modelos basados en agentes

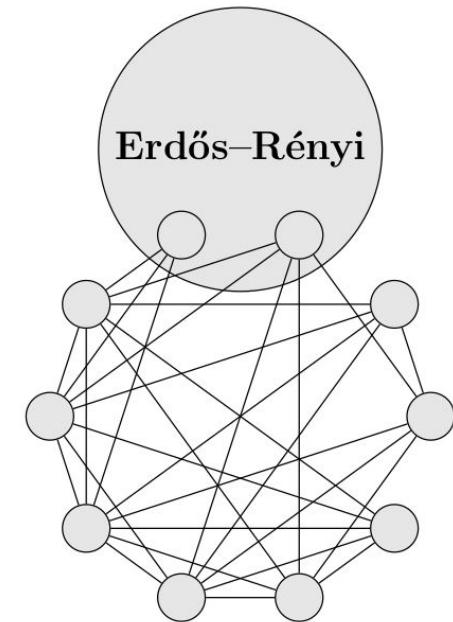
3.a. Redes aleatorias (Erdős-Rényi)

- Las **redes aleatorias** constituyen uno de los modelos fundamentales en la teoría de redes complejas.
- Utiliza **herramientas probabilísticas** para estudiar la aparición de **propiedades estructurales** en grafos generados al azar.
- Aunque el modelo es conceptualmente simple, sirve como referencia teórica para **comparar topologías más realistas**.



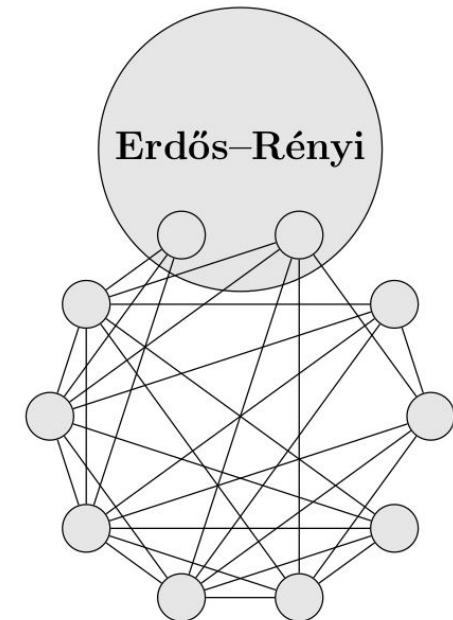
3.a. Redes aleatorias EJEMPLOS

1. **Redes de comunicación simples.** Ejemplos incluyen redes de sensores inalámbricos donde cada sensor se conecta aleatoriamente con otros dentro de un rango de comunicación, o topologías iniciales de redes *ad hoc* antes de cualquier proceso de optimización.
2. **Redes sociales no estructuradas.** Interacciones ocasionales entre individuos en espacios altamente mezclados, como eventos multitudinarios, aeropuertos o ferias, donde las conexiones no siguen patrones de homofilia ni presentan modularidad.
3. **Modelos físicos de interacción aleatoria.** La red de colisiones entre moléculas en un gas ideal o el contacto aleatorio entre partículas suspendidas en un fluido pueden aproximarse como conexiones independientes con probabilidad constante.
4. **Redes epidemiológicas bajo mezcla homogénea.** Procesos de contagio en poblaciones donde los individuos se mueven aleatoriamente y no existe estructura social marcada. En estos casos, el grafo de contactos instantáneos puede modelarse como un $G(N, p)$.



3.a. Redes aleatorias EJEMPLOS

5. **Redes de colaboración aleatoria.** En grupos donde los participantes se asignan de forma aleatoria a tareas o proyectos, la red resultante de coautoría o cooperación presenta características de una red aleatoria homogénea.
6. **Redes de fallo aleatorio en sistemas técnicos.** En modelos de robustez, los fallos pueden representarse como desconexiones aleatorias entre componentes. La estructura funcional se aproxima entonces a un grafo $G(N, p)$ con enlaces activos o inactivos al azar.
7. **Redes ecológicas con mezcla uniforme.** Si los individuos de una especie se desplazan sin territorialidad y las interacciones (competencia, contacto, depredación) se producen de forma aleatoria, la estructura puede representarse mediante una red aleatoria.
8. **Redes tecnológicas generadas sintéticamente para pruebas.** Muchas simulaciones computacionales utilizan grafos $G(N, p)$ como referencia para evaluar algoritmos de detección de comunidades, análisis de centralidad, controlabilidad o resiliencia frente a fallos.



3.a. Redes aleatorias (Erdős-Rényi)

Existen dos formulaciones equivalentes del modelo Erdős–Rényi:

1. **Modelo $G(N, p)$:** Se parte de un conjunto de N nodos y cada par de nodos se conecta de manera independiente con probabilidad p .

$$P((i, j) \in E) = p$$

2. **Modelo $G(N, M)$:** Se parte de N nodos y se seleccionan exactamente M enlaces de forma uniforme entre todos los pares posibles.

3.a. Redes aleatorias (Erdős-Rényi)

El modelo $G(N, p)$ es el más utilizado por su simplicidad para análisis analíticos, pues cada enlace se modela como una variable aleatoria Bernoulli.

El número esperado de enlaces del modelo es:

$$\mathbb{E}[|E|] = p \binom{N}{2}$$

y el grado medio de la red:

$$\langle k \rangle = p(N - 1) \approx pN$$

para N grande.

3.a. Redes aleatorias (Erdős-Rényi)

- **Distribución de grados**

Dado que cada nodo tiene probabilidad p de conectarse con cada uno de los $N - 1$ restantes, el grado k de un nodo se distribuye de manera binomial:

$$P(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k}$$

Para redes grandes ($N \gg 1$) y grados medios pequeños ($\langle k \rangle \ll N$), la distribución binomial puede aproximarse mediante una Poisson:

$$P(k) \approx \frac{\langle k \rangle^k}{k!} e^{-\langle k \rangle}$$

3.a. Redes aleatorias (Erdős-Rényi)

- **Distribución de grados**

Esto implica que:

- la mayoría de los nodos tienen un grado cercano al valor medio,
- la red es homogénea estructuralmente,
- no aparecen hubs (nodos con grado muy grande), en contraste con redes libres de escala.

3.a. Redes aleatorias (Erdős-Rényi)

- **Coeficiente de clustering**

En redes aleatorias, el clustering local (probabilidad de que dos vecinos de un nodo estén conectados entre sí) es simplemente igual a la probabilidad de enlace p :

$$C = p = \frac{\langle k \rangle}{N}$$

El clustering es típicamente muy bajo, lo que contrasta con redes de mundo pequeño o redes sociales reales, donde suele observarse un clustering elevado.

3.a. Redes aleatorias (Erdős-Rényi)

- **Longitud de camino promedio**

En el modelo $G(N, p)$, la longitud media del camino más corto entre dos nodos escala como:

$$L \sim \frac{\ln N}{\ln \langle k \rangle}$$

Esto indica que las redes aleatorias presentan la propiedad de **mundo pequeño**: incluso con un número grande de nodos, la distancia promedio crece muy lentamente.

3.a. Redes aleatorias (Erdős-Rényi)

Supongamos una red $G(N, p)$ con $N = 100$ nodos y $p = 0,05$.

$$\langle k \rangle = p(N - 1) \approx 4,95$$

Esto implica que:

- Cada nodo tiene en promedio 5 conexiones.
- El clustering esperado es $C = p = 0,05$.
- La distancia media:

$$L \approx \frac{\ln 100}{\ln 5} \approx 2,86$$

es decir, cualquier nodo está separado de otro por menos de 3 pasos.

3.a. Redes aleatorias (Erdős-Rényi)

Las características más importantes del modelo Erdős–Rényi pueden resumirse así:

- Es un modelo **homogéneo**: todos los nodos tienen aproximadamente el mismo grado.
- No contiene **hubs**, por lo que no sigue una ley de potencias.
- El clustering es bajo, lo cual no refleja la mayoría de redes reales.
- Presenta una **distancia media pequeña** entre nodos.
- Es un modelo ideal para análisis matemático y como punto de referencia para otras topologías.

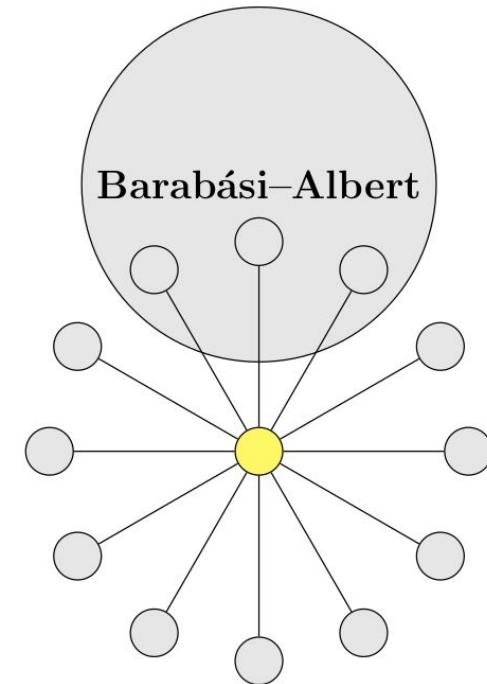
<https://www.networkpages.nl/CustomMedia/Animations/RandomGraph/ERRG/AddoneEdgeAtATime.html>

Índice

1. Introducción
2. Definición y propiedades fundamentales
3. **Topologías típicas de redes complejas**
 - a. Redes aleatorias (Erdős-Rényi)
 - b. Redes libres de escala (Barabási-Albert)**
 - c. Redes de mundo pequeño (Watts-Strogatz)
 - d. Comparativa entre modelos de redes
4. Dinámica en redes
 - a. Difusión en redes
 - b. Sincronización de redes
 - c. Modelos epidémicos en redes
5. Estabilidad en dinámica de redes
6. Relación entre redes complejas y modelos basados en agentes

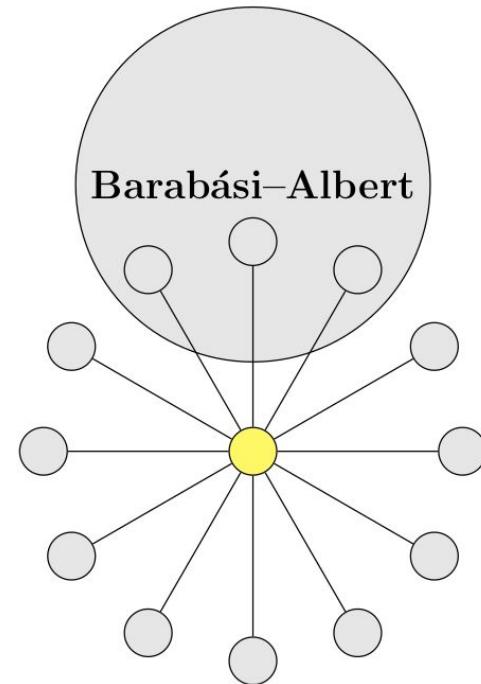
3.b. Redes libres de escala (Barabási-Albert)

- Las redes **libres de escala** constituyen una clase de topologías caracterizadas por la presencia de unos pocos **nodos altamente conectados** (denominados hubs) y una gran cantidad de **nodos con bajo grado**.
- proporciona un mecanismo generativo sencillo que explica la emergencia de estas estructuras mediante los principios de **crecimiento y conexión preferencial**.
- Aparece de manera natural en **numerosos sistemas reales** y se describe mediante distribuciones de grados de tipo ley de potencias.



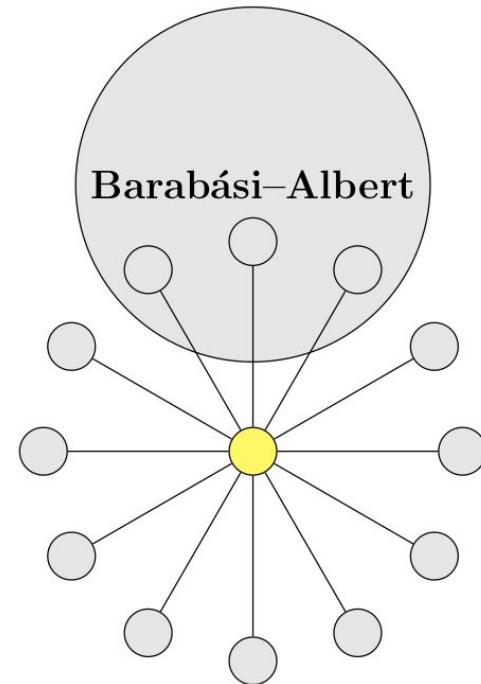
3.b. Redes libres de escala EJEMPLOS

1. **Internet a nivel de routers y dominios.** Estudios empíricos muestran que la distribución de grados de routers y sistemas autónomos sigue leyes de potencias.
2. **Redes sociales reales.** Ejemplos como Facebook, Twitter o redes de colaboración científica presentan nodos muy conectados (influencers o autores prolíficos) junto a miles de nodos con pocas conexiones.
3. **Redes metabólicas y bioquímicas.** Ciertos metabolitos actúan como hubs (ATP, NADH), conectando numerosos procesos.
4. **Redes de interacción proteína–proteína.** Proteínas clave poseen alto grado, mientras la mayoría tienen pocos enlaces.
5. **Redes de transporte aéreo.** Ciudades como Londres, Nueva York o Dubái funcionan como hubs intercontinentales.



3.b. Redes libres de escala EJEMPLOS

6. **Redes financieras.** Grandes bancos o instituciones actúan como nodos altamente conectados en la red de préstamos o derivados.
7. **Redes neuronales estructurales.** Algunas neuronas o regiones cerebrales actúan como superconectores.
8. **Redes de distribución eléctrica.** Estaciones de transformación principales funcionan como hubs en la infraestructura.
9. **Redes lingüísticas.** Palabras muy frecuentes actúan como hubs dentro de redes semánticas.
10. **Redes de citas científicas.** Artículos seminales reciben un gran número de citas, mientras la mayoría reciben pocas.



3.b. Redes libres de escala (Barabási-Albert)

1. Crecimiento La red no es estática. Comienza con un número pequeño de nodos, m_0 , y en cada paso se añade un nodo nuevo.

2. Conexión preferencial El nodo nuevo se conecta a m nodos existentes con una probabilidad proporcional al grado de los nodos ya presentes. Es decir, un nodo con mayor grado tiene mayor probabilidad de recibir nuevas conexiones.

Formalmente, la probabilidad de que un nodo i sea escogido para recibir un enlace es:

$$\Pi(k_i) = \frac{k_i}{\sum_j k_j}$$

Este mecanismo genera la **retroalimentación positiva** que conduce a la aparición de hubs.

3.b. Redes libres de escala (Barabási-Albert)

- **Heterogeneidad extrema:** el grado medio no es representativo de la estructura real.
- **Hubs bien definidos:** unos pocos nodos concentran una fracción significativa de las conexiones.
- **Robustez frente a fallos aleatorios:** eliminar nodos al azar no suele fragmentar la red.
- **Vulnerabilidad a ataques dirigidos:** eliminar hubs puede desconectar la red rápidamente.
- **Longitud de camino promedio pequeña,** con crecimiento del orden:

$$L \sim \frac{\ln N}{\ln \ln N}$$

3.b. Redes libres de escala (Barabási-Albert)

Consideremos una red BA con:

$$N = 1000, \quad m_0 = 5, \quad m = 3.$$

Propiedades típicas obtenidas por simulación:

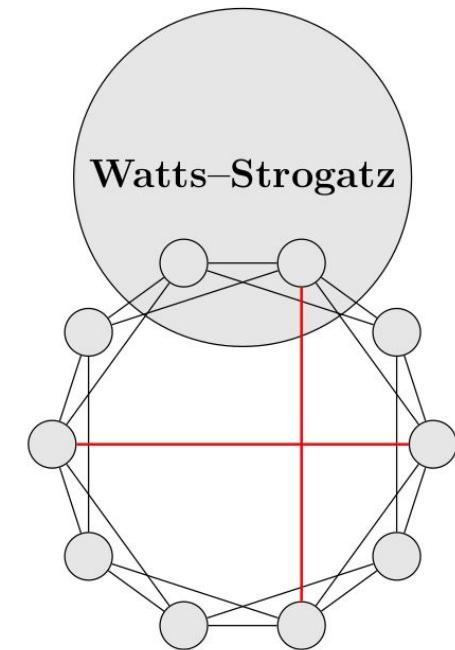
- La distribución de grados sigue aproximadamente $P(k) \sim k^{-3}$.
- El nodo más conectado puede alcanzar un grado del orden de $k_{\text{máx}} \approx 60\text{--}80$.
- La longitud promedio del camino es cercana a $L \approx 3$.
- El clustering es mayor que en una red aleatoria con el mismo grado medio, pero sigue siendo relativamente bajo.

Índice

1. Introducción
2. Definición y propiedades fundamentales
3. **Topologías típicas de redes complejas**
 - a. Redes aleatorias (Erdős-Rényi)
 - b. Redes libres de escala (Barabási-Albert)
 - c. **Redes de mundo pequeño (Watts-Strogatz)**
 - d. Comparativa entre modelos de redes
4. Dinámica en redes
 - a. Difusión en redes
 - b. Sincronización de redes
 - c. Modelos epidémicos en redes
5. Estabilidad en dinámica de redes
6. Relación entre redes complejas y modelos basados en agentes

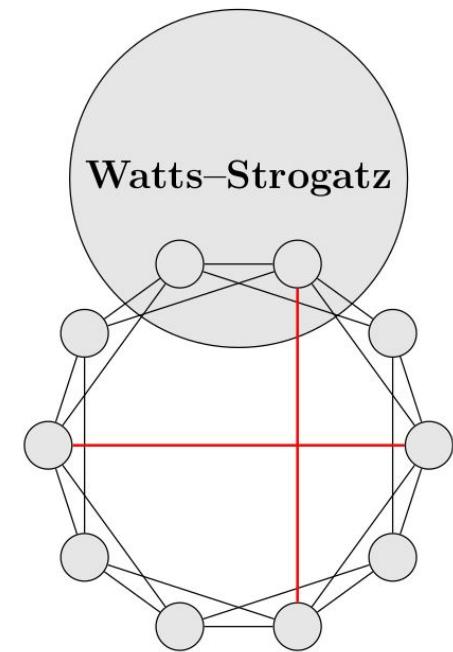
3.c. Redes mundo pequeño (Watts-Strogatz)

- representan una topología que combina simultáneamente dos propiedades fundamentales que se observan en muchos sistemas reales: un **alto coeficiente de clustering** y una **baja longitud promedio de camino**.
- modelo intermedio entre redes regulares y redes aleatorias, capaz de reproducir fenómenos de **conectividad eficiente con estructuras altamente locales**.



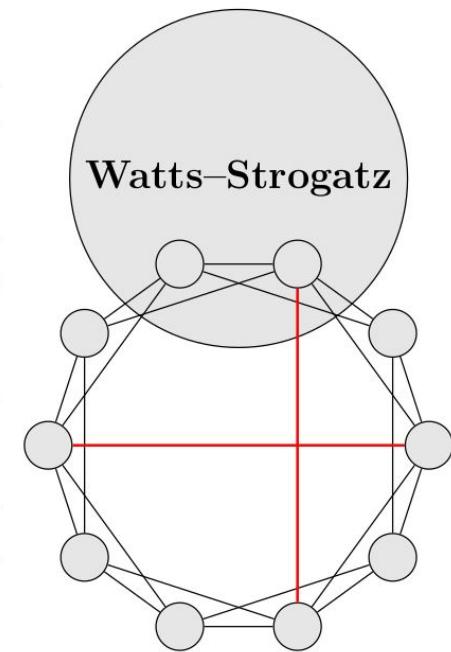
3.c. Redes mundo pequeño EJEMPLOS

1. **Redes sociales humanas.** La mayoría de redes sociales exhiben alto clustering (amigos de un individuo suelen conocerse entre sí) y caminos globales cortos.
2. **Redes neurológicas y cerebrales.** El cerebro presenta agrupaciones locales muy fuertes (columnas corticales) con conexiones largas que actúan como atajos entre regiones.
3. **Redes de colaboración científica.** Los investigadores colaboran en grupos altamente conectados, pero existen enlaces entre grupos que reducen la distancia entre disciplinas.
4. **Redes de transporte.** Muchas infraestructuras de metro, tren y autopistas presentan rutas locales densas y enlaces rápidos entre zonas alejadas.



3.c. Redes mundo pequeño EJEMPLOS

5. **Redes de energía en ciudades.** Las redes eléctricas urbanas se organizan en módulos locales con conexiones redundantes que reducen la distancia entre componentes.
6. **Redes de comunicación entre robots móviles.** Los robots tienden a comunicarse con vecinos cercanos, pero algunos enlaces de largo alcance reducen los tiempos de coordinación.
7. **Redes semánticas y lingüísticas.** Palabras relacionadas forman grupos densos, pero existen conexiones semánticas que actúan como atajos entre conceptos.
8. **Redes de interacciones en ecosistemas.** Grupos de especies presentan fuerte concentración local, mientras unas pocas relaciones de larga distancia conectan diferentes módulos ecológicos.



3.c. Redes mundo pequeño (Watts-Strogatz)

El modelo original se construye mediante el siguiente procedimiento:

1. Red regular inicial Se parte de un anillo de N nodos, donde cada nodo está conectado simétricamente a sus k vecinos más cercanos (habitualmente $k \ll N$).

2. Reenlace aleatorio con probabilidad p Cada enlace de la red regular se **reconecta** con probabilidad p hacia un nodo elegido al azar, evitando duplicar enlaces y lazos propios. El parámetro p controla la transición entre:

- $p = 0$: red regular altamente ordenada,
- $p = 1$: red aleatoria (modelo Erdős–Rényi),
- $0 < p \ll 1$: **red de mundo pequeño**.

