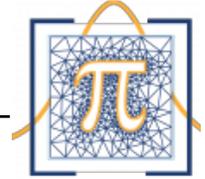




UNIVERSIDAD NACIONAL AUTÓNOMA DE
HONDURAS - UNAH



FACULTAD DE CIENCIAS

**Identificación dispersa de sistemas epidemiológicos lineales
y no lineales mediante SINDy y SDSI**

TESIS

PARA OBTENER EL GRADO DE

**Máster en Matemática con Orientación en Ingeniería
Matemática**

PRESENTA:

William Alberto Funez Izaguirre

ASESOR:

Dr. Fredy Vides

Tegucigalpa, Honduras

Mayo, 2022

Resumen

Muchos sistemas dinámicos de interés son no lineales, con ejemplos en la turbulencia, la epidemiología, la neurociencia, las finanzas y entre otros, lo que dificulta su control mediante enfoques lineales. El control predictivo de modelos (CPM) es una potente técnica de optimización basada en modelos que permite controlar estos sistemas no lineales con restricciones. Sin embargo, los sistemas modernos carecen a menudo de modelos computables, lo que motiva el uso de técnicas de identificación del sistema para aprender modelos precisos y eficientes para el control en tiempo real. En este trabajo, se presentan algunos resultados generales de los métodos emergentes basados en datos para el descubrimiento de modelos y cómo se utilizan para el CPM no lineal. En particular, nos centramos en los algoritmos de identificación dispersa de dinámicas no lineales (SINDy) y la identificación de sistemas dinámicos dispersos (SDSI); mostramos cómo pueden emplearse con CPM en un experimento de control de enfermedades infecciosas usando un modelo epidemiológico SEIR. Comparamos el rendimiento con el de los CPM basados en un modelo de descomposición en modo dinámica (DMD). También consideramos un experimento para sistemas dinámicos lineales, para ello se utilizó el modelo SIRD. En este caso comparamos SINDy y SDSI con la programación cuadrática secuencial (SQP). En ambos experimentos se mostrará la fortaleza de la novedosa herramienta SDSI para la identificación de sistemas epidemiológicos.

Abstract

Many dynamical systems of interest are nonlinear, with examples in turbulence, epidemiology, neuroscience, finance, and others, making it difficult to control using linear approaches. Model predictive control (MPC) is a powerful model-based optimization technique that allows you to control these nonlinear systems with constraints. However, modern systems often lack computable models, which motivates the use of system identification techniques to learn accurate and efficient models for real-time control. In this work, some general results of emerging data-driven methods for model discovery and how they are used for nonlinear MPC are presented. In particular, we focus on sparse identification of nonlinear dynamics (SINDy) and sparse dynamical system identification (SDSI) algorithms; we show how they can be used with MPC in an infectious disease control experiment using an epidemiological SEIR model. We compare the performance with that of MPCs based on a dynamic mode decomposition (DMD) model. We also consider an experiment for a linear dynamic system, for which the SIRD model was used. In this case we compare SINDy and SDSI with Sequential Quadratic Programming (SQP). In both experiments, the strength of the new SDSI tool for the identification of epidemiological systems will be shown.

Agradecimientos

Agradeciendo en primer lugar a Dios por haberme dado la sabiduría, conocimiento y fuerza para poder realizar este trabajo.

A mi asesor, el Dr. Fredy Vides, por haber confiado en mí y brindarme su apoyo, asesoría, tiempo y guía para la realización de este trabajo de tesis. Agradecerle de todo corazón por todos sus sabios consejos que fueron de gran ayuda en mi trabajo de investigación, ya que de no ser así todo esto no hubiese sido posible, junto con el apoyo y recursos computacionales del Centro de Innovación en Computación Científica (CICC-UNAH) de la Universidad Nacional Autónoma de Honduras.

A mis padres Jose Funez y Nusly Izaguirre junto con mi familia por todo el cariño, amor y apoyo que me han brindado toda mi vida.

A mi universidad por formarme con cada cosa que aprendí de cada profesor que estuvo en mi formación en este proceso, por sus charlas y enseñanzas de Matemáticas, las experiencias que quiérase o no ayudaron a mi formación, me ayudaron a tener inspiración, motivación y me hicieron entender que por muy difíciles que se vean las cosas, por muy poco que sea el tiempo que haya en el reloj, el carácter de las personas sale a flote y es donde se ve quién está preparado para enfrentar sus miedos. A todos los que fueron parte gracias.