3.c. Redes mundo pequeño (Watts-Strogatz)

1. Coeficiente de clustering La red regular inicial tiene un clustering alto:

$$C_{WS}(p = 0) \approx \frac{3(k - 2)}{4(k - 1)}$$

Cuando p aumenta ligeramente, el clustering disminuye, pero sigue siendo mucho mayor que en una red aleatoria con el mismo grado medio.

3.c. Redes mundo pequeño (Watts-Strogatz)

2. Longitud media de camino Para $p = 0$, la longitud media crece linealmente con N :

$$L_{WS}(0) \sim \frac{N}{2k}$$

Cuando p toma valores muy pequeños (por ejemplo, $p \approx 0,01$), se introducen *atajos* que reducen drásticamente la distancia entre nodos, provocando una caída abrupta de L :

$$L_{WS}(p \ll 1) \approx L_{\text{aleatoria}} \sim \frac{\ln N}{\ln k}$$

Mientras tanto, el clustering se mantiene elevado. Este régimen constituye la propiedad de **mundo pequeño**.

3.c. Redes mundo pequeño (Watts-Strogatz)

$$N = 1000, \quad k = 10, \quad p = 0,01.$$

Propiedades típicas:

- Clustering inicial (antes del reenlace):

$$C(0) \approx 0,75.$$

- Clustering para $p = 0,01$:

$$C(0,01) \approx 0,60,$$

mucho mayor que en una red aleatoria equivalente ($C \approx 0,01$).

- Longitud media del camino:

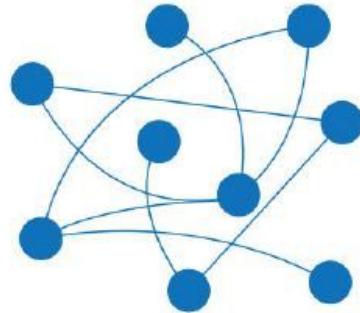
$$L(0,01) \approx 3,2,$$

cercana a la longitud de camino de una red aleatoria.

Índice

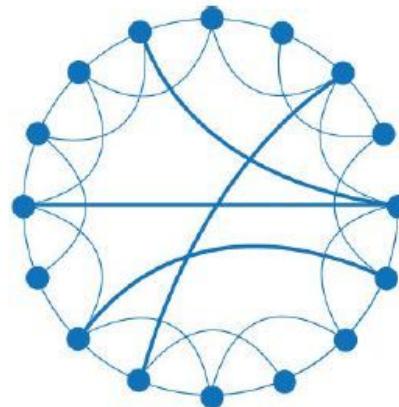
1. Introducción
2. Definición y propiedades fundamentales
3. **Topologías típicas de redes complejas**
 - a. Redes aleatorias (Erdős-Rényi)
 - b. Redes libres de escala (Barabási-Albert)
 - c. Redes de mundo pequeño (Watts-Strogatz)
 - d. **Comparativa entre modelos de redes**
4. Dinámica en redes
 - a. Difusión en redes
 - b. Sincronización de redes
 - c. Modelos epidémicos en redes
5. Estabilidad en dinámica de redes
6. Relación entre redes complejas y modelos basados en agentes

3.d. Comparativa entre modelos de redes



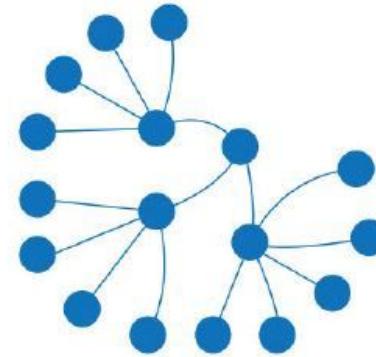
Random

Average distributions.
No structure or hierachal patterns.



Small-World

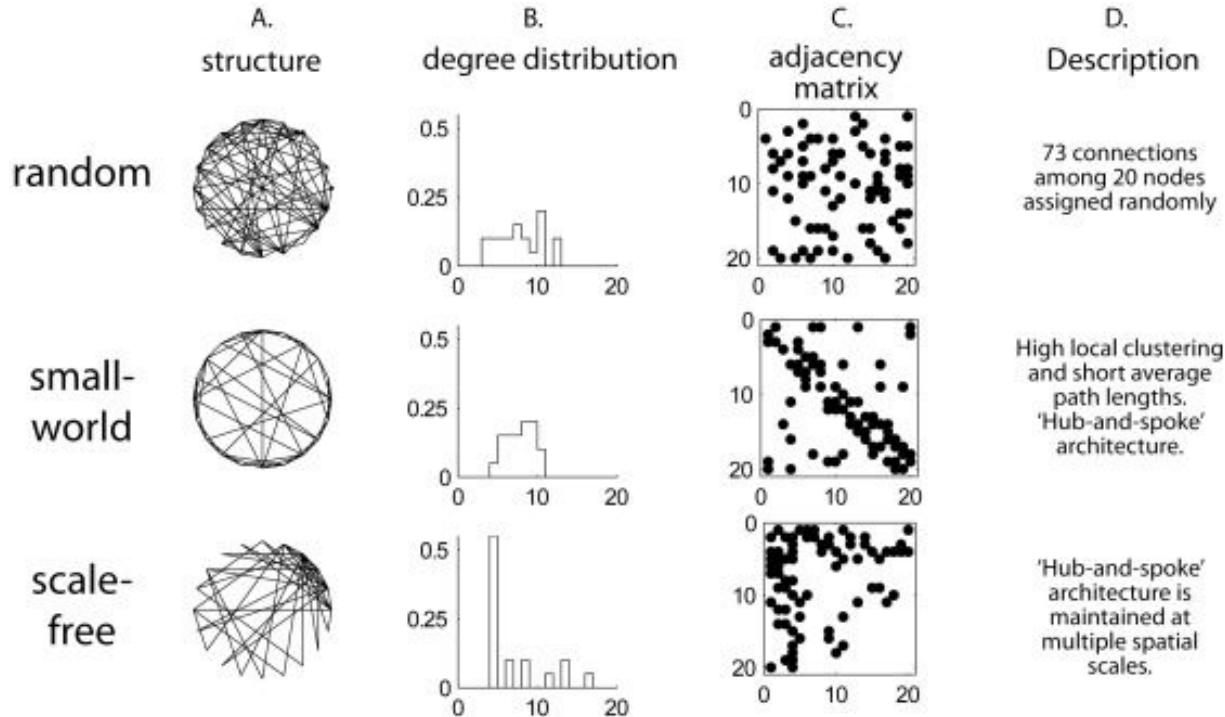
High local clustering and short
average path lengths.
Hub-and-spoke architecture.



Scale-Free

Hub-and-spoke architecture preserved at
multiple scales.
High power law distribution.

3.d. Comparativa entre modelos de redes



Propiedad	Erdős–Rényi (ER)	Watts–Strogatz (WS)	Barabási–Albert (BA)
Mecanismo generativo	Enlaces independientes con probabilidad p .	Reenlace aleatorio sobre red regular con probabilidad p .	Crecimiento + conexión preferencial.
Distribución de grados	Binomial/Poisson.	Concentrada alrededor de k .	Ley de potencias $P(k) \sim k^{-\gamma}$.
Heterogeneidad de grados	Baja (homogénea).	Moderada.	Alta (hubs).
Coeficiente de clustering	Muy bajo: $C = p$.	Alto para $p \ll 1$.	Moderado, decrece con el tamaño.
Longitud media del camino	$L \sim \frac{\ln N}{\ln \langle k \rangle}$.	Muy baja incluso para $p \ll 1$.	Muy baja: $L \sim \ln N / \ln \ln N$.
Estructura modular	No.	Sí (clustering local).	Moderada, emergen comunidades débiles.
Presencia de hubs	No.	No.	Sí, muy pronunciados.
Robustez a fallos aleatorios	Baja.	Media.	Muy alta.
Vulnerabilidad a ataques dirigidos	Media.	Media.	Muy alta (cae si se eliminan hubs).
Aplicaciones típicas	Interacciones aleatorias, redes sintéticas.	Redes sociales, neuronales, transporte.	Internet, redes sociales reales, biología molecular.

3.d. Comparativa entre modelos de redes

- **Para simular redes y su dinámica:**
<https://networkx.org/documentation/stable/index.html>



Índice

1. Introducción
2. Definición y propiedades fundamentales
3. Topologías típicas de redes complejas
 - a. Redes aleatorias (Erdős-Rényi)
 - b. Redes libres de escala (Barabási-Albert)
 - c. Redes de mundo pequeño (Watts-Strogatz)
 - d. Comparativa entre modelos de redes
- 4. Dinámica en redes**
 - a. Difusión en redes
 - b. Sincronización de redes
 - c. Modelos epidémicos en redes
5. Estabilidad en dinámica de redes
6. Relación entre redes complejas y modelos basados en agentes

4. Dinámica de redes

- La **dinámica en redes** estudia cómo ciertos procesos evolucionan en sistemas donde los elementos interactúan **a través de una estructura de conexiones** representada mediante un grafo.
- A diferencia de los modelos tradicionales que asumen interacciones globales o uniformes, la dinámica en redes **incorpora explícitamente la topología** subyacente, lo que modifica de manera profunda el **comportamiento colectivo del sistema**.

4. Dinámica de redes

- Algunos de los procesos dinámicos más estudiados incluyen:
 - **Difusión** de información, energía o sustancias.
 - **Sincronización** de osciladores.
 - **Propagación** de epidemias o rumores.
 - Dinámica de consenso.
 - Procesos de random walks.
 - Dinámica de opinión o comportamiento social.

Índice

1. Introducción
2. Definición y propiedades fundamentales
3. Topologías típicas de redes complejas
 - a. Redes aleatorias (Erdős-Rényi)
 - b. Redes libres de escala (Barabási-Albert)
 - c. Redes de mundo pequeño (Watts-Strogatz)
 - d. Comparativa entre modelos de redes
- 4. Dinámica en redes**
 - a. Difusión en redes**
 - b. Sincronización de redes
 - c. Modelos epidémicos en redes
5. Estabilidad en dinámica de redes
6. Relación entre redes complejas y modelos basados en agentes

4.a. Difusión en redes

- La **difusión en redes** es un proceso mediante el cual una cantidad (información, calor, moléculas, energía o probabilidad) se **redistribuye a través de los nodos**, siguiendo las conexiones definidas por la red.
- Este tipo de proceso se modela habitualmente mediante la **matriz laplaciana**, que captura cómo la diferencia entre estados locales impulsa la transferencia entre nodos.

4.a. Difusión en redes

Sea una red definida por un grafo $G = (V, E)$ con matriz de adyacencia A y matriz laplaciana $L = D - A$, donde D es la matriz de grados. Si $x_i(t)$ representa la magnitud asociada al nodo i en el tiempo t , el modelo continuo de difusión viene dado por:

$$\frac{dx_i}{dt} = - \sum_j L_{ij}x_j.$$

En forma vectorial:

$$\frac{d\mathbf{x}}{dt} = -L\mathbf{x}.$$

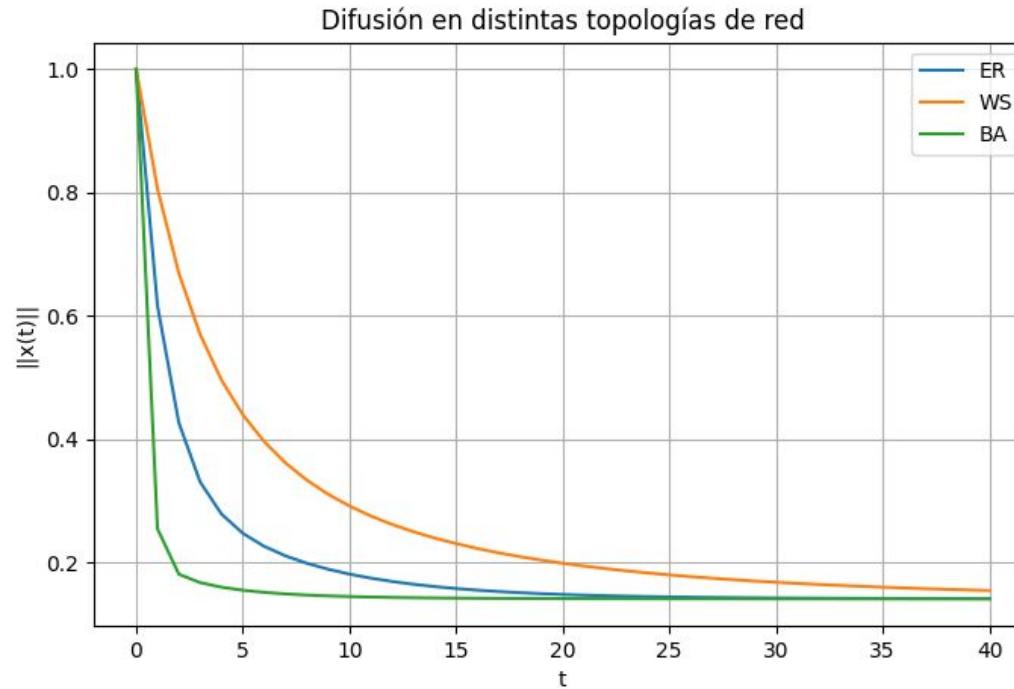
Este sistema converge hacia un estado uniforme cuando la red es conexa, pues el laplaciano posee un autovalor cero asociado a la solución estacionaria.

4.a. Difusión en redes

La ecuación de difusión indica que **cada nodo ajusta su valor hacia los valores de sus vecinos**. Si un nodo tiene un valor alto y sus vecinos valores más bajos, la cantidad fluye hacia ellos. Esto imita fenómenos como:

- Nivelación de temperatura (conducción térmica).
- Difusión de un rumor en redes sociales.
- Transferencia de carga o energía.
- Promedios iterativos en sistemas de formación en robótica.

4.a. Difusión en redes



Índice

1. Introducción
2. Definición y propiedades fundamentales
3. Topologías típicas de redes complejas
 - a. Redes aleatorias (Erdős-Rényi)
 - b. Redes libres de escala (Barabási-Albert)
 - c. Redes de mundo pequeño (Watts-Strogatz)
 - d. Comparativa entre modelos de redes
- 4. Dinámica en redes**
 - a. Difusión en redes
 - b. Sincronización de redes**
 - c. Modelos epidémicos en redes
5. Estabilidad en dinámica de redes
6. Relación entre redes complejas y modelos basados en agentes

4.b. Sincronización de redes

- La sincronización en redes describe cómo un **conjunto de osciladores acoplados** tienden a ajustar sus fases o frecuencias debido a las interacciones entre ellos.
- Este fenómeno es fundamental en áreas como neurociencia, sistemas eléctricos, relojes distribuidos, redes de sensores y dinámica colectiva de robots.
- El modelo más utilizado para describir sincronización es el **modelo de Kuramoto**.

4.b. Sincronización de redes

- **Modelo de Kuramoto.**

Sea una red con matriz de adyacencia A . Cada nodo i tiene una fase $\theta_i(t)$ y una frecuencia natural ω_i . El modelo se define como:

$$\frac{d\theta_i}{dt} = \omega_i + K \sum_j a_{ij} \sin(\theta_j - \theta_i),$$

donde K es el parámetro de acoplamiento.

La red influye en la sincronización mediante a_{ij} : existen redes donde la sincronización ocurre fácilmente y otras donde es difícil lograr coherencia global.

4.b. Sincronización de redes

- Cada oscilador intenta seguir su frecuencia natural, pero las conexiones con sus vecinos lo “empujan” a sincronizarse. Si K es suficientemente grande, **los nodos alcanzan un estado de fase común**.
- En una red regular donde cada oscilador se conecta sólo con dos vecinos, la sincronización es lenta y requiere un acoplamiento mayor. En redes libres de escala, **los hubs aceleran enormemente la sincronización global**.
- Las **redes BA son particularmente eficientes** para sincronizar porque los hubs actúan como puntos de anclaje. Sin embargo, esta misma estructura las vuelve vulnerables: si un hub falla, la sincronización puede colapsar. Ejemplos donde aparece sincronización:
 - Redes de neuronas (señales rítmicas);
 - Redes eléctricas (frecuencia estable de 50/60 Hz);
 - Enjambres de robots (consenso en movimiento);
 - Biología (osciladores químicos, ritmos circadianos).

4.b. Sincronización de redes

Dado un conjunto de N osciladores con fases $\theta_1(t), \dots, \theta_N(t)$, el parámetro de orden se define como el módulo del promedio de los vectores unitarios asociados a cada fase:

$$r(t)e^{i\psi(t)} = \frac{1}{N} \sum_{j=1}^N e^{i\theta_j(t)},$$

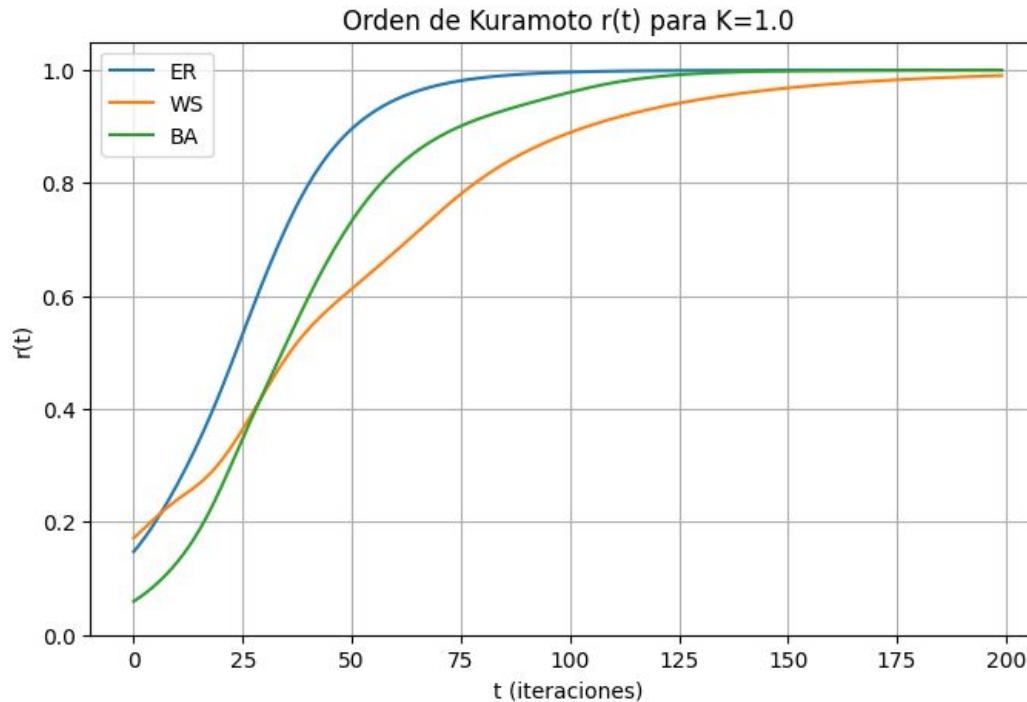
donde:

- $r(t)$ mide el grado de coherencia global,
- $\psi(t)$ es la *fase promedio* del sistema.

Tomando el módulo de la expresión anterior, se obtiene la forma más comúnmente usada:

$$r(t) = \left| \frac{1}{N} \sum_{j=1}^N e^{i\theta_j(t)} \right|. \quad 0 \leq r(t) \leq 1.$$

4.b. Sincronización de redes



Índice

1. Introducción
2. Definición y propiedades fundamentales
3. Topologías típicas de redes complejas
 - a. Redes aleatorias (Erdős-Rényi)
 - b. Redes libres de escala (Barabási-Albert)
 - c. Redes de mundo pequeño (Watts-Strogatz)
 - d. Comparativa entre modelos de redes
- 4. Dinámica en redes**
 - a. Difusión en redes
 - b. Sincronización de redes
 - c. Modelos epidémicos en redes**
5. Estabilidad en dinámica de redes
6. Relación entre redes complejas y modelos basados en agentes

4.c. Modelos epidémicos en redes

- La **propagación de epidemias** es uno de los campos donde la teoría de redes ha tenido mayor impacto.
- A diferencia de los modelos epidemiológicos clásicos (SIR, SIS), donde se asume mezcla homogénea, los modelos en red **permiten especificar quién puede contagiar a quién** a partir de una estructura de contactos.

4.c. Modelos epidémicos en redes

Cada nodo representa un individuo y puede encontrarse en estado susceptible (S), infectado (I) o recuperado (R). La dinámica discreta puede escribirse como:

$$S_i(t+1) = S_i(t) - \beta S_i(t) \sum_j a_{ij} I_j(t),$$

$$I_i(t+1) = I_i(t) + \beta S_i(t) \sum_j a_{ij} I_j(t) - \gamma I_i(t),$$

$$R_i(t+1) = R_i(t) + \gamma I_i(t),$$

donde:

- a_{ij} indica si el individuo i puede ser contagiado por j ;
- β es la tasa de transmisión;
- γ es la tasa de recuperación.

4.c. Modelos epidémicos en redes

Una de las conclusiones clave de los modelos epidémicos en redes es que el umbral epidémico depende fuertemente de la heterogeneidad de la red. En una red con distribución de grados $P(k)$, el umbral aproximado es:

$$\lambda_c = \frac{\langle k \rangle}{\langle k^2 \rangle},$$

donde $\lambda = \beta/\gamma$.

- En redes aleatorias ER: $\langle k^2 \rangle$ es pequeño \rightarrow umbral positivo.
- En redes libres de escala: $\langle k^2 \rangle$ puede divergir \rightarrow umbral cercano a 0.

Esto implica que en redes libres de escala incluso infecciones muy débiles pueden generar epidemias masivas.