Tabla de contenidos

Resumen	ii
Abstract	iii
Índice de figuras	vi
1 Introducción	1
1.1 Visión general	1
1.2 Modelado basado en datos	2
1.3 Objetivos y justificación	3
2 Preliminares	5
2.1 Definiciones y terminología	5
2.2 Descomposición en Valores Singulares (SVD)	7
2.2.1 Descripción y definición	7
2.2.2 Existencia y unicidad	8
2.2.3 Propiedades de la matriz	8
2.3 Descomposición Ortogonal Adecuada (POD)	11
2.3.1 Optimalidad de la base POD	12
2.3.2 Cálculo de modelos de orden reducido	12
2.3.3 Métodos basados en mediciones	13
2.3.4 Reducción de dimensión	13
2.4 Descomposición en Modo Dinámico (DMD)	14
2.4.1 Descripción y definición	15
2.4.2 El algoritmo de DMD	15
2.4.3 DMD en observables no lineales	17
2.4.4 El algoritmo de Koopman	17
3 Métodos	19
3.1 Identificación dispersa de dinámicas no lineales (SINDy)	19
3.1.1 Algoritmo	20
3.2 Identificación de sistemas dinámicos dispersos (SDSI)	21
3.2.1 Aproximación de bajo rango y solucionadores de mínimos cuadrados lineales dispersos	22

3.2.2	Métodos de aproximación de rango bajo para la identificación de modelos dispersos	24
3.2.3	Algoritmos	25
3.3	Modelo basado en datos para CPM	30
4	Experimentos y Resultados	33
4.1	Experimento 1	33
4.2	Experimento 2	35
4.3	Experimento 3	37
4.4	Experimento 4	42
5	Conclusiones y Trabajos futuros	48
5.1	Conclusiones	48
5.2	Trabajos futuros	48
	Bibliografía	50

Índice de figuras

3.1	Señales identificadas para diferentes valores de retardo. $L = 17$ (primera fila superior). $L = 16$ (segunda fila). $L = 10$ (tercera fila). $L = 5$ (última fila). Fuente: [64]	29
3.2	Coefficientes identificados para diferentes valores de retardo. $L = 17$ (primera fila superior). $L = 16$ (segunda fila). $L = 10$ (tercera fila). $L = 5$ (última fila). Fuente: [64]	30
4.1	Experimento 1. Dinámica espacio-temporal de dos señales (a) $f_1(x, t)$, (b) $f_2(x, t)$ y la mezcla en (c) $f(x, t) = f_1(x, t) + f_2(x, t)$. La función $f(x, t)$ se puede representar en su lugar por $x(t)$. Se calculó la DMD de x , y en la imagen (d) se muestra una reconstrucción aproximada de rango 2 de la señal $x_{DMD}(t)$. La reconstrucción es casi perfecta, con los modos y espectros DMD que coinciden estrechamente con las señales $f_1(x, t)$ y $f_2(x, t)$. Fuente: Elaboración propia.	34
4.2	En las imágenes se muestra una comparación de DMD con PCA aplicada a los datos del Experimento 1. Fuente: Elaboración propia.	34
4.3	Experimento 2 de la ecuación no lineal de Schrödinger (NLS) 2.9 donde en la imagen (a) se muestra la simulación completa. La aproximación se muestra en la imagen (b) usando el algoritmo de DMD. Fuente: Elaboración propia.	36
4.4	En la imagen se muestran las funciones propias de Koopman y distribución de valores propios para los tres observables considerados en los incisos (a)-(c) [fila superior], el algoritmo de DMD es dado por $\mathbf{g}_{DMD}(\mathbf{x}) = \mathbf{x} = u(\xi, t_k)$, se tiene los observables $\mathbf{g}_1(x) = [\mathbf{x}, \mathbf{x} ^2 \mathbf{x}]^T$ y $\mathbf{g}_2(\mathbf{x}) = [\mathbf{x}, \mathbf{x} ^2]^T$. Tener en cuenta que el observable $\mathbf{g}_1(\mathbf{x})$ [segunda imagen de la primera fila (b), segunda imagen de la segunda fila (e)] nos da la mejor aproximación al espectro de valores propios puramente imaginarios. En el análisis de errores de la reconstrucción de Koopman de los datos simulados, los tres observables considerados se comparan con la solución verdadera, el observable correspondiente a la teoría de DMD esta dado por $\mathbf{g}_{DMD}(\mathbf{x})$, y produce el error E_{DMD} , los observables $\mathbf{g}_1(\mathbf{x})$ y $\mathbf{g}_2(x)$ producen errores E_1 y E_2 , respectivamente. Fuente: Elaboración propia.	36
4.5	Esquema del algoritmo SINDy o SDSI con control. Se muestra el proceso para la identificación del sistema. Fuente: Elaboración propia.	39
4.6	Experimento 3. SINDy-CPM vs. DMD-CPM para el problema de control de enfermedades infecciosas. Fuente: Elaboración propia.	40