4.c. Modelos epidémicos en redes

En un grafo ER con $\langle k \rangle = 5$:

$$\lambda_c = \frac{5}{5^2 + 5} \approx 0,17.$$

La infección sólo se vuelve epidémica si $\beta/\gamma > 0,17$.

4.c. Modelos epidémicos en redes

En una red BA donde existen hubs con alto grado, $\langle k^2 \rangle$ es muy grande:

$$\lambda_c \rightarrow 0.$$

Esto explica por qué:

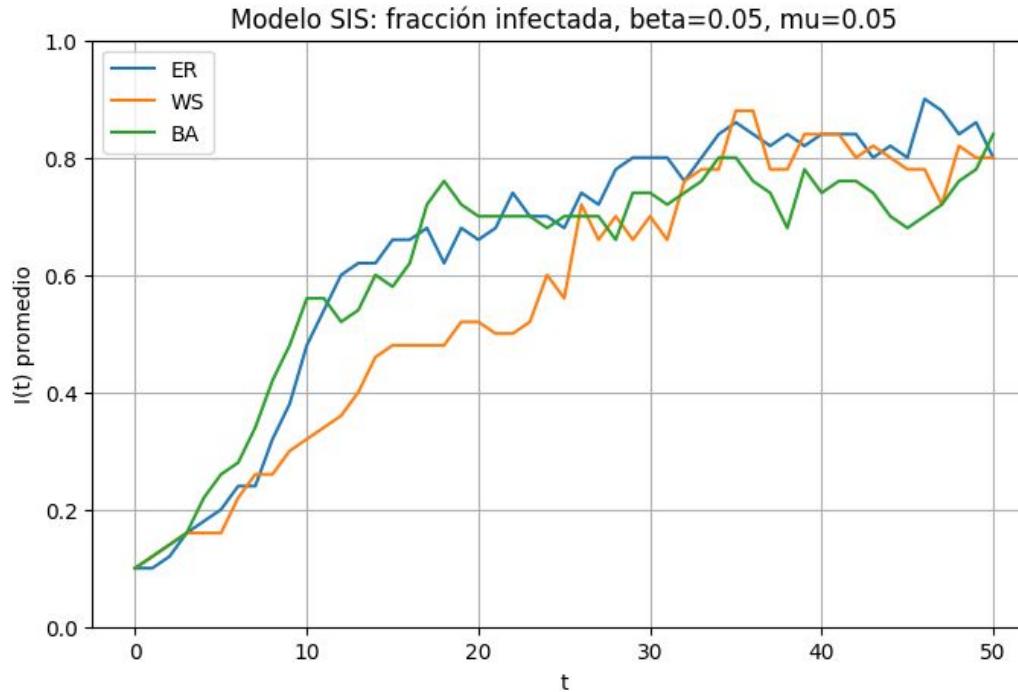
- los virus informáticos se propagan fácilmente por Internet,
- las epidemias reales se benefician de “supercontagiadores”,
- los hubs son puntos críticos para controlar epidemias.

4.c. Modelos epidémicos en redes

Los modelos epidémicos en redes se aplican a:

- Contagio de enfermedades (COVID-19, gripe, sarampión);
- Propagación de rumores o fake news en redes sociales;
- Difusión de malware en sistemas informáticos;
- Propagación de fallos en redes eléctricas.

4.c. Modelos epidémicos en redes



Índice

1. Introducción
2. Definición y propiedades fundamentales
3. Topologías típicas de redes complejas
 - a. Redes aleatorias (Erdős-Rényi)
 - b. Redes libres de escala (Barabási-Albert)
 - c. Redes de mundo pequeño (Watts-Strogatz)
 - d. Comparativa entre modelos de redes
4. Dinámica en redes
 - a. Difusión en redes
 - b. Sincronización de redes
 - c. Modelos epidémicos en redes
- 5. Estabilidad en dinámica de redes**
6. Relación entre redes complejas y modelos basados en agentes

5. Estabilidad en dinámica de redes

La diferencia esencial con la dinámica de un sistema individual es que, en redes, la estabilidad depende fuertemente de:

- la **dinámica local** de cada nodo (las ecuaciones que gobiernan su evolución),
- el **acoplamiento** entre nodos (ganancias, funciones de interacción, intensidad de conexión),
- la **topología del grafo**, capturada en la estructura de A y L y en sus **autovalores**.

Esto implica que:

- Dos redes con la misma dinámica local y el mismo acoplamiento pueden tener comportamientos estables o inestables diferentes, simplemente por estar conectadas de distinta manera.
- La presencia de *hubs*, comunidades o cuellos de botella topológicos puede facilitar u obstaculizar procesos como la sincronización o la propagación epidémica.

5. Estabilidad en dinámica de redes

En muchos casos, aparecen cantidades espectrales clave, como:

- el **segundo autovalor más pequeño del laplaciano**, $\lambda_2(L)$, o *conectividad algebraica*;
- el **mayor autovalor de la matriz de adyacencia**, $\lambda_{\max}(A)$.

Índice

1. Introducción
2. Definición y propiedades fundamentales
3. Topologías típicas de redes complejas
 - a. Redes aleatorias (Erdős-Rényi)
 - b. Redes libres de escala (Barabási-Albert)
 - c. Redes de mundo pequeño (Watts-Strogatz)
 - d. Comparativa entre modelos de redes
4. Dinámica en redes
 - a. Difusión en redes
 - b. Sincronización de redes
 - c. Modelos epidémicos en redes
5. Estabilidad en dinámica de redes
- 6. Relación entre redes complejas y modelos basados en agentes**

6. Redes complejas y m basados en agentes

Aspecto	Redes complejas	Modelado basado en agentes
Representación	Relación entre nodos.	Comportamiento individual de entidades.
Naturaleza	Estructural, topológica.	Dinámica, orientada a reglas.
Formalismo	Teoría de grafos, matrices, análisis estadístico.	Simulación discreta, autómatas, programación.
Objetivo	Analizar propiedades globales de la red.	Capturar emergencias desde acciones locales.
Enfoque	Quién conecta con quién.	Qué hace cada agente y cómo interactúa.

Temario

B1 - Sistemas Dinámicos

T1: Fundamentos del modelado de sistemas dinámicos

T2: Estabilidad, controlabilidad y observabilidad

T3: Paradigmas de simulación

B2 - Sistemas Complejos

T4: Lenguajes formales para modelos conceptuales

T5: Redes complejas y modelado estructural

B3 - Modelado con IA

T6: Identificación de sistemas

T7: Inteligencia artificial aplicada al modelado de sistemas

Tema 5: Redes complejas y modelado estructural

MODELOS COMPUTACIONALES Y SIMULACIÓN DE SISTEMAS
Curso 2025-2026

Tema 6: Identificación de sistemas

MODELOS COMPUTACIONALES Y SIMULACIÓN DE SISTEMAS
Curso 2025-2026



UNIVERSITAT D'ALACANT
UNIVERSIDAD DE ALICANTE
Escola Politècnica Superior
Escuela Politécnica Superior



Temario

B1 - Sistemas Dinámicos

T1: Fundamentos del modelado de sistemas dinámicos

T2: Estabilidad, controlabilidad y observabilidad

T3: Paradigmas de simulación

B2 - Sistemas Complejos

T4: Lenguajes formales para modelos conceptuales

T5: Redes complejas y modelado estructural

B3 - Modelado con IA

T6: Identificación de sistemas

T7: Inteligencia artificial aplicada al modelado de sistemas

Índice

1. Introducción
2. Fundamentos de la identificación de sistemas a partir de datos
3. Modelos paramétricos vs. no paramétricos
4. Métodos clásicos
 - a. Método de mínimos cuadrados (LS)
 - b. Modelos ARX
 - c. Modelos ARMAX
 - d. Recursividad y mínimos cuadrados
5. Introducción a redes neuronales y modelos híbridos
 - a. Redes neuronales para identificación
 - b. Modelos híbridos: físicos + datos
6. Evaluación de desempeño del modelo

Índice

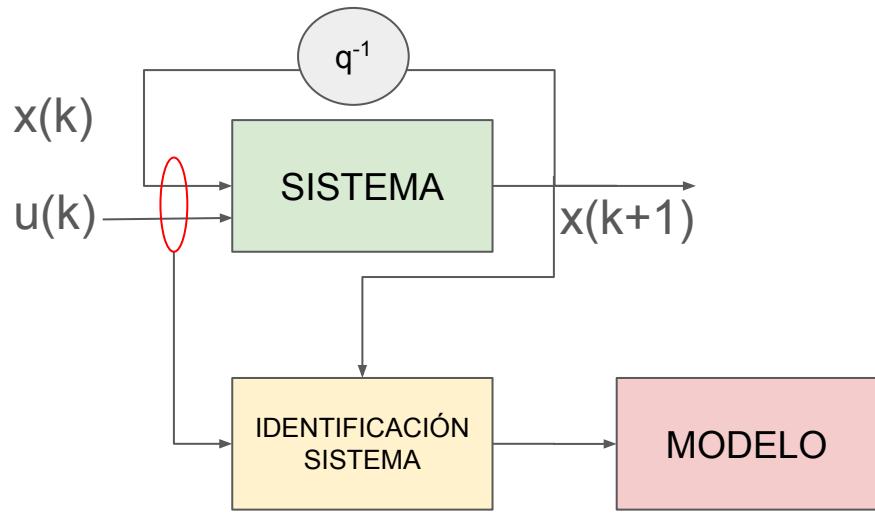
- 1. Introducción**
2. Fundamentos de la identificación de sistemas a partir de datos
3. Modelos paramétricos vs. no paramétricos
4. Métodos clásicos
 - a. Método de mínimos cuadrados (LS)
 - b. Modelos ARX
 - c. Modelos ARMAX
 - d. Recursividad y mínimos cuadrados
5. Introducción a redes neuronales y modelos híbridos
 - a. Redes neuronales para identificación
 - b. Modelos híbridos: físicos + datos
6. Evaluación de desempeño del modelo

1. Introducción

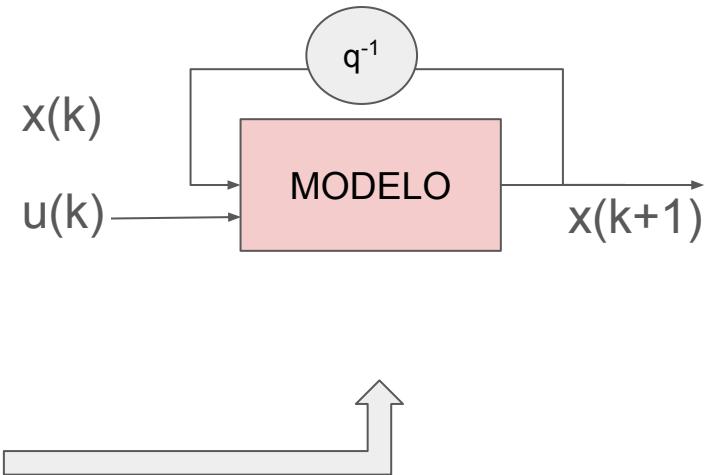
- La **identificación de sistemas** es la rama del modelado y la simulación que se ocupa de construir modelos matemáticos de sistemas dinámicos **a partir de datos experimentales**.
- A diferencia del modelado puramente **físico** o de **primeros principios**, en identificación partimos de mediciones de entradas y salidas, y tratamos de ajustar un modelo que reproduzca, con cierta precisión, el comportamiento observado.

1. Introducción

Entrenamiento



Inferencia



Índice

1. Introducción
2. **Fundamentos de la identificación de sistemas a partir de datos**
3. Modelos paramétricos vs. no paramétricos
4. Métodos clásicos
 - a. Método de mínimos cuadrados (LS)
 - b. Modelos ARX
 - c. Modelos ARMAX
 - d. Recursividad y mínimos cuadrados
5. Introducción a redes neuronales y modelos híbridos
 - a. Redes neuronales para identificación
 - b. Modelos híbridos: físicos + datos
6. Evaluación de desempeño del modelo

2. Fundamentos de la identificación

Consideremos un sistema dinámico (desconocido) que recibe una señal de entrada $u(t)$ y produce una salida $x(t)$. Según el contexto, trabajaremos en tiempo continuo o discreto. Para identificación por ordenador suele ser más práctico el tiempo discreto, donde tomamos muestras:

$$u(k) = u(kT_s), \quad x(k) = x(kT_s),$$

siendo T_s el periodo de muestreo y $k \in \mathbb{Z}$ el índice de tiempo discreto.

2. Fundamentos de la identificación

- **Objetivo básico**

El objetivo de la identificación es encontrar un modelo M en alguna clase de modelos (lineales, no lineales, estocásticos, etc.) tal que, dado $u(k)$, el modelo genere una salida $\hat{x}(k)$ que aproxime bien a la salida real $x(k)$:

$$\hat{x}(k) \approx x(k), \quad \forall k \text{ en el conjunto de interés.}$$

2. Fundamentos de la identificación

- **Objetivo básico**

Típicamente, un modelo se describe mediante una estructura con parámetros desconocidos θ :

$$M : \quad \hat{x}(k) = f(u(k), u(k-1), \dots, x(k-1), \dots; \theta),$$

y el problema de identificación se convierte en estimar θ a partir de los datos medidos:

$$\{u(k), x(k)\}_{k=1}^N.$$

2. Fundamentos de la identificación

Según el grado de conocimiento físico previo del sistema, se habla de:

- **Modelo de caja blanca (white-box)**: basado casi exclusivamente en leyes físicas conocidas; pocos parámetros desconocidos.
- **Modelo de caja gris (grey-box)**: se conoce parcialmente la estructura física, pero se dejan algunos términos o parámetros a estimar a partir de datos.
- **Modelo de caja negra (black-box)**: se desconoce la estructura interna del sistema; se usa una clase de modelos muy flexible (ej.\ series de Volterra, redes neuronales, etc.) y se determinan los parámetros sólo a partir de datos.

2. Fundamentos de la identificación

El procedimiento típico de identificación comprende las siguientes fases:

1. **Diseño del experimento y adquisición de datos.** Elegir cómo excitar al sistema (tipo de entrada), duración del experimento, muestreo, etc.
2. **Selección de la estructura del modelo.** Elegir una familia de modelos: ARX, ARMAX, modelos de espacio de estados, redes neuronales, etc.
3. **Estimación de parámetros.** Ajustar los parámetros theta minimizando algún criterio de error, típicamente basado en mínimos cuadrados o máxima verosimilitud.
4. **Validación del modelo.** Verificar si el modelo reproduce adecuadamente datos diferentes a los de entrenamiento, analizando residuales y criterios de calidad.
5. **Refinamiento.** Si el modelo no es satisfactorio, se revisan la estructura, los órdenes, la calidad de los datos o el tipo de excitación, y se repite el proceso.

2. Fundamentos de la identificación

Supongamos que el sistema real es discretizado y descrito por:

$$x(k+1) = a x(k) + b u(k) + e(k),$$

donde $e(k)$ es un ruido de medida (no conocido), a y b son parámetros constantes desconocidos. Dado un conjunto de datos $\{u(k), x(k)\}$, el problema de identificación consiste en estimar \hat{a} y \hat{b} tales que el modelo

$$\hat{x}(k+1) = \hat{a} x(k) + \hat{b} u(k)$$

reproduzca lo mejor posible las observaciones $x(k+1)$.

Índice

1. Introducción
2. Fundamentos de la identificación de sistemas a partir de datos
- 3. Modelos paramétricos vs. no paramétricos**
4. Métodos clásicos
 - a. Método de mínimos cuadrados (LS)
 - b. Modelos ARX
 - c. Modelos ARMAX
 - d. Recursividad y mínimos cuadrados
5. Introducción a redes neuronales y modelos híbridos
 - a. Redes neuronales para identificación
 - b. Modelos híbridos: físicos + datos
6. Evaluación de desempeño del modelo

3. Modelos paramétricos vs. no paramétricos

- **Modelos paramétricos**

En un modelo paramétrico se asume de antemano una *estructura funcional* para la relación entrada-salida, que depende de un número finito de parámetros. Ejemplos típicos:

- Modelos lineales invariantes en el tiempo (LTI) en forma de función de transferencia:

$$G(q^{-1}; \theta) = \frac{B(q^{-1}; \theta)}{A(q^{-1}; \theta)} = \frac{b_0 + b_1 q^{-1} + \cdots + b_{n_b} q^{-n_b}}{1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a}}.$$

3. Modelos paramétricos vs. no paramétricos

- **Modelos paramétricos**

- Modelos ARX/ARMAX, donde la dinámica del sistema y del ruido se describen mediante polinomios en q^{-1} .
- Modelos de espacio de estados discretos:

$$x(k+1) = Ax(k) + Bu(k), \quad y(k) = Cx(k) + Du(k).$$

3. Modelos paramétricos vs. no paramétricos

- **Modelos NO paramétricos**

En los modelos no paramétricos no se impone una estructura de modelo de orden finito a priori, sino que se trata de estimar directamente ciertas características del sistema a partir de los datos:

- **Respuesta al impulso:** estimar la secuencia $\{g(k)\}$ que relaciona entrada y salida mediante

$$x(k) = \sum_{i=0}^{\infty} g(i) u(k - i) + e(k).$$

3. Modelos paramétricos vs. no paramétricos

- **Modelos NO paramétricos**

- **Función de transferencia en frecuencia:** estimar $G(e^{j\omega})$ mediante técnicas de análisis espectral.
- **Espectros de potencia:** estimación de la densidad espectral $S_{yy}(\omega)$ y la coherencia entre u y x .

Índice

1. Introducción
2. Fundamentos de la identificación de sistemas a partir de datos
3. Modelos paramétricos vs. no paramétricos
- 4. Métodos clásicos**
 - a. **Método de mínimos cuadrados (LS)**
 - b. Modelos ARX
 - c. Modelos ARMAX
 - d. Recursividad y mínimos cuadrados
5. Introducción a redes neuronales y modelos híbridos
 - a. Redes neuronales para identificación
 - b. Modelos híbridos: físicos + datos
6. Evaluación de desempeño del modelo

4.a. Métodos clásicos: Mínimos Cuadrados

El **método de mínimos cuadrados** es una técnica fundamental para estimar parámetros en modelos lineales respecto a los parámetros.

Consideremos el modelo lineal en parámetros:

$$x(k) = \varphi^\top(k) \theta + e(k),$$

donde:

- $\theta \in \mathbb{R}^p$ es el vector de parámetros desconocidos,
- $\varphi(k) \in \mathbb{R}^p$ es el vector de regresores (compuesto de entradas y salidas pasadas),
- $e(k)$ es el error de modelado/ruido.

4.a. Métodos clásicos: Mínimos Cuadrados

Acumulando N muestras:

$$X = \Phi\theta + E,$$

donde

$$X = \begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(N) \end{bmatrix}, \quad \Phi = \begin{bmatrix} \varphi^\top(1) \\ \varphi^\top(2) \\ \vdots \\ \varphi^\top(N) \end{bmatrix}, \quad E = \begin{bmatrix} e(1) \\ e(2) \\ \vdots \\ e(N) \end{bmatrix}.$$

4.a. Métodos clásicos: Mínimos Cuadrados

El método de mínimos cuadrados busca $\hat{\theta}$ que minimice la suma de cuadrados del error:

$$J(\theta) = \sum_{k=1}^N (x(k) - \varphi^\top(k)\theta)^2 = \|X - \Phi\theta\|^2.$$

Derivando e igualando a cero se obtiene la solución LS (si $\Phi^\top\Phi$ es invertible):

$$\hat{\theta}_{\text{LS}} = (\Phi^\top\Phi)^{-1}\Phi^\top X.$$

4.a. Métodos clásicos: Mínimos Cuadrados

Ejemplo: estimación de un sistema de primer orden Retomemos el ejemplo:

$$x(k+1) = ax(k) + bu(k) + e(k).$$

Podemos escribir:

$$x(k+1) = \underbrace{\begin{bmatrix} x(k) & u(k) \end{bmatrix}}_{\varphi^\top(k)} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\theta} + e(k).$$

Con datos de $k = 0$ a $N - 1$, se construye Φ con filas $\varphi^\top(k)$ y X con elementos $x(k+1)$, y se calcula $\hat{\theta}$ usando la fórmula LS.

Índice

1. Introducción
2. Fundamentos de la identificación de sistemas a partir de datos
3. Modelos paramétricos vs. no paramétricos
- 4. Métodos clásicos**
 - a. Método de mínimos cuadrados (LS)
 - b. Modelos ARX**
 - c. Modelos ARMAX
 - d. Recursividad y mínimos cuadrados
5. Introducción a redes neuronales y modelos híbridos
 - a. Redes neuronales para identificación
 - b. Modelos híbridos: físicos + datos
6. Evaluación de desempeño del modelo

4.b. Métodos clásicos: Modelos ARX

Un modelo ARX (*AutoRegresivo con entrada eXógena*) describe la salida de un sistema como combinación lineal de salidas pasadas y entradas pasadas:

$$A(q^{-1})x(k) = B(q^{-1})u(k) + e(k),$$

donde

$$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} + \cdots + a_{n_a}q^{-n_a},$$

$$B(q^{-1}) = b_1q^{-1} + b_2q^{-2} + \cdots + b_{n_b}q^{-n_b},$$

y $e(k)$ es un término de error (ruido blanco idealmente).