4.7 Experimento 3. Comparación de la dinámica verdadera versus cada uno de los modelos de identificación, podemos notar que tanto SDSI y SINDy logran una buena identificación de la dinámica verdadera, en el caso de SDSI se tiene que el valor del invariante topológico es igual a cero ($\text{drk}_{*,\delta}(\Gamma_T) = 0$). Fuente: Elaboración propia. 41

4.8 Experimento 3. Esquema del marco CPM basado en datos. Uso de DMD lineal, SDSI y SINDy no lineal disperso el control predictivo. Fuente: Elaboración propia. 41

4.9 Entrada de control. Fuente: Elaboración propia. 42

4.10 Costo. Fuente: Elaboración propia. 42

4.11 Experimento 4. Identificación del modelo SIRD mediante SINDy, SDSI y optimización regular. Fuente: Elaboración propia. 45

4.12 Experimento 4. Modelo SIRD, errores de aproximación. Fuente: Elaboración propia. 46

Capítulo 1

Introducción

1.1 Visión general

Existe una larga historia del modelado de dinámicas a partir de datos, lo que da como resultado técnicas poderosas para la identificación de sistemas [40]. Muchos de estos métodos surgieron de la necesidad de comprender estructuras complejas. Los métodos para la identificación de sistemas se han aplicado ampliamente en casi todas las áreas de la matemática, física e ingeniería, sobre todo para el control basado en modelos. Sin embargo, los métodos para la identificación de sistemas normalmente requieren suposiciones sobre la forma del modelo y la mayoría de las veces dan como resultado una dinámica lineal, lo que limita su eficacia a perturbaciones transitorias de pequeña amplitud alrededor de un punto fijo de la dinámica [28].

Los novedosos algoritmos SDSI [65] y SINDy [11], proporcionan un marco de descubrimiento de modelos basado en datos, lo que da como resultado modelos interpretables que evitan el sobreajuste, confiando en la optimización que promueve la dispersión para identificar modelos a partir de datos limitados. SINDy está estrechamente relacionado con la descomposición en modo dinámico (DMD) [54], que da como resultado un modelo lineal para la evolución de algunas estructuras espacio temporales coherentes a partir de datos de series temporales de alta dimensión. DMD se generalizó para incluir entradas externas y control [12], y este enfoque se aplicó posteriormente para desarrollar SINDy con control [33]. Ambos enfoques se han combinado con CPM, lo que permite modelos interpretables en el límite de datos bajos para el control en tiempo real de sistemas no lineales.

En esta tesis, se estudia SINDy y SDSI con control y se muestra la eficacia con CPM en un experimento de control de enfermedades infecciosas. Un objetivo principal de este trabajo es proporcionar las herramientas para aplicar métodos de identificación de sistemas controlados por datos a un problema de control predictivo de modelos.

En este trabajo se presentan algunas técnicas teóricas y computacionales que han sido desarrolladas para la creación del algoritmo SDSI para estructuras aproximadas, preservando la identificación dispersa de sistemas dinámicos de tiempo discreto basados en datos medidos desde los sistemas. En particular, exploramos la idea de utilizar aproximaciones de bajo rango de submatrices de las matrices de trayectoria tipo Hankel correspondientes a las muestras de datos, para el cálculo de las representaciones dispersas aproximadas de las matrices de parámetros a identificar como parte de la identificación del modelo.

Como parte del proceso descrito anteriormente, se obtienen algunos resultados generales en teoría de aproximación y análisis matricial con aplicaciones a la identificación dispersa de modelos de series de tiempo y sistemas dinámicos no lineales. La aproximación de los correspondientes operadores de transición de estado determinados por matrices de parámetros, se aborda identificando condiciones para la existencia de números enteros fácilmente computables que se pueden aplicar para estimar la computabilidad de representaciones dispersas aproximadas de las matrices de parámetros, y como producto del cálculo de estos números, se obtienen aproximaciones de bajo rango de submatrices de las matrices de trayectoria correspondientes a algunos datos medidos del sistema en estudio, que se pueden utilizar para calcular las aproximaciones dispersas de las matrices de parámetros.

Una de las ventajas del enfoque de aproximación de bajo rango presentado en este trabajo es la reducción de las complejidades aritméticas, como lo estudiaron Chen, Avron y Sindhvani en [16] en el contexto del aprendizaje no paramétrico escalable. El enfoque de aproximación de bajo rango implementado en este estudio hace que los algoritmos de resolución de mínimos cuadrados lineales dispersos, como el algoritmo 1 en [65], sean adecuados para la paralelización.

El estudio presentado en este trabajo se inspiró en las preguntas teóricas, computacionales y los resultados presentados por Boutsidis y Magdon-Ismail en [5], por Brockett y Willsky en [8], por Proctor, Brunton y Kutz en [12], por Farhood y Dullerud en [20], por Finzi, Stanton, Izmailov y Wilson en [21], por Franke y Selgrade en [22], por Freedman y Press en [24], por Kaheman, Kutz y Brunton en [32], por Kaiser, Kutz y Brunton en [33], y por Loring y Vides en [42], por Moskvina y Schmidt en [48], por Salova, Emenheiser, Rupe, Crutchfield y D'Souza en [51], por Schaeffer, Tran, Ward y Zhang en [52], por Schmid en [54].

Entre las referencias anteriores, desde una perspectiva computacional, una fuente clave de inspiración para el trabajo realizado en esta tesis fue la implementación computacional del algoritmo SINDy para la identificación de sistemas dispersos no lineales a lo largo de las líneas de [11] y [32].

Los métodos descritos en este trabajo comparten cierta similitud con la descomposición en modo dinámico (DMD), que es una regresión dinámica lineal [50, 54]. DMD es un ejemplo de un método sin ecuaciones [34], ya que únicamente se basa en datos de medición. Los avances en la DMD han desarrollado conexiones entre la DMD construido sobre funciones observables no lineales y la teoría del operador de Koopman para sistemas dinámicos no lineales [50, 47].

1.2 Modelado basado en datos

Se han dedicado siglos de investigación matemática a la creación de modelos que describan y predigan con precisión el comportamiento de sistemas no lineales [53], generalmente en forma de modelos deductivos de la mecánica derivados de teorías preexistentes. En los últimos años, con los avances en el machine learning [31], la mayor disponibilidad y comprensión de los datos se han logrado avances en la automatización del proceso de descubrimiento de modelos, creando modelos inductivos. Actualmente, existen técnicas rigurosas, como los métodos de regresión, para comprender los datos estáticos [30], pero los avances análogos dados los datos dinámicos no se han desarrollado tan rápidamente.

Uno de los primeros y más influyentes intentos realizados para motivar modelos dinámicos

utilizando datos empíricos fue realizado por Edward Lorenz con su artículo de 1963 titulado: “Deterministic Nonperiodic Flow” (ver [41]). A través de este famoso trabajo, desarrolló una teoría que sentaría las bases para la predicción meteorológica moderna, derivando técnicas de modelado estadístico no lineal a partir de datos atmosféricos [27]. Esto condujo a una comprensión mucho mejor de los sistemas dinámicos caóticos que a menudo están presentes en la naturaleza. Otros intentos tempranos de reconstruir dinámicas no lineales que se derivan de la teoría del caos involucraron la metodología de sistemas de coordenadas de retardo [49, 58]. Este método, aunque tuvo éxito en la reconstrucción de características del sistema como la dimensionalidad, los exponentes de Lyapunov y las órbitas periódicas inestables [66], no pudo usarse para recuperar una forma simbólica funcional que pudiera analizarse usando métodos tradicionales de plano de fase.