Desarrollando la ecuación:

$$x(k) + a_1x(k-1) + \cdots + a_{n_a}x(k-n_a) = b_1u(k-1) + \cdots + b_{n_b}u(k-n_b) + e(k).$$

4.b. Métodos clásicos: Modelos ARX

Esto puede reescribirse en forma lineal en los parámetros:

$$x(k) = -a_1x(k-1) - \cdots - a_{n_a}x(k-n_a) + b_1u(k-1) + \cdots + b_{n_b}u(k-n_b) + e(k).$$

Definiendo el vector de regresores:

$$\varphi^\top(k) = \begin{bmatrix} -x(k-1) & \dots & -x(k-n_a) & u(k-1) & \dots & u(k-n_b) \end{bmatrix},$$

4.b. Métodos clásicos: Modelos ARX

y el vector de parámetros:

$$\theta^\top = \begin{bmatrix} a_1 & \dots & a_{n_a} & b_1 & \dots & b_{n_b} \end{bmatrix},$$

obtenemos:

$$x(k) = \varphi^\top(k) \theta + e(k),$$

y podemos estimar θ por mínimos cuadrados.

Los modelos ARX son sencillos de identificar y muy usados como punto de partida aunque su tratamiento del ruido puede ser limitado.

4.b. Métodos clásicos: Modelos ARX

- **Inferencia:**

Supongamos que hemos identificado un modelo ARX de la forma:

$$x(k) = a_1x(k-1) + a_2x(k-2) + \dots + b_1u(k-1) + b_2u(k-2).$$

Una vez determinados los parámetros \hat{a}_i, \hat{b}_j , la **predicción a un paso** es:

$$\hat{x}(k+1) = \hat{a}_1x(k) + \hat{a}_2x(k-1) + \dots + \hat{b}_1u(k) + \hat{b}_2u(k-1).$$

4.b. Métodos clásicos: Modelos ARX

- **Inferencia:**

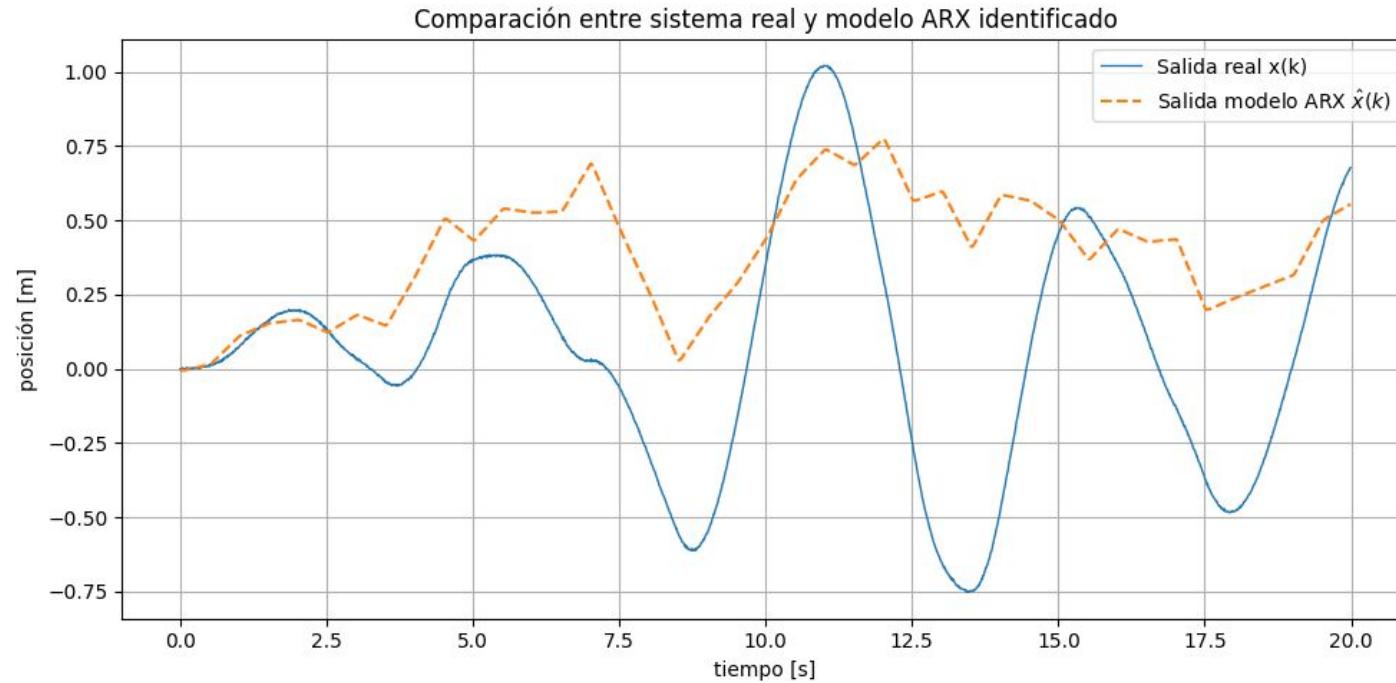
Si se desea predecir más lejos en el tiempo, se utiliza recursivamente la propia predicción como si fuera el dato real:

$$\hat{x}(k+2) = \hat{a}_1\hat{x}(k+1) + \hat{a}_2x(k) + \dots + \hat{b}_1u(k+1) + \hat{b}_2u(k).$$

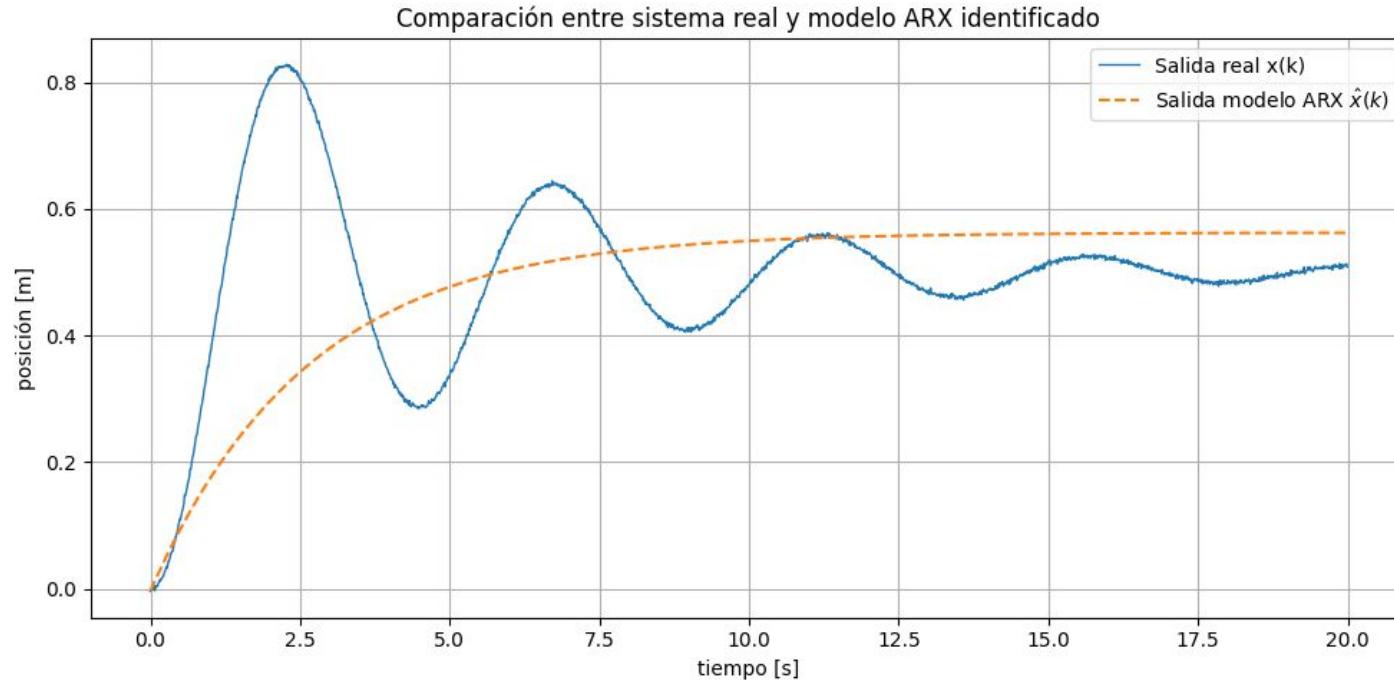
$$\hat{x}(k+3) = \hat{a}_1\hat{x}(k+2) + \hat{a}_2\hat{x}(k+1) + \dots$$

Esto permite **simular** el sistema identificado a futuro.

4.b. Métodos clásicos: Modelos ARX



4.b. Métodos clásicos: Modelos ARX



Índice

1. Introducción
2. Fundamentos de la identificación de sistemas a partir de datos
3. Modelos paramétricos vs. no paramétricos
- 4. Métodos clásicos**
 - a. Método de mínimos cuadrados (LS)
 - b. Modelos ARX
 - c. Modelos ARMAX**
 - d. Recursividad y mínimos cuadrados
5. Introducción a redes neuronales y modelos híbridos
 - a. Redes neuronales para identificación
 - b. Modelos híbridos: físicos + datos
6. Evaluación de desempeño del modelo

4.c. Métodos clásicos: Modelos ARMAX

Los modelos ARMAX (*AutoRegresivo–Media Móvil con entrada eXógena*) incorporan un modelo más rico del ruido:

$$A(q^{-1})x(k) = B(q^{-1})u(k) + C(q^{-1})e(k),$$

donde, además de A y B , se introduce:

$$C(q^{-1}) = 1 + c_1q^{-1} + \cdots + c_{n_c}q^{-n_c}.$$

Aquí el ruido $e(k)$ entra filtrado por $C(q^{-1})$, lo que permite modelar correlaciones temporales en el ruido (es decir, ruido con estructura de media móvil).

4.c. Métodos clásicos: Modelos ARMAX

A diferencia del ARX, la presencia de $e(k)$ filtrado hace que el modelo ya no sea lineal en los parámetros de forma directa, por lo que la identificación de ARMAX suele requerir métodos de optimización iterativos (p. ej. máxima verosimilitud, gradiente, Gauss-Newton, etc.).

Índice

1. Introducción
2. Fundamentos de la identificación de sistemas a partir de datos
3. Modelos paramétricos vs. no paramétricos
- 4. Métodos clásicos**
 - a. Método de mínimos cuadrados (LS)
 - b. Modelos ARX
 - c. Modelos ARMAX
 - d. Recursividad y mínimos cuadrados**
5. Introducción a redes neuronales y modelos híbridos
 - a. Redes neuronales para identificación
 - b. Modelos híbridos: físicos + datos
6. Evaluación de desempeño del modelo

4.d. Recursividad y mínimos cuadrados

En aplicaciones en tiempo real, donde llegan datos continuamente, puede interesar actualizar la estimación de parámetros en línea. El **método de mínimos cuadrados recursivos** (RLS) permite actualizar $\hat{\theta}(k)$ al llegar cada nueva muestra $(\varphi(k), x(k))$.

La actualización típica es:

$$\begin{aligned}
 K(k) &= \frac{P(k-1)\varphi(k)}{\lambda + \varphi^\top(k)P(k-1)\varphi(k)}, \\
 \hat{\theta}(k) &= \hat{\theta}(k-1) + K(k)[x(k) - \varphi^\top(k)\hat{\theta}(k-1)], \\
 P(k) &= \frac{1}{\lambda} \left[P(k-1) - K(k)\varphi^\top(k)P(k-1) \right],
 \end{aligned}$$

4.d. Recursividad y mínimos cuadrados

donde:

- $P(k)$ es la matriz de covarianza asociada a la incertidumbre en los parámetros,
- $K(k)$ es la ganancia de adaptación,
- $\lambda \in (0, 1]$ es un factor de olvido que permite dar más peso a datos recientes.

4.d. Recursividad y mínimos cuadrados

En aplicaciones en tiempo real, donde llegan datos continuamente, puede interesar actualizar la estimación de parámetros en línea. El **método de mínimos cuadrados recursivos** (RLS) permite actualizar $\hat{\theta}(k)$ al llegar cada nueva muestra $(\varphi(k), x(k))$.

La actualización típica es:

$$\begin{aligned} K(k) &= \frac{P(k-1)\varphi(k)}{\lambda + \varphi^\top(k)P(k-1)\varphi(k)}, \\ \hat{\theta}(k) &= \hat{\theta}(k-1) + K(k)[x(k) - \varphi^\top(k)\hat{\theta}(k-1)], \\ P(k) &= \frac{1}{\lambda} \left[P(k-1) - K(k)\varphi^\top(k)P(k-1) \right], \end{aligned}$$

Índice

1. Introducción
2. Fundamentos de la identificación de sistemas a partir de datos
3. Modelos paramétricos vs. no paramétricos
4. Métodos clásicos
 - a. Método de mínimos cuadrados (LS)
 - b. Modelos ARX
 - c. Modelos ARMAX
 - d. Recursividad y mínimos cuadrados
- 5. Introducción a redes neuronales y modelos híbridos**
 - a. Redes neuronales para identificación
 - b. Modelos híbridos: físicos + datos
6. Evaluación de desempeño del modelo

5. Introducción a redes neuronales

- Cuando los sistemas son fuertemente no lineales o presentan fenómenos difíciles de capturar con modelos lineales tradicionales (ARX/ARMAX), se recurre a técnicas de **aprendizaje automático**, especialmente redes neuronales.
- Las **redes neuronales artificiales (RNA)** pueden aproximar funciones no lineales complejas. Para identificación, se suelen usar **arquitecturas que modelan la dependencia temporal**.

Índice

1. Introducción
2. Fundamentos de la identificación de sistemas a partir de datos
3. Modelos paramétricos vs. no paramétricos
4. Métodos clásicos
 - a. Método de mínimos cuadrados (LS)
 - b. Modelos ARX
 - c. Modelos ARMAX
 - d. Recursividad y mínimos cuadrados
- 5. Introducción a redes neuronales y modelos híbridos**
 - a. **Redes neuronales para identificación**
 - b. Modelos híbridos: físicos + datos
6. Evaluación de desempeño del modelo

5.a. Redes neuronales para identificación

- Modelos **NARX** (Nonlinear AutoRegressive with eXogenous input):

$$x(k) = F(x(k-1), \dots, x(k-n_y), u(k-1), \dots, u(k-n_u)) + e(k),$$

donde $F(\cdot)$ se implementa mediante una red neuronal (por ejemplo, un perceptrón multicapa).

5.a. Redes neuronales para identificación

- **Redes recurrentes (RNN, LSTM):** incorporan memoria interna para capturar dependencias de largo plazo.

En el caso de un perceptrón multicapa, se tiene típicamente:

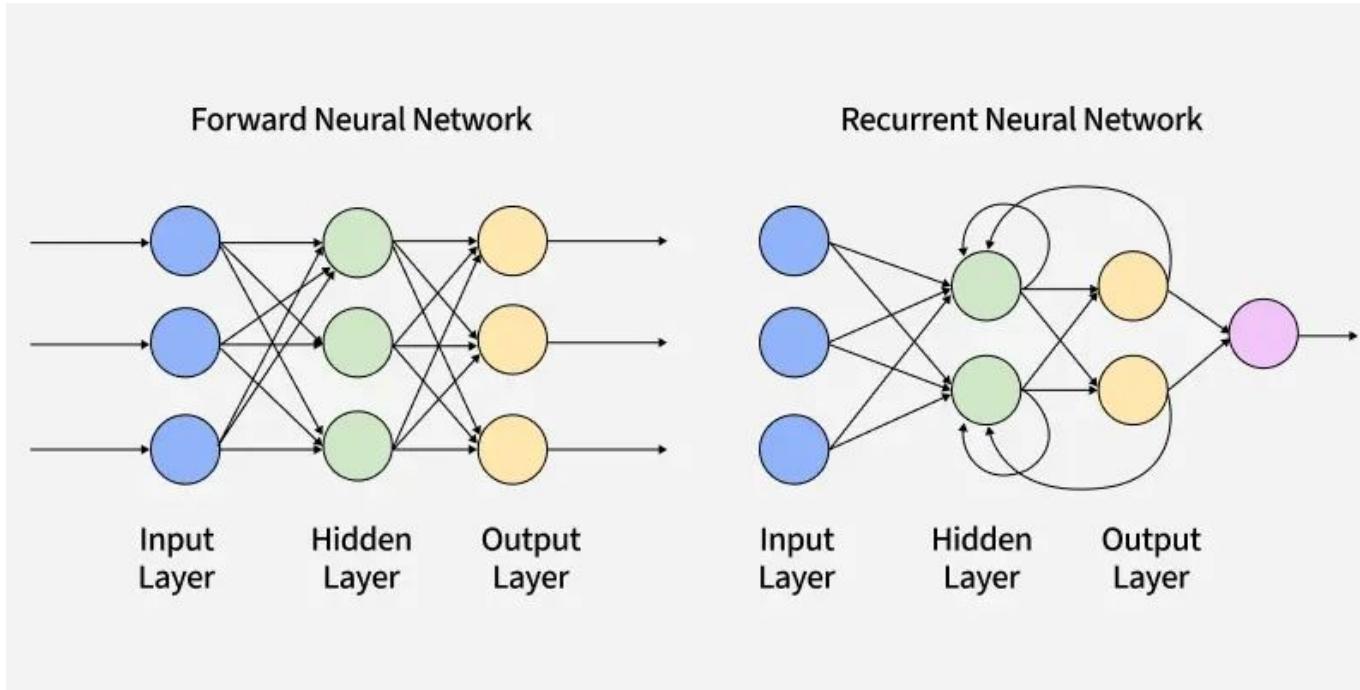
$$\hat{x}(k) = f_{\text{NN}}(\varphi(k); \theta),$$

donde $\varphi(k)$ incluye salidas y entradas pasadas, y θ agrupa todos los pesos y sesgos de la red. La estimación de θ se realiza minimizando un coste (p. ej. suma de cuadrados) mediante algoritmos como *backpropagation*.

Índice

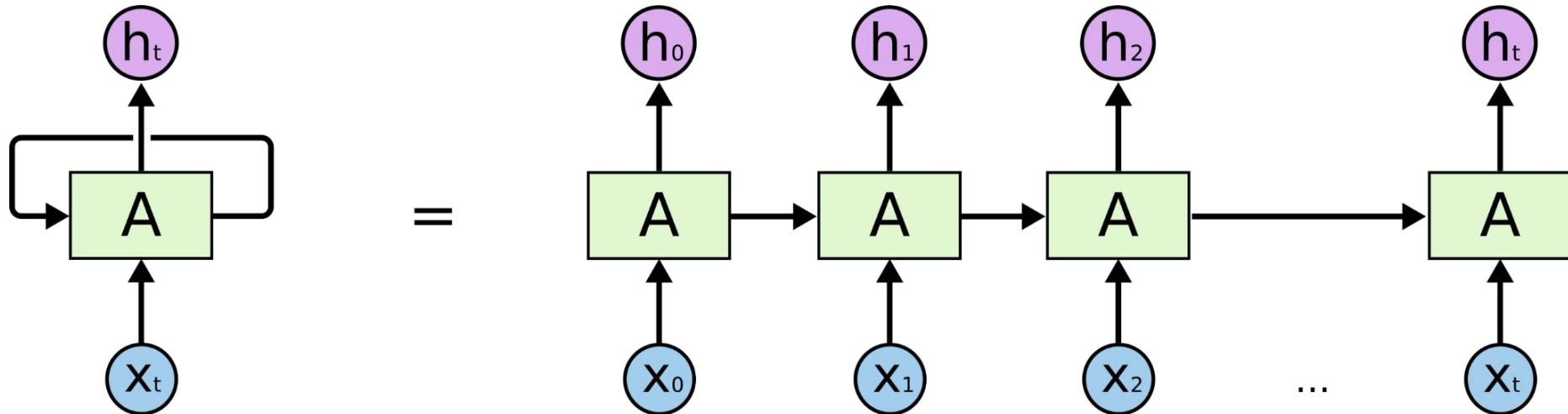
1. Introducción
2. Fundamentos de la identificación de sistemas a partir de datos
3. Modelos paramétricos vs. no paramétricos
4. Métodos clásicos
 - a. Método de mínimos cuadrados (LS)
 - b. Modelos ARX
 - c. Modelos ARMAX
 - d. Recursividad y mínimos cuadrados
- 5. Introducción a redes neuronales y modelos híbridos**
 - a. **Redes neuronales para identificación (RNN)**
 - b. Modelos híbridos: físicos + datos
6. Evaluación de desempeño del modelo

5.a. Recurrent Neural Networks (RNN)



5.a. Recurrent Neural Networks (RNN)

- Es como si la red se “desenrollara” en el tiempo



5.a. Recurrent Neural Networks (RNN)

- Las **Redes Neuronales Recurrentes (RNN)** constituyen una extensión natural de los modelos de identificación basados en series temporales.
- A diferencia de los modelos ARX, donde la memoria se introduce explícitamente mediante retardos, una RNN posee un **estado interno recurrente** que se actualiza en cada instante temporal.
- Esto las hace especialmente adecuadas para modelar dinámicas no lineales complejas.

5.a. Recurrent Neural Networks (RNN)

$$h(k) = \sigma(W_h h(k-1) + W_u u(k) + W_x x(k)),$$

$$\hat{x}(k+1) = W_o h(k),$$

donde:

- $h(k)$ es el estado interno recurrente,
- $\sigma(\cdot)$ es una función de activación no lineal (típicamente tanh o ReLU),
- W_h, W_u, W_x, W_o son parámetros aprendidos mediante optimización.