Un gran avance en el modelado de dinámicas no lineales se produjo en [4] y complementado por [55] aplicando regresión simbólica para recuperar ecuaciones diferenciales. Este trabajo fue el primer intento exitoso de automatizar el proceso de encontrar la estructura simbólica del sistema dinámico que rigen un proceso natural, ser capaz de modelar un sistema simbólicamente en lugar de numéricamente es crucial debido al valor explicativo de un modelo construido con funciones elementales. Sin embargo, estos primeros intentos que utilizaban programación genética estaban sujetos a sobreajuste, además de ser computacionalmente costosos y carecer de la capacidad de escalar bien a sistemas con mayor dimensionalidad. En la siguiente sección se presenta los objetivos y justificación de la tesis, donde se dará la importancia y líneas de investigación relacionada con este trabajo.

1.3 Objetivos y justificación

Una de las motivaciones de esta tesis es el uso de técnicas de aproximación de matrices de bajo rango para producir algoritmos que se pueden aplicar de manera efectiva para procesos de identificación de modelos dispersos. Las técnicas de aproximación de bajo rango que se han implementado proporcionan una forma de controlar la sensibilidad del modelo predictivo al ruido en los datos de entrenamiento. Como parte de esta tesis, se han desarrollado varias herramientas computacionales de propósito general para la identificación de modelos dispersos en ciencia e ingeniería.

Como se mencionó en la sección 1.1, el trabajo realizado en [11] ha demostrado que el algoritmo SINDy es eficaz para recuperar las ecuaciones que rigen los sistemas dinámicos dadas las simulaciones del sistema. El objetivo de este trabajo, es presentar la fortaleza de la novedosa herramienta SDSI para la identificación de sistemas epidemiológicos, mostramos cómo puede emplearse con CPM en un experimento de control de enfermedades infecciosas usando un modelo epidemiológico SEIR. Comparamos el rendimiento con el de los CPM basados en un modelo de descomposición en modo dinámica (DMD) y el algoritmo SINDy. También consideramos un experimento para sistemas dinámicos lineales, para ello se utilizó el modelo SIRD. En este caso comparamos SDSI con la programación cuadrática secuencial (SQP) y SINDy.

La importancia de este trabajo recae en la creciente intersección de los métodos basados en datos, sistemas dinámicos, optimización, estadística y otros temas de la matemática. La mayoría de los programas de pregrado en ingeniería y los campos de la ciencia en la academia,

la industria, el gobierno y entre otros tienen poca exposición a los métodos basados en datos y optimización, el objetivo es proporcionar una introducción a la ciencia de datos. Se ha elegido SDSI y SINDy por su relevancia, simplicidad y generalidad, esto ya que cada vez más los investigadores están utilizando enfoques basados en datos.

El descubrimiento basado en datos está revolucionando actualmente la forma en que modelamos, predecimos, identificamos y controlamos sistemas complejos. Los problemas científicos y de ingeniería más apremiantes de la era moderna no son basados en principios básicos. Cada vez más, los investigadores recurren a enfoques basados en datos para una amplia gama de sistemas complejos, como la turbulencia, el cerebro, el clima, la epidemiología, las finanzas, la robótica y la autonomía, entre otros [38]. Estos sistemas son típicamente no lineales, dinámicos, de escala múltiple en espacio y tiempo, de alta dimensión, con patrones dominantes que deben ser modelados para el objetivo eventual de detección, predicción, estimación y control [25, 60]. Por lo tanto, las soluciones a menudo se pueden aproximar a través de métodos de reducción de dimensionalidad donde si n es la dimensión del sistema original de alta dimensión y r es la dimensión del subespacio donde la dinámica está incorporada, entonces $r \ll n$ [62].

El trabajo presentado a continuación se ha organizado de la siguiente forma. En el capítulo 2 se presentan algunos preliminares que sirven de base para el desarrollo de este trabajo. La teoría y los algoritmos de los métodos SDSI y SINDy se describe en el Capítulo 3. Los experimentos y resultados computacionales se presentan en el capítulo 4. Por último en el Capítulo 5 realizamos las conclusiones más relevantes de este trabajo, así como también se describe los posibles trabajos futuros que se identificaron al hacer este estudio.

Capítulo 2

Preliminares

En este capítulo se presentan algunos preliminares de importancia para la comprensión de los capítulos y secciones siguientes de este trabajo.

2.1 Definiciones y terminología

En esta sección se presentan algunas definiciones y terminología que sirven de base para el desarrollo de este trabajo.