5.a. Recurrent Neural Networks (RNN)

En una RNN, la memoria del sistema está codificada en el vector $h(k)$, que actúa como una aproximación al estado dinámico real del sistema. Esto permite aproximar funciones dinámicas no lineales del tipo:

$$x(k+1) = F(x(k), x(k-1), \dots, u(k), u(k-1), \dots),$$

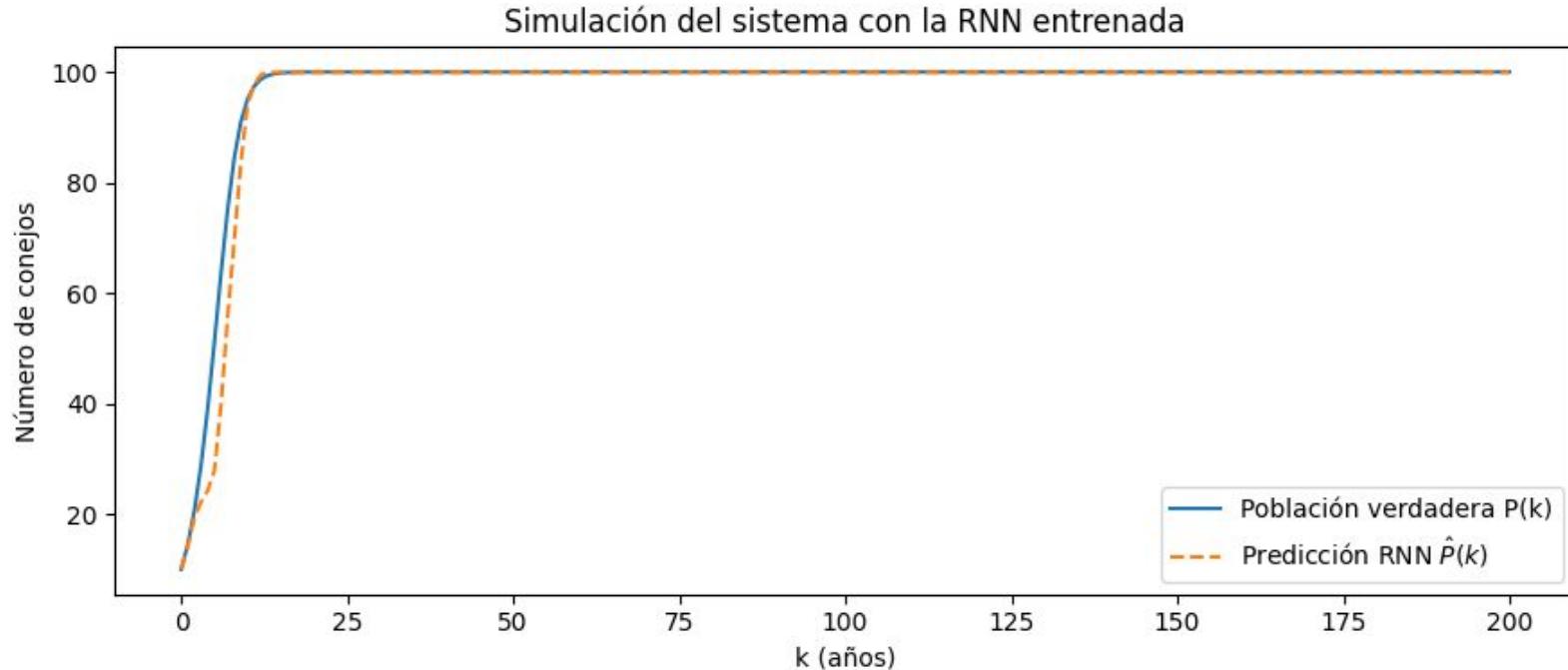
sin necesidad de elegir manualmente cuántos retardos incluir, como ocurre en los modelos ARX o NARX clásicos.

5.a. Recurrent Neural Networks (RNN)

Las RNN suelen funcionar bien para:

- Sistemas suaves con dependencias temporales cortas.
- Dinámica no lineal moderada.
- Señales de entrada ricas.
- Predicción a uno o varios pasos vista como un problema de modelado secuencial.

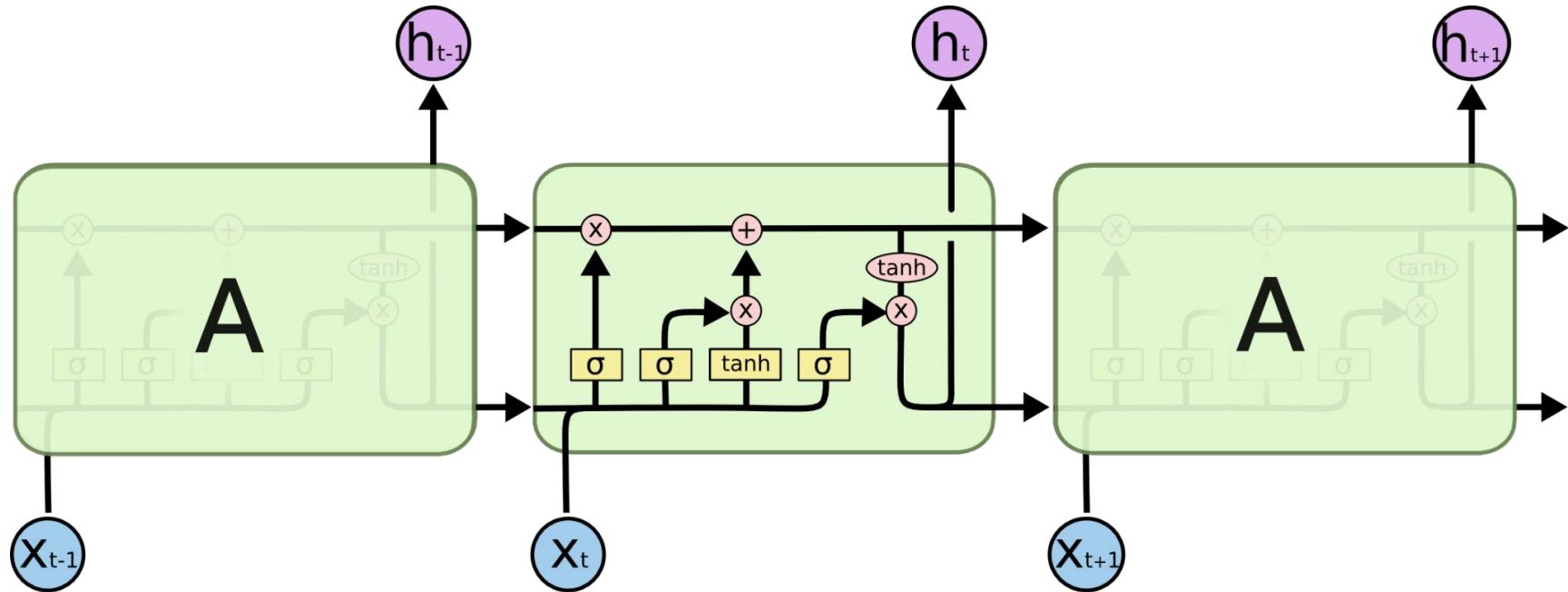
5.a. Recurrent Neural Networks (RNN)



Índice

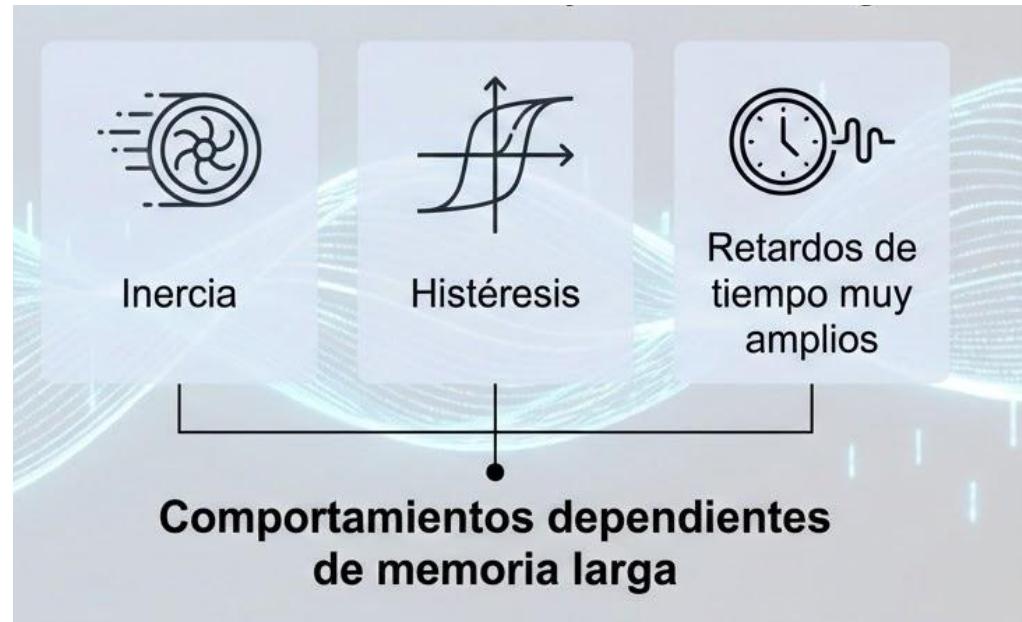
1. Introducción
2. Fundamentos de la identificación de sistemas a partir de datos
3. Modelos paramétricos vs. no paramétricos
4. Métodos clásicos
 - a. Método de mínimos cuadrados (LS)
 - b. Modelos ARX
 - c. Modelos ARMAX
 - d. Recursividad y mínimos cuadrados
- 5. Introducción a redes neuronales y modelos híbridos**
 - a. **Redes neuronales para identificación (LSTM)**
 - b. Modelos híbridos: físicos + datos
6. Evaluación de desempeño del modelo

5.a. Long Short-Term Memory (LSTM)



5.a. Long Short-Term Memory (LSTM)

- Las redes **LSTM (Long Short-Term Memory)** se diseñaron específicamente para resolver el problema de los gradientes vanishing/exploding de las RNN estándar.
- En dinámica de sistemas, las LSTM permiten aprender comportamientos dependientes de **memoria larga**, como inercia, histéresis o retardos de tiempo muy amplios.



5.a. Long Short-Term Memory (LSTM)

Las LSTM son modelos extremadamente potentes para:

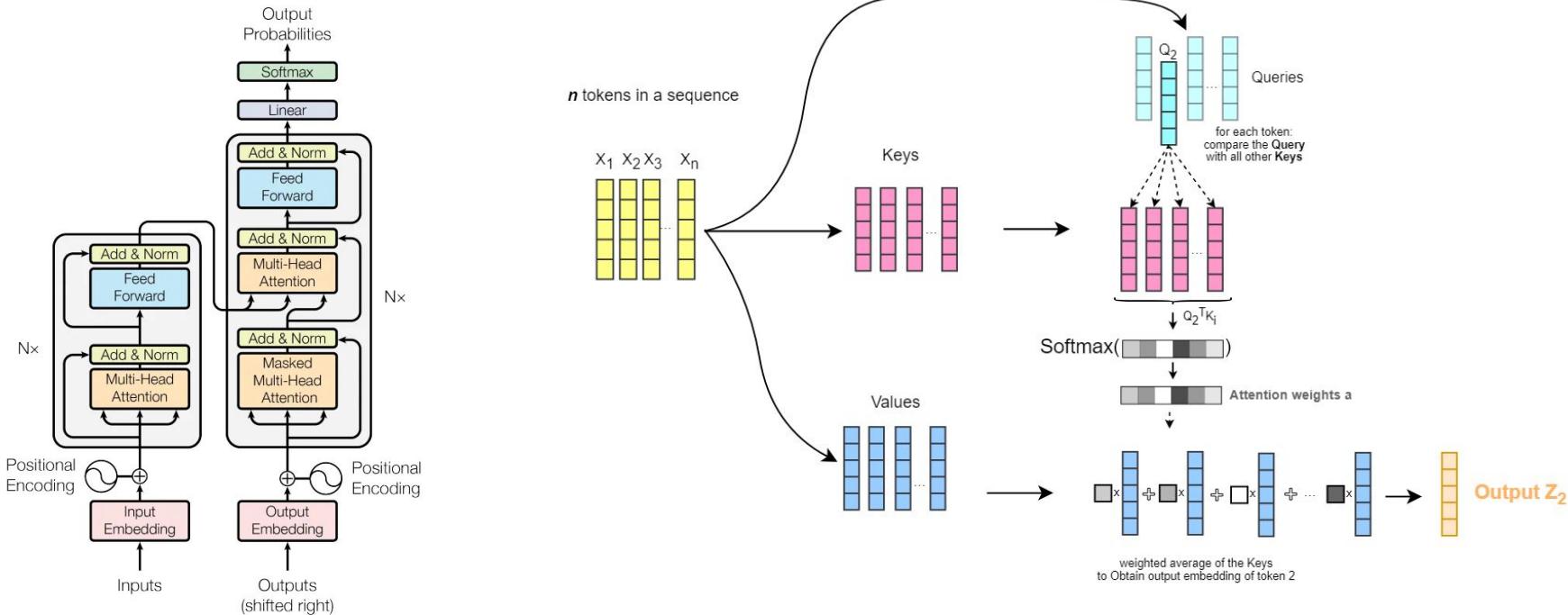
- Sistemas mecánicos con histéresis,
- Sistemas con retardos largos,
- Sistemas biológicos o ecológicos con dinámica lenta,
- Series temporales ruidosas donde la estructura se acumula lentamente.

Además, las LSTM son muy utilizadas como modelo base en estructuras **NARX no lineales**, permitiendo capturar dinámicas que serían imposibles de representar con modelos ARX clásicos de orden razonable.

Índice

1. Introducción
2. Fundamentos de la identificación de sistemas a partir de datos
3. Modelos paramétricos vs. no paramétricos
4. Métodos clásicos
 - a. Método de mínimos cuadrados (LS)
 - b. Modelos ARX
 - c. Modelos ARMAX
 - d. Recursividad y mínimos cuadrados
- 5. Introducción a redes neuronales y modelos híbridos**
 - a. **Redes neuronales para identificación (Transformers)**
 - b. Modelos híbridos: físicos + datos
6. Evaluación de desempeño del modelo

5.a. Transformers (Attention / Self Attention)



5.a. Transformers (Attention / Self Attention)

Los **Transformers** han revolucionado el modelado secuencial en los últimos años, desplazando en muchos ámbitos a las RNN y LSTM. Su característica distintiva es el mecanismo de **self-attention**, que permite relacionar directamente todos los instantes temporales entre sí sin recurrencia.

- Aunque nacieron para lenguaje natural, hoy en día se aplican cada vez más a:
 - **Identificación de sistemas.**
 - Simulación de dinámica compleja.
 - Predicción multihorizonte.
 - Sistemas multiagente.
 - Predicción de series muy largas.

5.a. Transformers (Attention / Self Attention)

Los Transformers son especialmente potentes para:

- Sistemas complejos con alta no linealidad.
- Series temporales largas donde la memoria de largo plazo es crucial.
- Dinámica con múltiples escalas temporales.
- Estimación conjunta de entrada y estado.
- Reconstrucción de series incompletas.

Índice

1. Introducción
2. Fundamentos de la identificación de sistemas a partir de datos
3. Modelos paramétricos vs. no paramétricos
4. Métodos clásicos
 - a. Método de mínimos cuadrados (LS)
 - b. Modelos ARX
 - c. Modelos ARMAX
 - d. Recursividad y mínimos cuadrados
- 5. Introducción a redes neuronales y modelos híbridos**
 - a. **Redes neuronales para identificación**
 - b. Modelos híbridos: físicos + datos
6. Evaluación de desempeño del modelo

5.a. Redes neuronales para identificación

Ventajas y desventajas:

- **Ventajas:** Capacidad para modelar no linealidades complejas, flexibilidad, buen desempeño cuando se dispone de muchos datos.
- **Desventajas:** Necesitan grandes conjuntos de datos, riesgo de sobreajuste, interpretabilidad limitada, selección delicada de arquitecturas (número de capas, neuronas, etc.).

5.a. Redes neuronales para identificación

Implementación (librerías):



PyTorch

Utilizada en la práctica 6 (parte 2)



TensorFlow



Keras

Índice

1. Introducción
2. Fundamentos de la identificación de sistemas a partir de datos
3. Modelos paramétricos vs. no paramétricos
4. Métodos clásicos
 - a. Método de mínimos cuadrados (LS)
 - b. Modelos ARX
 - c. Modelos ARMAX
 - d. Recursividad y mínimos cuadrados
- 5. Introducción a redes neuronales y modelos híbridos**
 - a. Redes neuronales para identificación
 - b. Modelos híbridos: físicos + datos**
6. Evaluación de desempeño del modelo

5.b. Modelos híbridos

Los **modelos híbridos** combinan:

- un modelo de *primeros principios* (físico),
- con un componente *aprendido a partir de datos* (por ejemplo, una red neuronal).

Una forma común de modelo híbrido es:

$$x(k) = x_{\text{físico}}(k; \theta_f) + x_{\text{NN}}(k; \theta_n) + e(k),$$

5.b. Modelos híbridos

donde:

- $x_{\text{físico}}(k; \theta_f)$ refleja la parte del sistema bien entendida (por ejemplo, ecuaciones de balance de masas, leyes de Newton, etc.),
- $x_{\text{NN}}(k; \theta_n)$ modela efectos desconocidos (no linealidades, fricciones complejas, retardos, etc.).

Otra variante es usar la red neuronal para modelar solo el *término de error* o incertidumbre de un modelo físico:

$$e_{\text{modelo}}(k) = x(k) - x_{\text{físico}}(k; \theta_f), \quad e_{\text{modelo}}(k) \approx f_{\text{NN}}(\cdot; \theta_n).$$

5.b. Modelos híbridos

Ventajas de los modelos híbridos:

- Aprovechan conocimiento físico previo, lo que reduce la cantidad de datos necesarios.
- Mejoran la interpretabilidad frente a un modelo puramente caja negra.
- Suelen generalizar mejor fuera de la región exacta donde se tomaron los datos.

Índice

1. Introducción
2. Fundamentos de la identificación de sistemas a partir de datos
3. Modelos paramétricos vs. no paramétricos
4. Métodos clásicos
 - a. Método de mínimos cuadrados (LS)
 - b. Modelos ARX
 - c. Modelos ARMAX
 - d. Recursividad y mínimos cuadrados
5. Introducción a redes neuronales y modelos híbridos
 - a. Redes neuronales para identificación
 - b. Modelos híbridos: físicos + datos
- 6. Evaluación de desempeño del modelo**

6. Evaluación de desempeño del modelo

Una vez identificado un modelo, es esencial evaluar su desempeño y verificar si es adecuado para el propósito (simulación, predicción, control, etc.).

Es habitual dividir los datos disponibles en (**IMPORTANTE**):

- **Datos de entrenamiento (training)**: usados para ajustar los parámetros del modelo.
- **Datos de validación (validation)**: usados para seleccionar hiperparámetros (orden del modelo, estructura, regularización) y evitar sobreajuste.
- **Datos de prueba (test)**: usados para estimar el desempeño final del modelo sobre datos no vistos.

Esta separación es especialmente importante cuando se usan modelos flexibles como redes neuronales.

6. Evaluación de desempeño del modelo

- **Índices de error**

Dado un conjunto de datos de evaluación $\{x(k), \hat{x}(k)\}$, se pueden usar distintos índices de error:

- **Error cuadrático medio (MSE):**

$$\text{MSE} = \frac{1}{N} \sum_{k=1}^N (x(k) - \hat{x}(k))^2.$$

- **Raíz del error cuadrático medio (RMSE):**

$$\text{RMSE} = \sqrt{\text{MSE}}.$$

6. Evaluación de desempeño del modelo

- Índices de error

- Error absoluto medio (MAE):

$$\text{MAE} = \frac{1}{N} \sum_{k=1}^N |x(k) - \hat{x}(k)|.$$

- Porcentaje de ajuste (fit):

$$\text{fit} = 100 \left(1 - \frac{\|x - \hat{x}\|}{\|x - \bar{x}\|} \right) \%,$$

donde \bar{x} es la media de x . Un valor de fit cercano al 100% indica un muy buen ajuste.

6. Evaluación de desempeño del modelo

Para comparar modelos con diferentes números de parámetros, no basta con ver el error: un modelo con más parámetros tiende a ajustar mejor los datos de entrenamiento, pero puede sobreajustar. Para penalizar la complejidad se usan criterios como:

- **Criterio de Información de Akaike (AIC):**

$$\text{AIC} = 2p + N \ln(\hat{\sigma}^2),$$

donde p es el número de parámetros del modelo y $\hat{\sigma}^2$ es una estimación de la varianza del error.

- **Criterio de Información Bayesiano (BIC):**

$$\text{BIC} = p \ln(N) + N \ln(\hat{\sigma}^2).$$

6. Evaluación de desempeño del modelo

- **Análisis de residuos**

Los **residuos** del modelo se definen como:

$$\varepsilon(k) = x(k) - \hat{x}(k).$$

6. Evaluación de desempeño del modelo

● Análisis de residuos

Para que el modelo sea adecuado, se espera que los residuos tengan las siguientes propiedades:

- Media aproximadamente cero: $\mathbb{E}[\varepsilon(k)] \approx 0$.
- No correlación temporal: la autocorrelación de $\varepsilon(k)$ debe ser cercana a cero para retardos distintos de cero.
- No correlación con la entrada $u(k)$: la correlación cruzada entre $u(k)$ y $\varepsilon(k)$ debe ser aproximadamente cero.

Si se observa estructura en la autocorrelación o correlación cruzada (picos significativos), indica que el modelo no ha capturado toda la dinámica o que hay errores de estructura.

6. Evaluación de desempeño del modelo

Validez frente al propósito:

La **validez** del modelo debe juzgarse en función del propósito para el cual se va a utilizar:

- Un modelo para **control** puede requerir precisión en un rango de frecuencias o en torno a ciertos puntos de operación.
- Un modelo para **simulación** puede necesitar buena precisión a largo plazo.
- Un modelo para **detección de fallos** puede centrarse en reproducir ciertos patrones anómalos.

Por ello, la evaluación no es solo numérica sino también contextual: un modelo muy complejo puede no ser necesario si uno más simple cumple el objetivo.

Temario

B1 - Sistemas Dinámicos

T1: Fundamentos del modelado de sistemas dinámicos

T2: Estabilidad, controlabilidad y observabilidad

T3: Paradigmas de simulación

B2 - Sistemas Complejos

T4: Lenguajes formales para modelos conceptuales

T5: Redes complejas y modelado estructural

B3 - Modelado con IA

T6: Identificación de sistemas

T7: Inteligencia artificial aplicada al modelado de sistemas

Tema 6: Identificación de sistemas

MODELOS COMPUTACIONALES Y SIMULACIÓN DE SISTEMAS
Curso 2025-2026



UNIVERSITAT D'ALACANT
UNIVERSIDAD DE ALICANTE
Escola Politècnica Superior
Escuela Politécnica Superior



Tema 7: Inteligencia artificial aplicada al modelado de sistemas

MODELOS COMPUTACIONALES Y SIMULACIÓN DE SISTEMAS

Curso 2025-2026



UNIVERSITAT D'ALACANT
UNIVERSIDAD DE ALICANTE
Escola Politècnica Superior
Escuela Politécnica Superior



Índice

1. Introducción
2. Localización: **Estimación de $x(t)$**
 - a. Odometría LiDAR
 - b. Geo-localización
3. Planificación y control: **Estimación de $u(t)$**
 - a. Planificación local
 - b. Planificación local mediante aprendizaje
4. Otros temas relacionados
 - a. Oxford Spires dataset
 - b. Proyecto Digiforest
 - c. Sapling-NeRF
5. Líneas de investigación abiertas

Índice

- 1. Introducción**
2. Localización: **Estimación de $x(t)$**
 - a. Odometría LiDAR
 - b. Geo-localización
3. Planificación y control: **Estimación de $u(t)$**
 - a. Planificación local
 - b. Planificación local mediante aprendizaje
4. Otros temas relacionados
 - a. Oxford Spires dataset
 - b. Proyecto Digiforest
 - c. Sapling-NeRF
5. Líneas de investigación abiertas

1. Introducción

- Vamos a ver diferentes aplicaciones de **Robótica Móvil**.
- Partimos de la representación en **espacio de estados**:

Lineal

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

No Lineal

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), t) \\ y(t) = g(x(t), u(t), t) \end{cases}$$

Índice

1. Introducción
2. **Localización: Estimación de $x(t)$**
 - a. Odometría LiDAR
 - b. Geo-localización
3. Planificación y control: **Estimación de $u(t)$**
 - a. Planificación local
 - b. Planificación local mediante aprendizaje
4. Otros temas relacionados
 - a. Oxford Spires dataset
 - b. Proyecto Digiforest
 - c. Sapling-NeRF
5. Líneas de investigación abiertas

2. Localización: Estimación de $x(t)$

- **Modelo de propagación:**

$$x_{k+1} = f(x_k, u_k) + w_k$$

$$f(x_k, u_k) = \begin{bmatrix} x_k + v_k \Delta t \cos \theta_k \\ y_k + v_k \Delta t \sin \theta_k \\ \theta_k + \omega_k \Delta t \end{bmatrix}$$

2. Localización: Estimación de $x(t)$

- **Modelo de observación:**

$$y_k = h(x_k) + v_k$$

$$\hat{y}_{k|k-1} = h(\hat{x}_{k|k-1})$$

$$\tilde{y}_k = y_k - \hat{y}_{k|k-1}$$

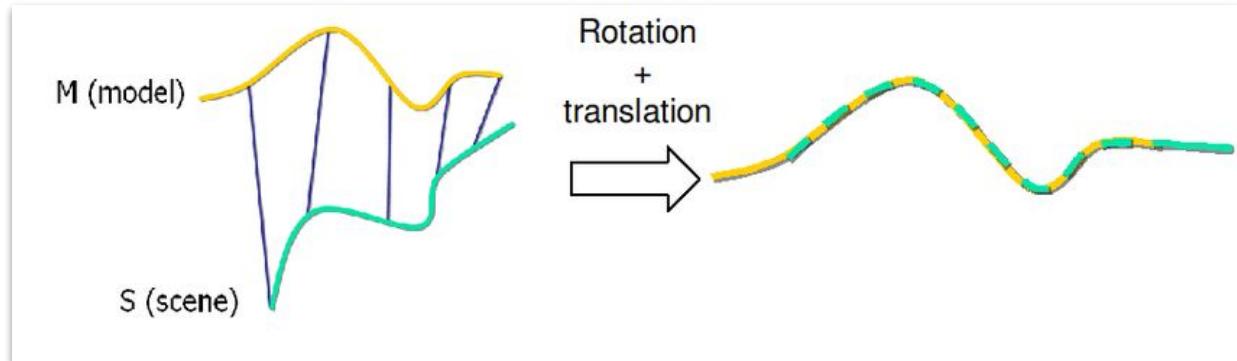
Índice

1. Introducción
2. Localización: Estimación de $x(t)$
 - a. Odometría LiDAR
 - b. Geo-localización
3. Planificación y control: Estimación de $u(t)$
 - a. Planificación local
 - b. Planificación local mediante aprendizaje
4. Otros temas relacionados
 - a. Oxford Spires dataset
 - b. Proyecto Digiforest
 - c. Sapling-NeRF
5. Líneas de investigación abiertas

2.a. Odometría LiDAR

- Con Iterative Closest Point y un LiDAR se puede obtener y_k

$$y_k = \begin{bmatrix} \hat{x}_k^{\text{lidar}} \\ \hat{y}_k^{\text{lidar}} \\ \hat{\theta}_k^{\text{lidar}} \end{bmatrix}$$



$$h(\hat{x}_k) = \hat{x}_k$$

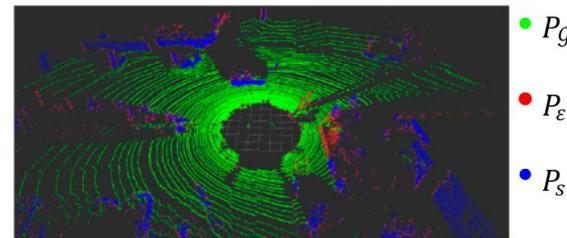
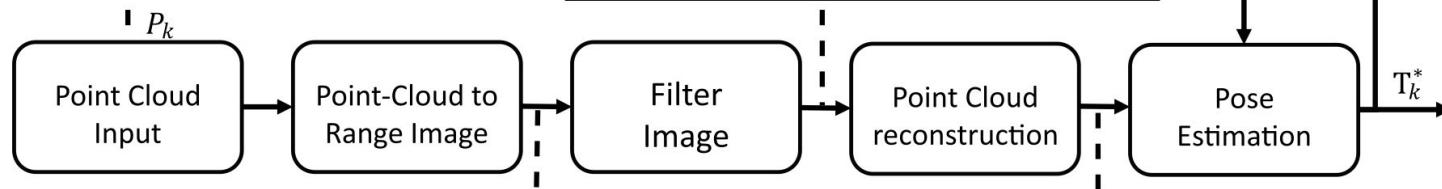
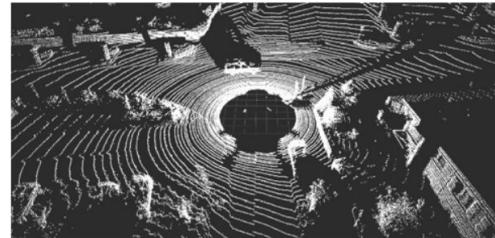
$$H_k = I$$



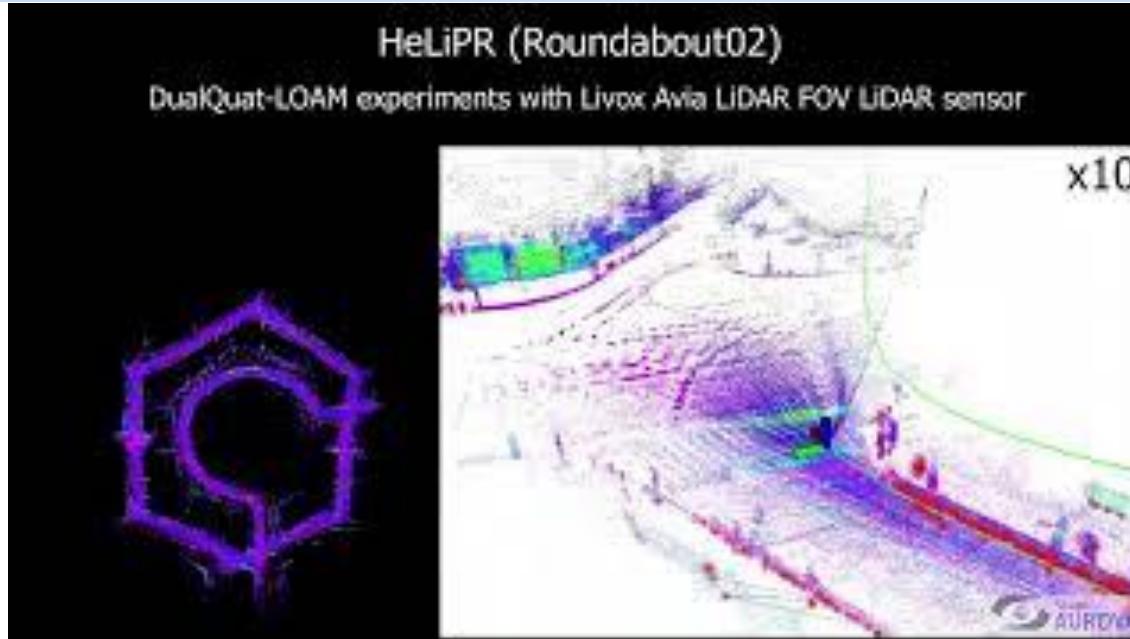
$$\tilde{y}_k = y_k - \hat{y}_{k|k-1}$$

$$\hat{y}_k = \hat{x}_k + v_k$$

2.a. Odometría LiDAR (caso real)



2.a. Odometría LiDAR (caso real)



<https://youtu.be/EDYU18tMmgI?si=Di5vVKBl1gdqWpV>

2.a. Odometría LiDAR (caso real)



Advanced Robotics

Taylor & Francis
Taylor & Francis Group

ISSN: 0169-1864 (Print) 1568-5535 (Online) Journal homepage: www.tandfonline.com/journals/tadr20

LiLO: lightweight and low-bias LiDAR odometry method based on spherical range image filtering

Edison P. Velasco-Sánchez, Miguel Ángel Muñoz-Bañón, Francisco A. Candelas, Santiago T. Puente & Fernando Torres

- Velasco-Sánchez, Edison P., et al. "LiLO: Lightweight and low-bias LiDAR odometry method based on spherical range image filtering." *Advanced Robotics* (2025): 1-17.

Índice

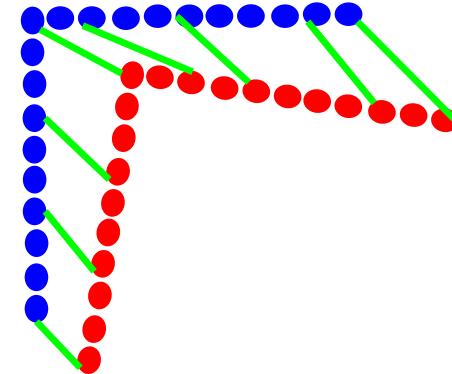
1. Introducción
2. **Localización: Estimación de $x(t)$**
 - a. Odometría LiDAR
 - b. Geo-localización**
3. Planificación y control: **Estimación de $u(t)$**
 - a. Planificación local
 - b. Planificación local mediante aprendizaje
4. Otros temas relacionados
 - a. Oxford Spires dataset
 - b. Proyecto Digiforest
 - c. Sapling-NeRF
5. Líneas de investigación abiertas

2.a. Localización

- y_k viene dado por el conjunto de **puntos del LiDAR**
- Se asocian puntos del sensor con **puntos de un mapa**

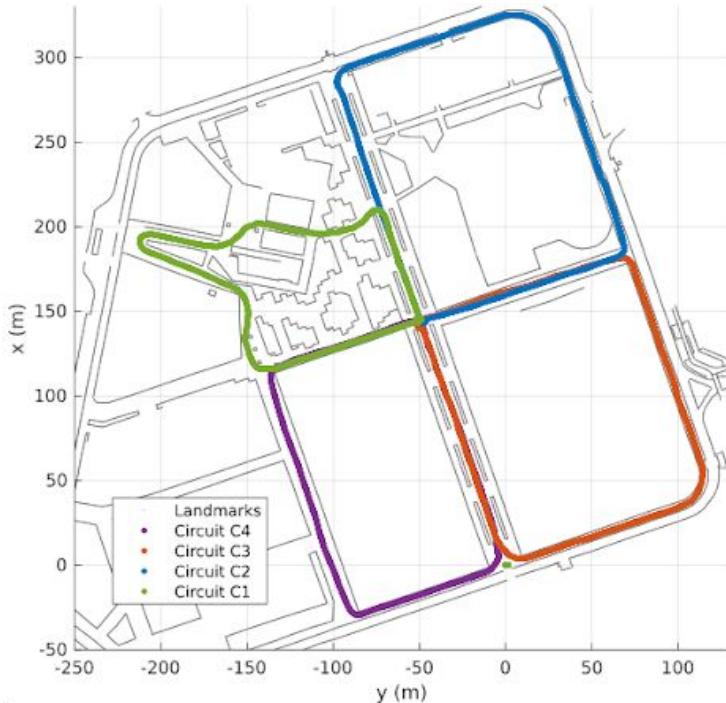
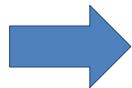
$$y_k = z_k = \begin{bmatrix} r_{k,1} \\ r_{k,2} \\ \vdots \\ r_{k,N} \end{bmatrix}$$

$$y_k = h(x_k) + v_k$$

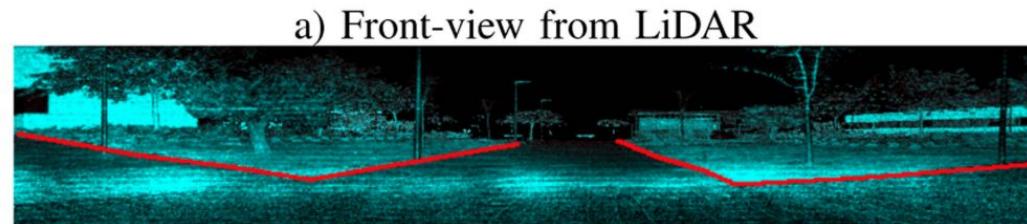


$$\tilde{y}_k = \textcircled{y}_k - \textcircled{\hat{y}}_{k|k-1}$$

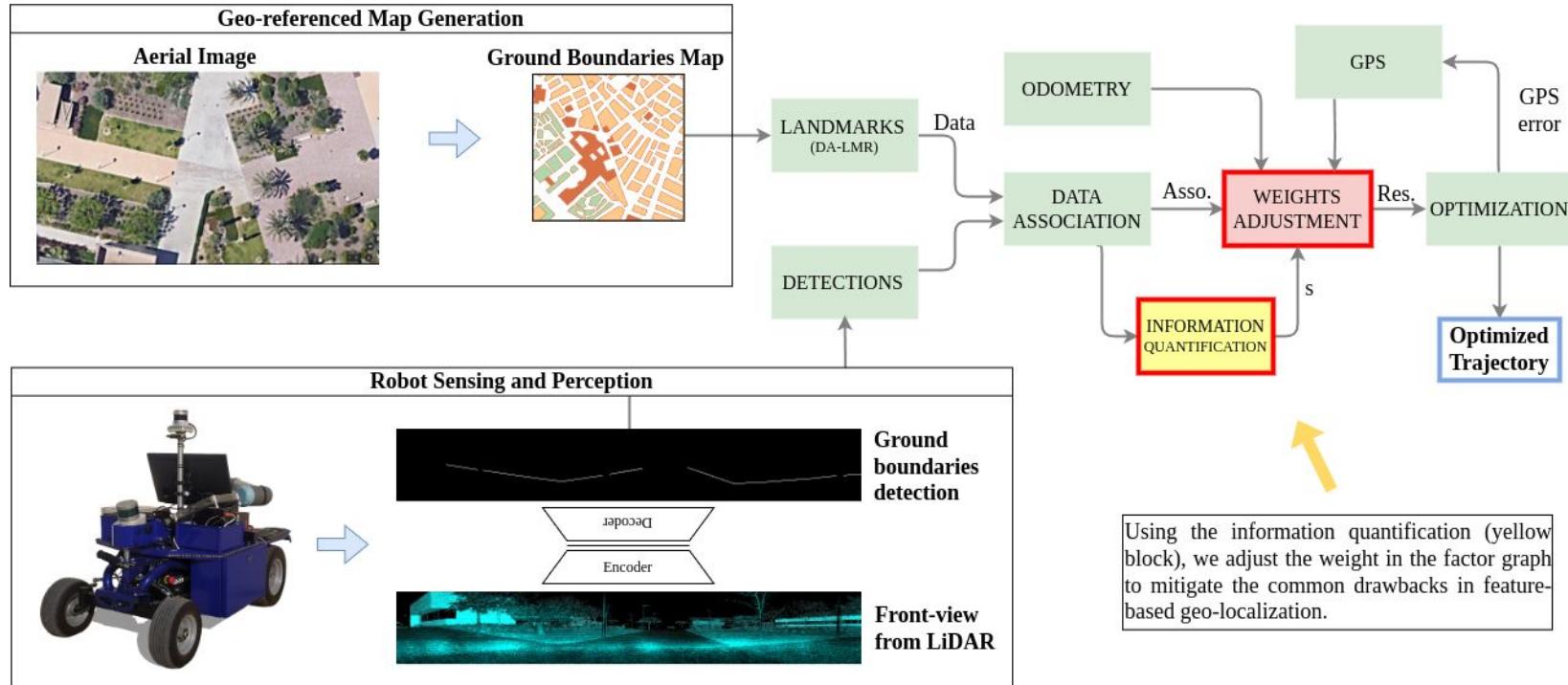
2.a. Geo-localización (caso real)



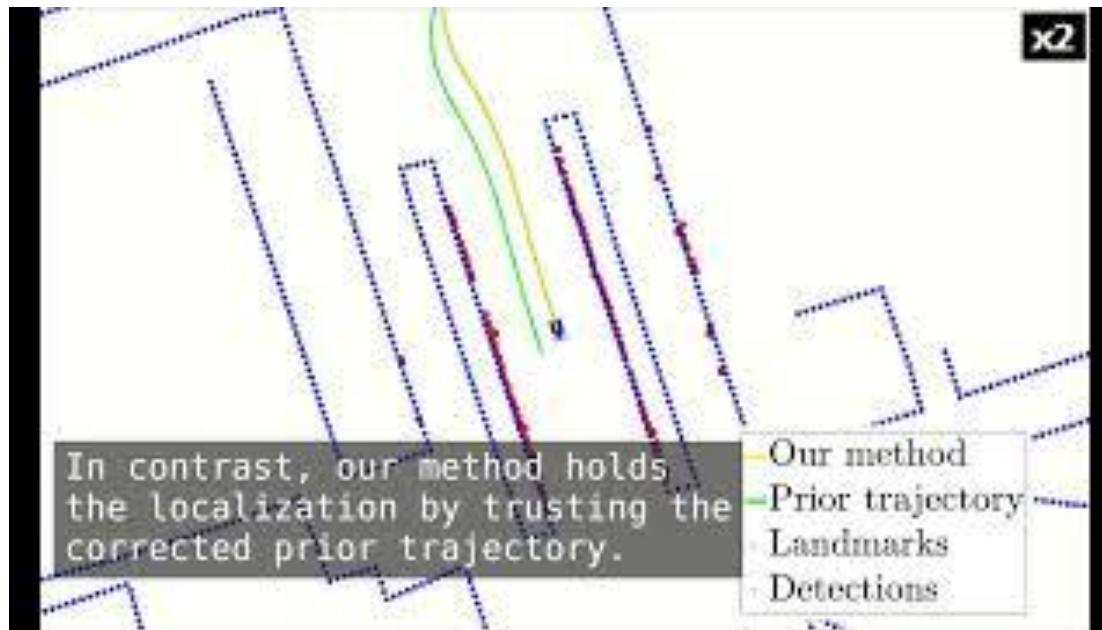
2.a. Geo-localización (caso real)



2.a. Geo-localización (caso real)



2.a. Geo-localización (caso real)



<https://youtu.be/z9zCJZAgoUU?si=B6SnEfz-DXs2jOml>

2.a. Geo-localización (caso real)



<https://www.mrt.kit.edu/english/index.php>



2.a. Geo-localización (caso real)

IEEE ROBOTICS AND AUTOMATION LETTERS, VOL. 7, NO. 4, OCTOBER 2022



12339

Robust Self-Tuning Data Association for Geo-Referencing Using Lane Markings

Miguel Ángel Muñoz-Bañón , Jan-Hendrik Pauls , Haohao Hu , Christoph Stiller ,
Francisco A. Candelas , and Fernando Torres 

<https://ieeexplore.ieee.org/document/9928353>

2.a. Geo-localización (caso real)

IEEE ROBOTICS AND AUTOMATION LETTERS, VOL. 9, NO. 6, JUNE 2024

5599

Geo-Localization Based on Dynamically Weighted Factor-Graph

Miguel Ángel Muñoz-Bañón , Alejandro Olivas , Edison Velasco-Sánchez , Francisco A. Candelas ,
and Fernando Torres 