Definición 2.1. Cada vez que nos referimos a un sistema estaremos considerando un sistema dinámico en tiempo discreto que se puede describir como un par (Γ, Υ) determinado por un conjunto de estados $\Gamma \subset \mathbb{C}^n$, y una función $\Upsilon : \Gamma \rightarrow \Gamma$ que se llamará operador de transición, tal que para cualquier serie de tiempo determinada por una sucesión $\{x_t\}_{t \geq 1} \subset \Gamma$, tendremos que $\Upsilon(x_t) = x_{t+1}$.

Para sistemas cuyos espacios de estado están contenidos en \mathbb{R}^n consideraremos la identificación habitual de \mathbb{R} con la línea real en \mathbb{C} , y aplicaremos las técnicas de identificación de sistemas presentadas en este trabajo, considerando restricciones adecuadas cuando sea necesario.

Dado $\varsigma \in \mathbb{R}$, escribiremos W_ς para denotar la función $W_\varsigma : \mathbb{R} \rightarrow \mathbb{R}$ definida por la siguiente expresión:

$$W_\varsigma(x) = \begin{cases} 1, & x > \varsigma \\ 0, & x \leq \varsigma. \end{cases} \quad (2.1)$$

El símbolo $\mathbb{C}^{n \times m}$ denotará espacio de matrices complejas de tamaño $n \times m$, y escribiremos I_n para denotar la matriz identidad en $\mathbb{C}^{n \times n}$ y $\mathbf{0}_{n,m}$ para denotar la matriz cero en $\mathbb{C}^{n \times m}$. De ahora en adelante, dada una matriz $X = [X_{ji}] \in \mathbb{C}^{m \times n}$, escribiremos X^* para denotar la transpuesta conjugada de X determinada por $X^* = X^T = [x_{ij}]$ en $\mathbb{C}^{n \times m}$. Representaremos los vectores en \mathbb{C}^n como matrices columna en $\mathbb{C}^{n \times 1}$ y como n-tuplas.

Escribiremos $\bar{e}_{p,n}$ para denotar las matrices en $\mathbb{C}^{n \times 1}$ que representan la base canónica de \mathbb{C}^n (cada $\bar{e}_{p,n}$ es la p -ésima columna de I_n), que están determinadas por las expresiones

$$\bar{e}_{p,n} = [\delta_{1,p} \quad \delta_{2,p} \quad \cdots \quad \delta_{n-1,p} \quad \delta_{n,p}]^T \quad (2.2)$$

para cada $1 \leq p \leq n$, donde $\delta_{\mu,p}$ es la delta de Kronecker determinado por la expresión:

$$\delta_{\mu,p} = \begin{cases} 1, & \mu = p \\ 0, & \mu \neq p. \end{cases} \quad (2.3)$$

Proyección Ortogonal

- Una matriz $R \in \mathbb{C}^{n \times n}$ se llamará proyección ortogonal siempre que $R^2 = R = R^*$.
- Una matriz $Q \in \mathbb{C}^{m \times n}$ tal que las matrices QQ^* y Q^*Q son proyecciones ortogonales se denominará isometría parcial.

Matrices Unitarias

- Escribiremos $\mathbb{U}(n)$ para denotar el grupo de matrices unitarias en $\mathbb{C}^{n \times n}$ definido por la expresión $\mathbb{U}(n) = \{X \in \mathbb{C}^{n \times n} : X^*X = XX^* = I_n\}$.

Producto de Kronecker

- Dados $X = [x_{ji}] \in \mathbb{C}^{m \times n}$ y $Y = [y_{pk}] \in \mathbb{C}^{r \times q}$, escribiremos $X \otimes Y$ para denotar el producto de Kronecker definido por la expresión $X \otimes Y = [x_{ji}Y] \in \mathbb{C}^{(mr) \times (nq)}$.

Conjuntos

- Escribiremos \mathbb{Z}^+ para denotar el conjunto de enteros positivos $\mathbb{Z} \cap [1, \infty)$.
- Dado un conjunto S , escribiremos $\#(S)$ para denotar el número de elementos en S .