<https://ieeexplore.ieee.org/document/10517630>

Índice

1. Introducción
2. Localización: **Estimación de $x(t)$**
 - a. Odometría LiDAR
 - b. Geo-localización
3. **Planificación y control: Estimación de $u(t)$**
 - a. Planificación local
 - b. Planificación local mediante aprendizaje
4. Otros temas relacionados
 - a. Oxford Spires dataset
 - b. Proyecto Digiforest
 - c. Sapling-NeRF
5. Líneas de investigación abiertas

3. Planificación y control: Estimación de $u(t)$

- La planificación y control se basa básicamente en encontrar u_k a partir de estado y las observaciones
 - Puede realizarse con algoritmos basados en geometría
 - O mediante aprendizaje

$$u_k = g(x_k, x_d, y_k)$$

Índice

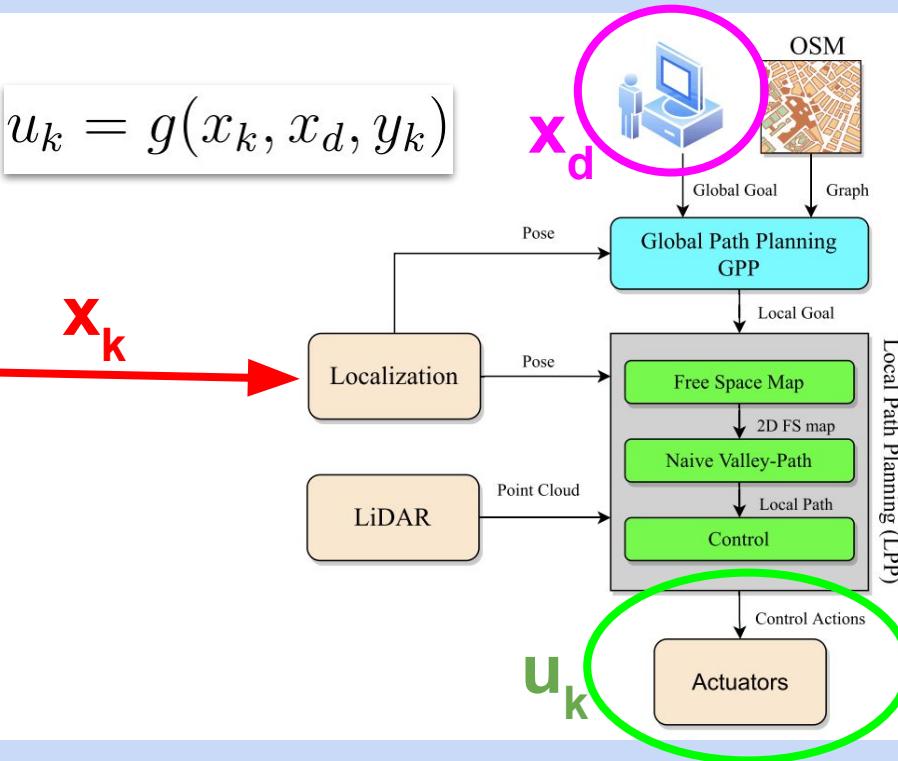
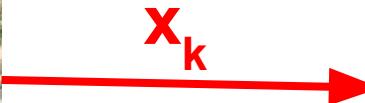
1. Introducción
2. Localización: **Estimación de $x(t)$**
 - a. Odometría LiDAR
 - b. Geo-localización
3. **Planificación y control: Estimación de $u(t)$**
 - a. **Planificación local**
 - b. Planificación local mediante aprendizaje
4. Otros temas relacionados
 - a. Oxford Spires dataset
 - b. Proyecto Digiforest
 - c. Sapling-NeRF
5. Líneas de investigación abiertas

3.a. Planificación local



$$u_k = g(x_k, x_d, y_k)$$

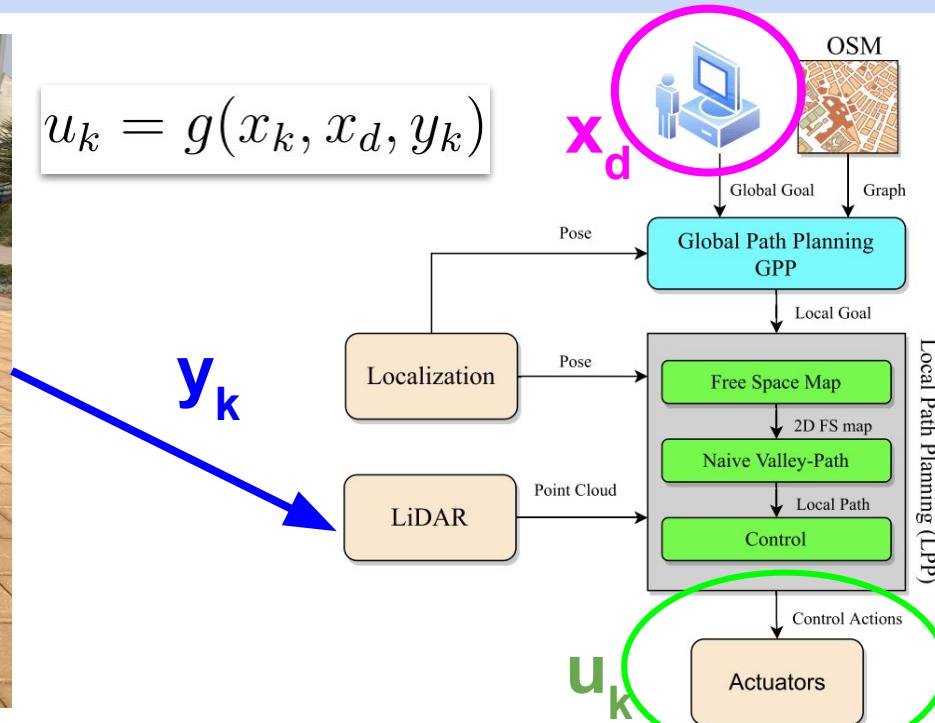
x_k



3.a. Planificación local



$$u_k = g(x_k, x_d, y_k)$$



3.a. Planificación local



<https://youtu.be/w0pikf8788k?si=XMBaObuxpOA32Owv>

3.a. Planificación local

24428

IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 23, NO. 12, DECEMBER 2022

OpenStreetMap-Based Autonomous Navigation With LiDAR Naive-Valley-Path Obstacle Avoidance

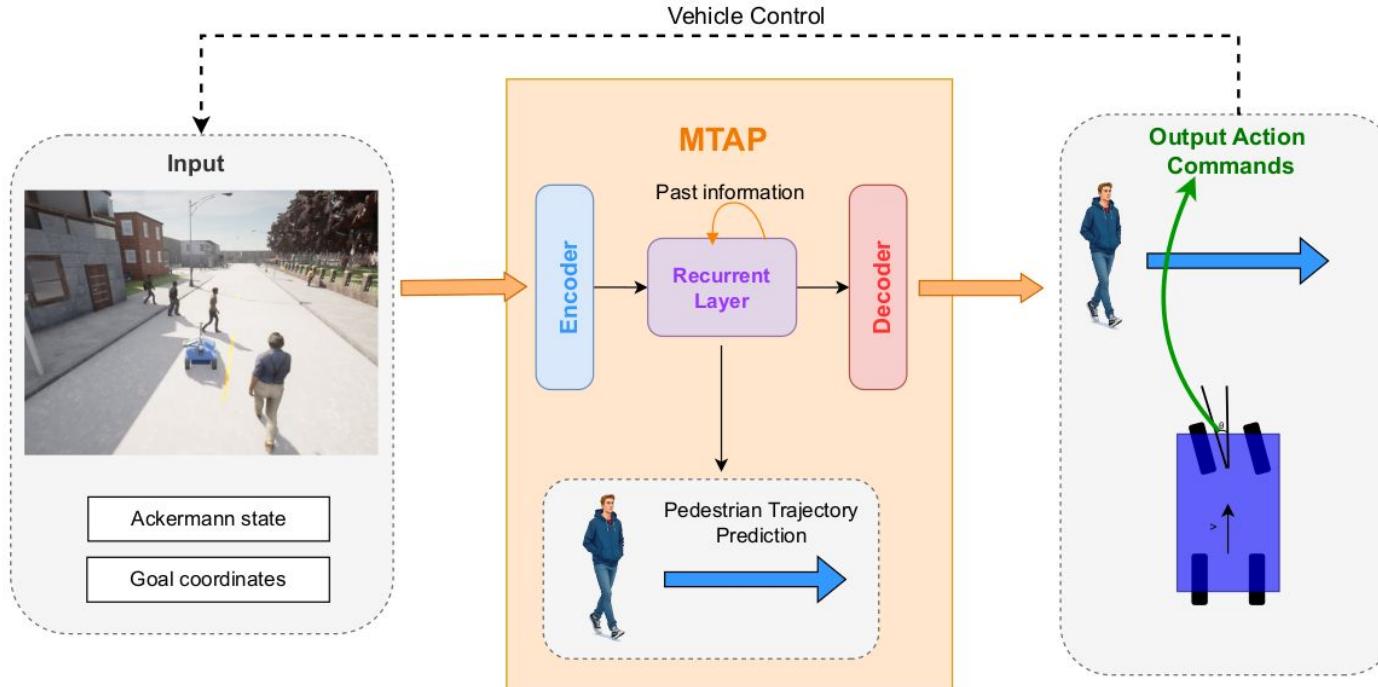
Miguel Ángel Muñoz-Bañón^{ID}, Edison Velasco-Sánchez^{ID}, Francisco A. Candelas^{ID},
and Fernando Torres^{ID}, *Senior member, IEEE*

<https://ieeexplore.ieee.org/abstract/document/9906811>

Índice

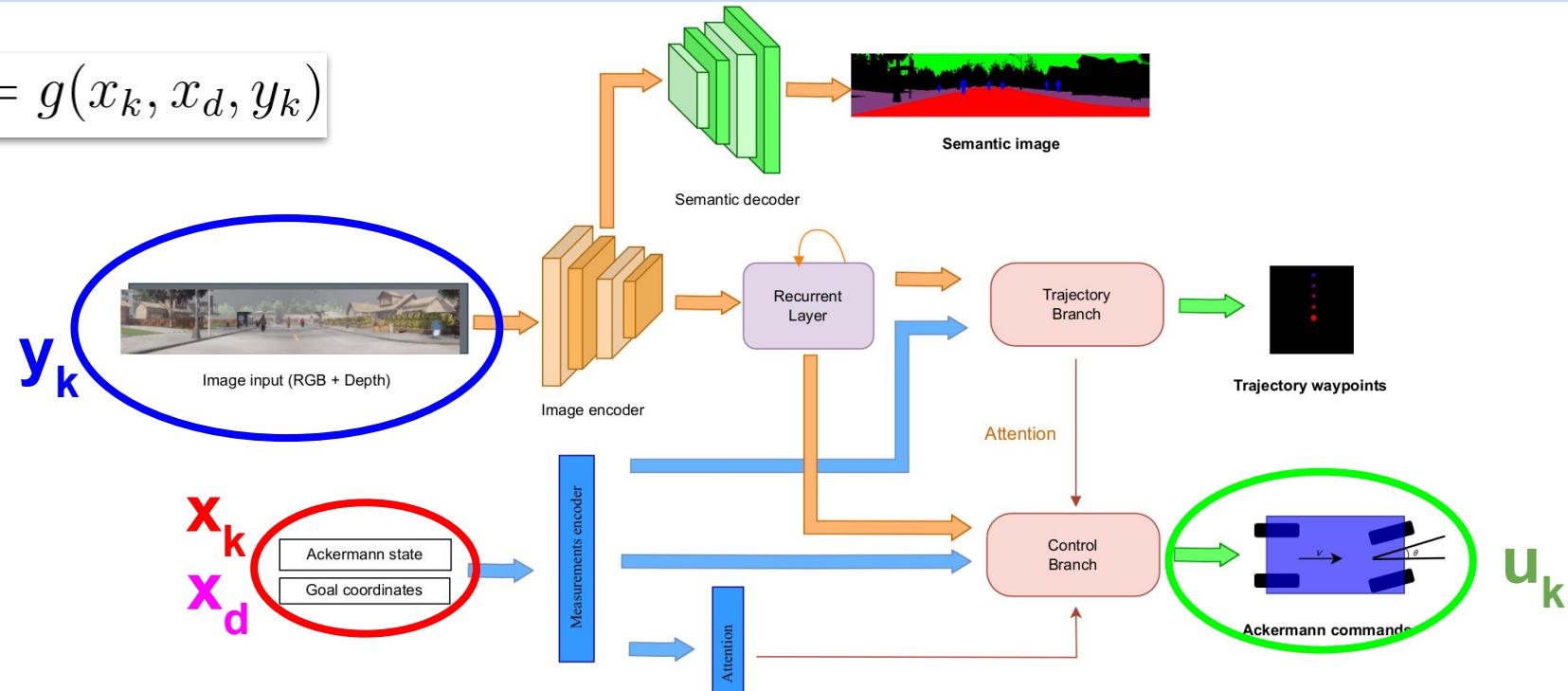
1. Introducción
2. Localización: **Estimación de $x(t)$**
 - a. Odometría LiDAR
 - b. Geo-localización
3. **Planificación y control: Estimación de $u(t)$**
 - a. Planificación local
 - b. Planificación local mediante aprendizaje**
4. Otros temas relacionados
 - a. Oxford Spires dataset
 - b. Proyecto Digiforest
 - c. Sapling-NeRF
5. Líneas de investigación abiertas

3.b. Planificación mediante aprendizaje (real)

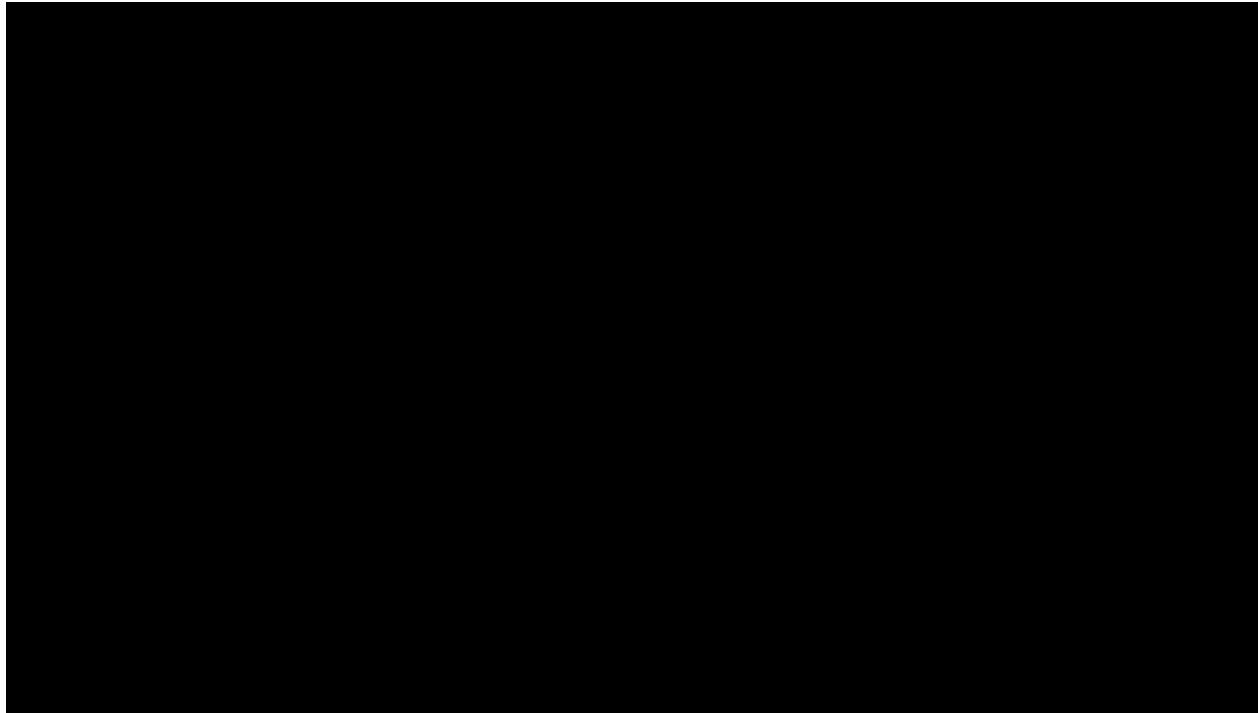


3.b. Planificación mediante aprendizaje (real)

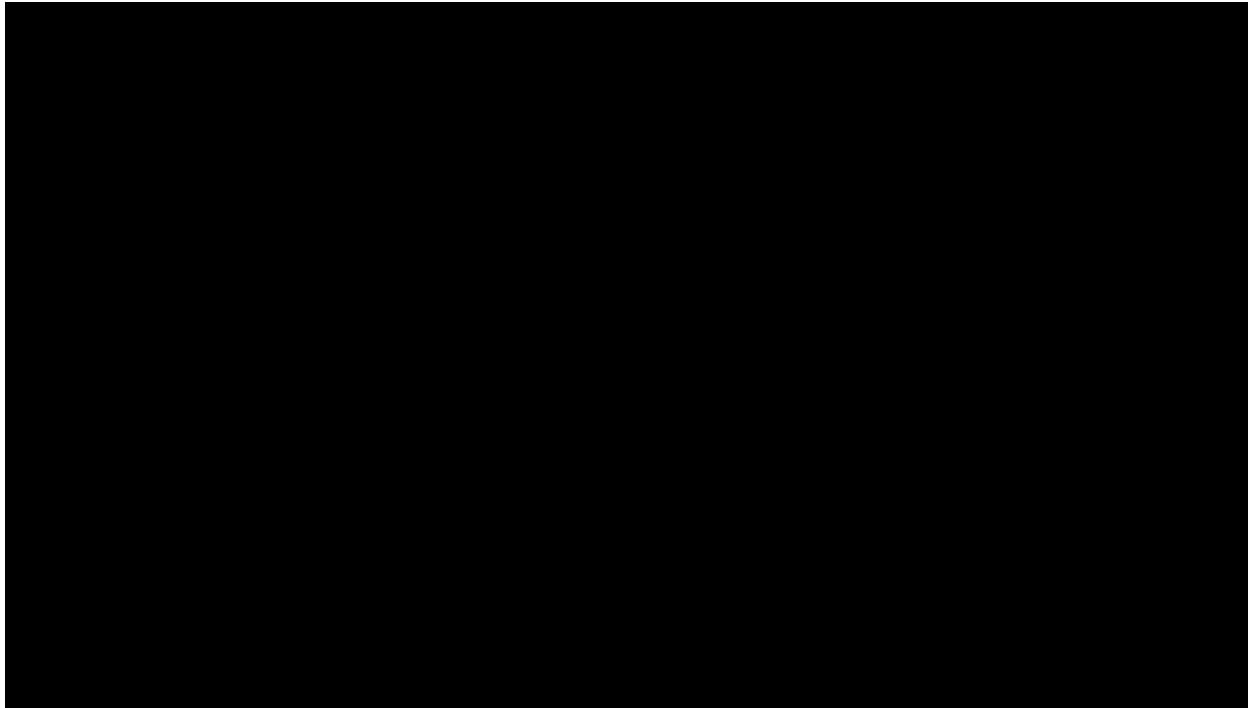
$$u_k = g(x_k, x_d, y_k)$$



3.b. Planificación mediante aprendizaje (real)



3.b. Planificación mediante aprendizaje (real)



Índice

1. Introducción
2. Localización: **Estimación de $x(t)$**
 - a. Odometría LiDAR
 - b. Geo-localización
3. Planificación y control: **Estimación de $u(t)$**
 - a. Planificación local
 - b. Planificación local mediante aprendizaje
4. **Otros temas relacionados**
 - a. **Oxford Spires dataset**
 - b. Proyecto Digiforest
 - c. Sapling-NeRF
5. Líneas de investigación abiertas

4.a. Oxford Spires dataset



<https://ori.ox.ac.uk/>

<https://dynamic.robots.ox.ac.uk/>

<https://dynamic.robots.ox.ac.uk/datasets/oxford-spires/>

Oxford Spires Dataset

Yifu Tao, Miguel Ángel Muñoz-Bañón, Lintong Zhang, Jiahao Wang, Lanke Frank Tarimo Fu, Maurice Fallon



Arxiv



YouTube



Code



HuggingFace



GDrive



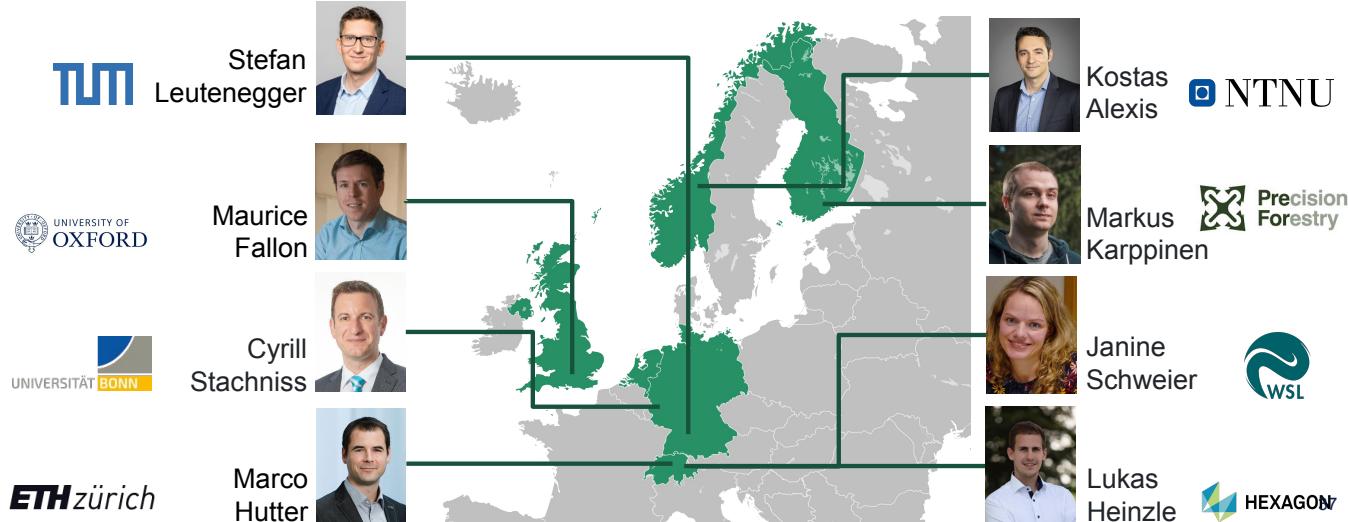
Índice

1. Introducción
2. Localización: **Estimación de $x(t)$**
 - a. Odometría LiDAR
 - b. Geo-localización
3. Planificación y control: **Estimación de $u(t)$**
 - a. Planificación local
 - b. Planificación local mediante aprendizaje
- 4. Otros temas relacionados**
 - a. Oxford Spires dataset
 - b. Proyecto Digiforest**
 - c. Sapling-NeRF
5. Líneas de investigación abiertas

UA

UNIVERSITAT D'ALACANT
UNIVERSIDAD DE ALICANTE
Escola Politècnica Superior
Escuela Politécnica Superior

4.b. Proyecto Digiforest



<https://digiforest.eu/>

4.b. Proyecto Digiforest

<https://digiforest.eu/news/2025-02-17-oxford-demo>



4.b. Proyecto Digiforest

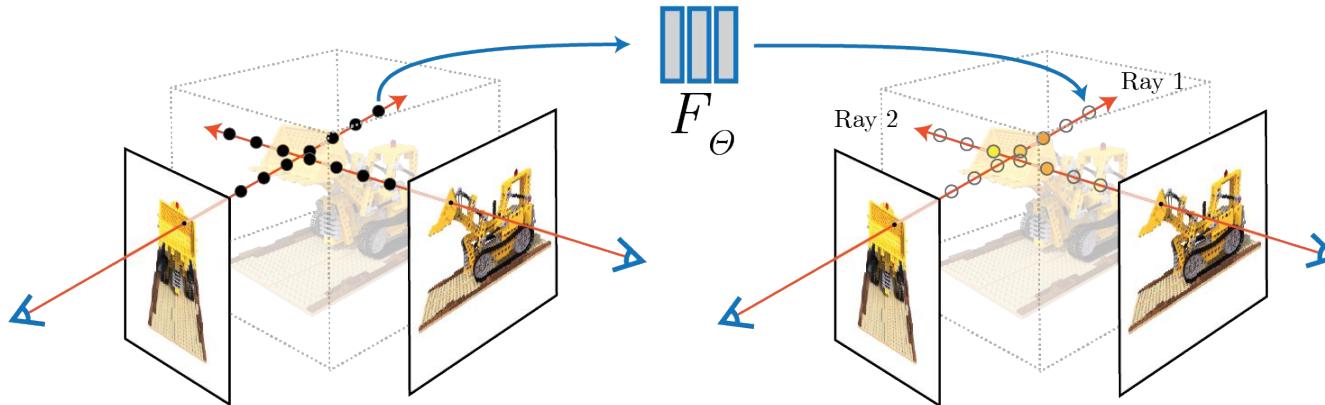


Índice

1. Introducción
2. Localización: **Estimación de $x(t)$**
 - a. Odometría LiDAR
 - b. Geo-localización
3. Planificación y control: **Estimación de $u(t)$**
 - a. Planificación local
 - b. Planificación local mediante aprendizaje
- 4. Otros temas relacionados**
 - a. Oxford Spires dataset
 - b. Proyecto Digiforest
 - c. Sapling-NeRF**
5. Líneas de investigación abiertas

4.c. Sapling-NeRF

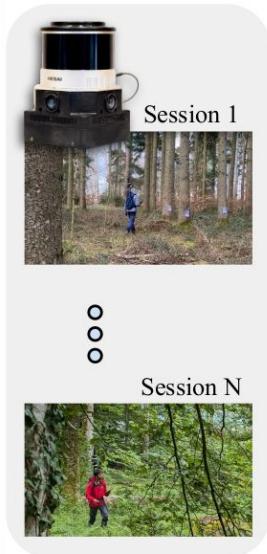
- **Neural Radiance Fields (NeRF)** es una tecnología que codifica el entorno en una red neuronal (volumetric representation).
 - Entrada: **Imágenes localizadas con precisión**.
 - Con pocas imágenes se puede obtener **imágenes sintéticas fotorealistas**.



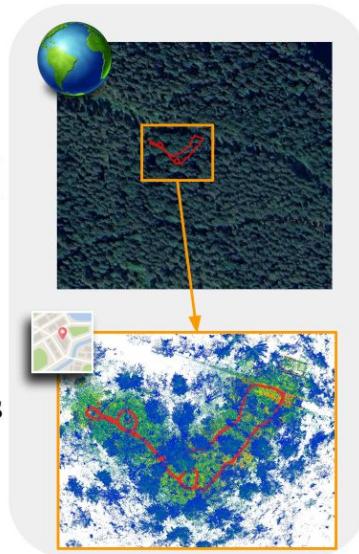
NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis
B Mildenhall, P. Srinivasan, M Tancik, J Barron, R Ramamoorthi, R Ng. 2020.

4.c. Sapling-NeRF

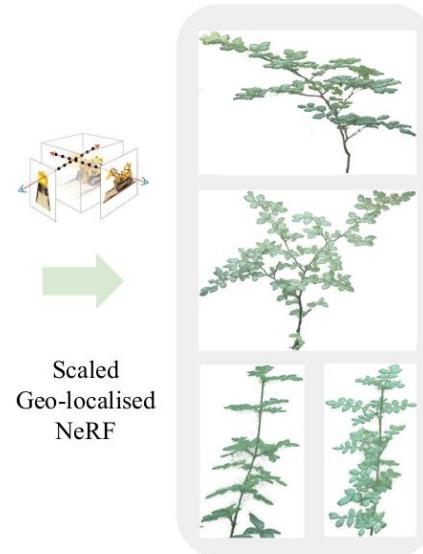
Data recording



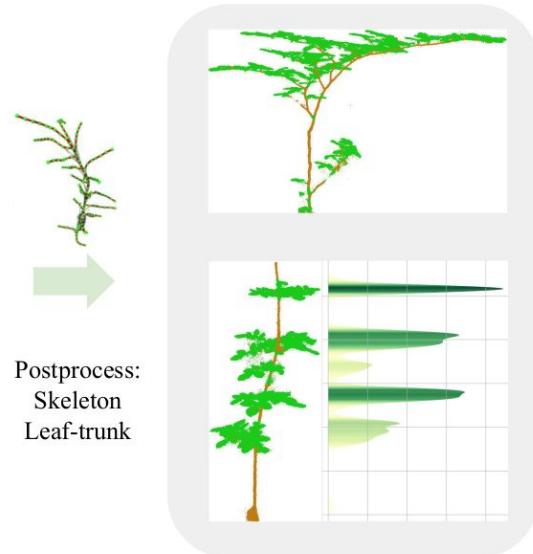
Global map



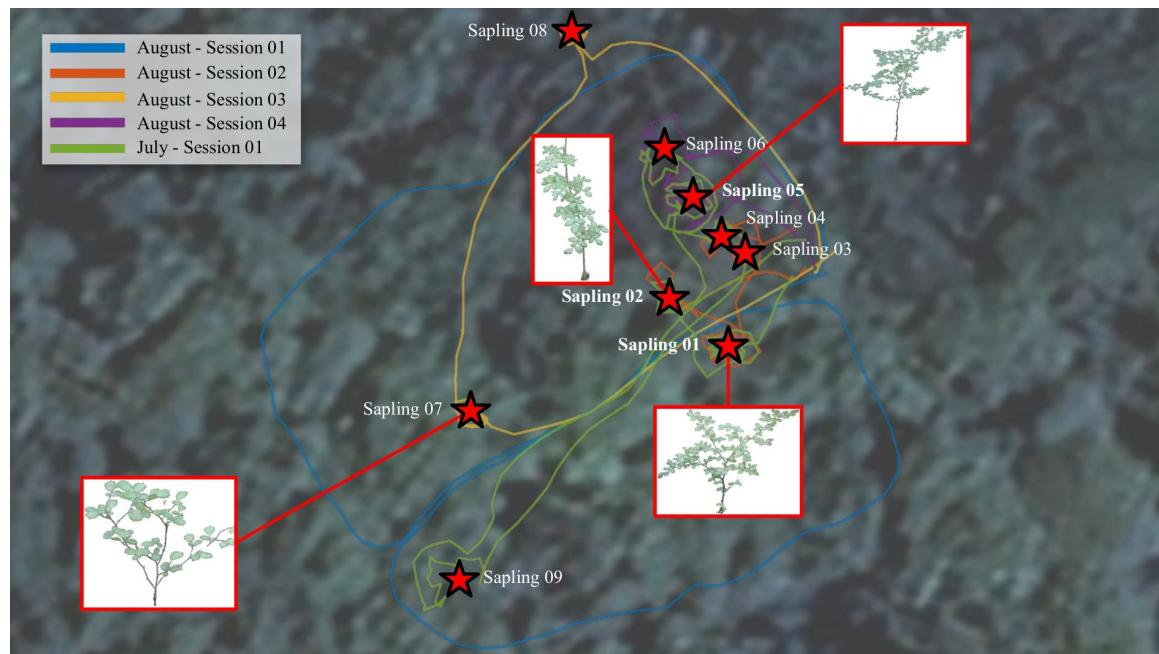
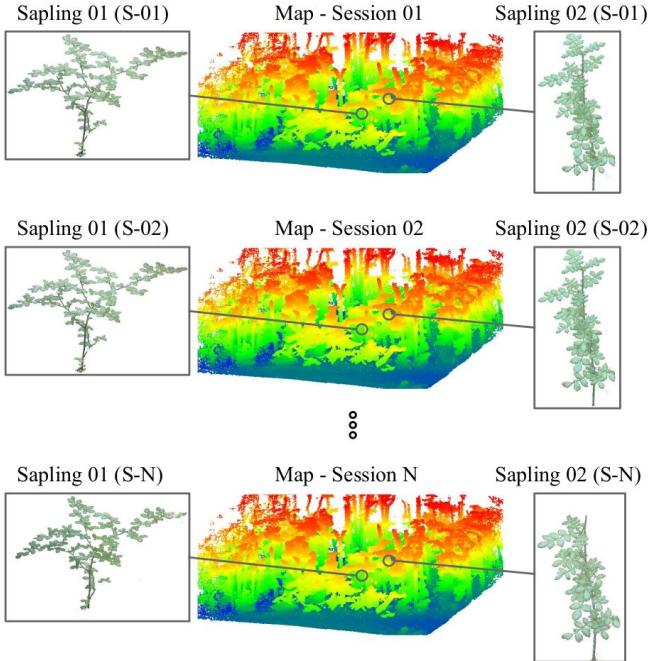
Dense sapling representation



Ecological monitoring



4.c. Sapling-NeRF



4.c. Sapling-NeRF

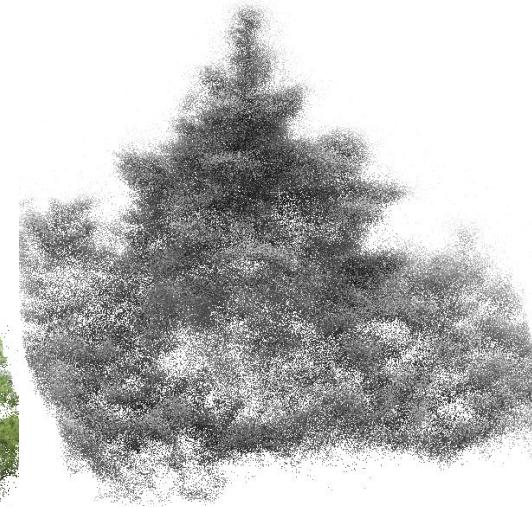
NeRF rendered video



Sapling recons. with **NeRF**

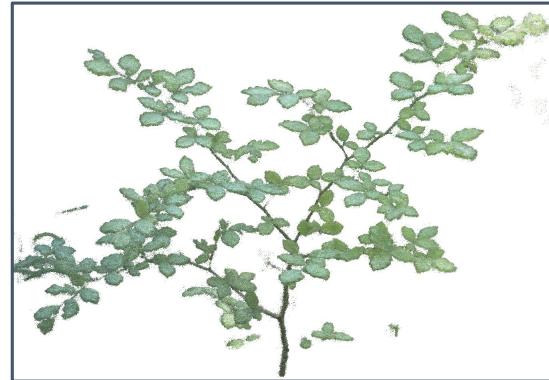


Sapling recons. with **LiDAR**



4.c. Sapling-NeRF

- Data recording for saplings in Wytham:



4.c. Sapling-NeRF

- Data recording for saplings in Wytham:



Sapling-NeRF: Geo-Localised Sapling Reconstruction in Forests for Ecological Monitoring

**Miguel Ángel Muñoz-Bañón^{1,2}, Nived Chebrolu¹, Sruthi M. Krishna Moorthy³,
Yifu Tao¹, Fernando Torres², Roberto Salguero-Gómez³ and Maurice Fallon¹**

¹Oxford Robotics Institute, Department of Engineering Science, University of Oxford, Oxford, UK

²Group of Automation, Robotics and Computer Vision, University of Alicante, Alicante, Spain

³Department of Biology, University of Oxford, Oxford, UK

Corresponding author: Miguel Ángel Muñoz-Bañón (email: miguelangel.munoz@ua.es).

This work is supported in part by the EU Horizon 2020 Project 101070405 (DigiForest) and a Royal Society University Research Fellowship. Miguel Ángel Muñoz-Bañón is supported by the Valencian Community Government and the European Union through the fellowship CIAPOS/2023/101

Índice

1. Introducción
2. Localización: **Estimación de $x(t)$**
 - a. Odometría LiDAR
 - b. Geo-localización
3. Planificación y control: **Estimación de $u(t)$**
 - a. Planificación local
 - b. Planificación local mediante aprendizaje
4. Otros temas relacionados
 - a. Oxford Spires dataset
 - b. Proyecto Digiforest
 - c. Sapling-NeRF
5. **Líneas de investigación abiertas**

5. Líneas de investigación abiertas

- **Geo-localización**
 - Utilizar aprendizaje automático
- **Planificación local mediante aprendizaje**
 - Sm-to-real para aprendizaje por refuerzo
- **Localización y planificación en bosques**
 - Mapeo de bosques para inventario de árboles secos (prevención de incendios)
 - Navegación autónoma con robot Unitree Go2
- **NeRF/3DGS para árboles pequeños en bosques**
 - Abordar problemas de iluminación
 - Integración de información multiespectral

Si a alguien le interesa realizar el TFG en alguno de estos temas contactar en: miguelangel.munoz@ua.es

Tema 7: Inteligencia artificial aplicada al modelado de sistemas

MODELOS COMPUTACIONALES Y SIMULACIÓN DE SISTEMAS
Curso 2025-2026



MODELOS COMPUTACIONALES Y SIMULACIÓN DE SISTEMAS

Examen Final de curso 2025-2026



UNIVERSITAT D'ALACANT
UNIVERSIDAD DE ALICANTE
Escola Politècnica Superior
Escuela Politécnica Superior



Temario

B1 - Sistemas Dinámicos

T1: Fundamentos del modelado de sistemas dinámicos

T2: Estabilidad, controlabilidad y observabilidad

T3: Paradigmas de simulación

B2 - Sistemas Complejos

T4: Lenguajes formales para modelos conceptuales

T5: Redes complejas y modelado estructural

B3 - Modelado con IA

T6: Identificación de sistemas

T7: Inteligencia artificial aplicada al modelado de sistemas

Evaluación

Descripción	Criterio	Tipo	Ponderación
Prácticas	Las prácticas se enfocan en el desarrollo de competencias técnicas y la implementación de modelos computacionales, fundamentales para la simulación y análisis de sistemas complejos.	ACTIVIDADES DE EVALUACIÓN DURANTE EL SEMESTRE	40
Trabajo final	El trabajo final, permite evaluar la aplicación de los conocimientos en un proyecto integrador, promoviendo la investigación, el trabajo autónomo y la capacidad de síntesis.	ACTIVIDADES DE EVALUACIÓN DURANTE EL SEMESTRE	10
Examen final de teoría	El examen final teórico tiene como objetivo medir la comprensión de los conceptos fundamentales y la capacidad de análisis crítico en los temas abordados.	EXAMEN FINAL	50

- Para la teoría se proporcionarán las transparencias. Pero se completarán con explicaciones en clase.
TOMAR APUNTES
- Las prácticas se ponderarán según el número de sesiones.
- Se realizará un trabajo final que se propondrá a final del curso.

Normas básicas

- La entrega de este examen implica el uso de **una convocatoria**.
- Se necesita llegar **al menos al 4 sobre 10** para hacer media con las prácticas y el trabajo teórico.
- **No se pueden tener apuntes** ni otros documentos a la vista, solamente bolígrafo o lápiz y calculadora para la Parte III.
- El tiempo para realizar este examen es de **100 minutos**.

Normas básicas

- La solución escogida para cada pregunta del test se debe **especificar rodeando con bolígrafo la letra** de la respuesta.
- Para cada pregunta del test se debe especificar **una sola respuesta** (A, B, C o D) de forma clara; de lo contrario, será considerada como respuesta incorrecta.
- De las preguntas tipo test: **a partir de 3 respuestas incorrectas**, cada nueva incorrecta descuenta una correcta. Las preguntas no contestadas no modifican la puntuación.
- Las preguntas cortas y el ejercicio **se deben contestar en folios aparte** con nombre, apellidos y DNI en cada folio.
- Responder de manera clara y concisa.

Estructura del examen

- Estructura del examen
 - La Parte I contiene **20 preguntas tipo test** (4 puntos).
 - La Parte II contiene **10 preguntas de desarrollo** (4 puntos).
 - La Parte III contiene **un ejercicio práctico** (2 puntos).

Preguntas de tipo test (Parte I)

En un sistema dinámico continuo lineal descrito por $\dot{x}(t) = Ax(t)$, ¿cuál de las siguientes afirmaciones describe correctamente la condición de **estabilidad asintótica**? (d)

- a) Todos los autovalores de A tienen parte real positiva.
- b) Todos los autovalores de A tienen parte real igual a cero.
- c) Al menos uno de los autovalores de A tiene parte real negativa.
- d) Todos los autovalores de A tienen parte real estrictamente negativa.

Preguntas de tipo test (Parte I)

Considera un robot móvil cuyo estado es $x = (x, y, \theta)$, es decir, posición y orientación. La única observación disponible proviene de un GPS que mide $y = (x, y)$. ¿Cuál de las siguientes afirmaciones es correcta? (c)

- a) El sistema es totalmente observable porque el GPS proporciona dos de las tres variables del estado que permiten actualizar la posición.
- b) El sistema es observable únicamente si θ es constante en el tiempo.
- c) El sistema no es observable porque la salida no contiene información suficiente para reconstruir la componente θ del estado.
- d) El sistema es observable si el robot se mueve con velocidad distinta de cero.

Preguntas de teoría (Parte II)

- Describe con palabras el proceso para estimar la varianza de la estimación del estado futuro $x(k+1)$ (en una dimensión) dada una señal de control $u(k)$ ruidosa.
 - **Respuesta:** Esto se podría realizar mediante la simulación de Monte Carlo. Se pueden generar N muestras de $x(k+1)$ con diferentes muestras de $u(k)$ que serán diferentes debido a la incertidumbre. Una vez obtenidas las N muestras se puede realizar un promedio (valor final estimado de $x(k+1)$), y obtener la varianza, que representaría la incertidumbre del estado.

Ejercicio práctico (Parte III)

E1. Análisis de estabilidad mediante el método de Euler

Considera el siguiente sistema dinámico continuo de una dimensión:

$$\dot{x}(t) = -2x(t) + x^3(t).$$

Queremos estudiar numéricamente la estabilidad de $x^* = 0$ usando el **método de Euler explícito** con paso de integración $h = 0,1$.

- Escribe la ecuación de actualización del método de Euler para este sistema, discretizando como $x_k \approx x(t_k)$ y $t_k = kh$.
- Partiendo de $x_0 = 0,05$, completa una tabla de la evolución aproximada de x_k hasta $k = 15$ (es decir, hasta $t = 1,5$) para observar el comportamiento del sistema.

Ejercicio práctico (Parte III)

- c) A partir de la evolución numérica obtenida. Basándote en la integración numérica, indica si el punto de equilibrio $x^* = 0$ es estable o inestable.
- d) Compara tu conclusión numérica con el análisis analítico local basado en la linealización del sistema en torno a $x^* = 0$:

$$f'(x) = -2 + 3x^2.$$

¿Coinciden las conclusiones? ¿Coménta por qué?

Últimas consideraciones

- Como ejercicio práctico también puede entrar alguno del tipo que se haya visto en teoría en las transparencias.
- Con el objetivo de mejorar la asignatura para el próximo año, se os pasará una encuesta preguntando por cómo habríais preferido la asignatura (se dejarán 10 minutos más).
 - La encuesta es **voluntaria y anónima**.
 - Se preguntarán cosas del tipo: “*Habría preferido menos contenido y profundizar más*”, “*Habría preferido tener que programar más en las prácticas*” o cosas por el estilo

MODELOS COMPUTACIONALES Y SIMULACIÓN DE SISTEMAS

Examen Final de curso 2025-2026



UNIVERSITAT D'ALACANT
UNIVERSIDAD DE ALICANTE
Escola Politècnica Superior
Escuela Politécnica Superior