Normas

- Dado $x \in \mathbb{C}^n$ escribiremos $\|x\|$ para denotar la norma l_2 en \mathbb{C}^n determinada por $\|x\| = \sqrt{x^*x} = \left(\sum_{j=1}^n |x_j|^2\right)^{\frac{1}{2}}$.
- Dado $x \in \mathbb{C}^n$ escribiremos $\|x\|_1$ para denotar la norma l_1 en \mathbb{C}^n determinada por $\|x\|_1 = \sum_{j=1}^n |x_j|$.
- Dado $x \in \mathbb{C}^n$ escribiremos $\|x\|_\infty$ para denotar la norma l_∞ en \mathbb{C}^n determinada por la expresión $\|x\|_\infty = \max_{1 \leq j \leq n} |x_j|$ para cada $x \in \mathbb{C}^n$.
- Dado $X \in \mathbb{C}^{n \times m}$ escribiremos $\|X\|_F$ para denotar la norma de Frobenius de X definida por

$$\|X\|_F = \sqrt{\text{tr}(X^*X)}, \quad (2.4)$$

donde tr denota la traza de una matriz, definida para cualquier $A = [a_{ij}] \in \mathbb{C}^{n \times n}$ por la expresión.

$$\text{tr}(A) = \sum_{i=1}^n a_{ii}.$$

- Dado $X \in \mathbb{C}^{n \times m}$ escribiremos $\|X\|_2$ para denotar la norma espectral de X definida por

$$\|X\|_2 = \sigma_{\max}(X) \quad (2.5)$$

donde $\sigma_{\max}(X)$ representa el valor singular más grande de la matriz X .

Matriz de trayectoria tipo Hankel

Dado un conjunto finito de vectores $\Gamma_T = \{x_1, \dots, x_T\} \subset \mathbb{C}^n$ escribiremos $\mathbb{H}_L(\Gamma_T)$ para denotar la matriz de trayectoria tipo Hankel en $\mathbb{C}^{nL \times (T-L+1)}$ definida por la siguiente expresión:

$$\mathbb{H}_L(\Gamma_T) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_{T-L+1} \\ x_2 & x_3 & x_4 & \cdots & x_{T-L+2} \\ x_3 & x_4 & x_5 & \cdots & x_{T-L+3} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ x_L & x_{L+1} & x_{L+2} & \cdots & x_T \end{bmatrix}.$$

Dada cualquier matriz $A \in \mathbb{C}^{m \times n}$ escribiremos $\text{rk}(A)$ para denotar el rango de A , que corresponde al número máximo de columnas linealmente independientes de A .

Para más detalles sobre las nociones antes mencionadas, se remite al lector a [25, 29].

2.2 Descomposición en Valores Singulares (SVD)

Como en este trabajo la información que necesitamos requiere ser almacenada en una matriz A , que a su vez el propósito principal es identificar la información de dicha matriz, se hará uso de la técnica de SVD. En términos generales la SVD es una factorización matricial que nos proporciona ciertas características de la matriz A , también es una generalización del teorema espectral [29], ya que en este caso podemos hacer la descomposición para cualquier matriz rectangular y no simétrica, nos permite describir el mapeo lineal asociado a la matriz A mediante el sistema $Ax = b$.

2.2.1 Descripción y definición

Definición 2.2. En términos generales para cualquier matriz $A \in \mathbb{C}^{n \times m}$ se puede obtener una factorización matricial de la forma:

$$A = U\Sigma V^*,$$

donde $U \in \mathbb{C}^{n \times n}$, $V \in \mathbb{C}^{m \times m}$ son matrices unitarias y $\Sigma \in \mathbb{R}^{n \times m}$ es diagonal y tienen las siguientes características:

- $U = \{u_i\}$: donde u_i son las columnas definidas como vectores singulares a la izquierda, con r vectores linealmente independientes y ortonormales en el espacio \mathbb{C}^n .
- $V = \{v_i\}$: donde v_i son las columnas definidas como vectores singulares a la derecha, con r vectores linealmente independientes y ortonormales en el espacio \mathbb{C}^m .
- $\Sigma = \text{diag}(\sigma_i)$: cuyas entradas σ_i , definidas como valores singulares son no negativas y decrecientes, es decir $\sigma_1 \geq \dots \geq \sigma_r \geq 0$.

Se tiene que $r = \text{rank}(A)$, esto equivale al Teorema 2.6.3 de [29].

2.2.2 Existencia y unicidad

Cada matriz tiene una SVD única, como se muestra en [60] en el siguiente teorema:

Teorema 2.3. Toda matriz $A \in \mathbb{C}^{n \times m}$ tiene una descomposición en valores singulares. Además, los valores singulares $\{\sigma_i\}$ se determinan de forma única, si A es cuadrada y los σ_i son distintos, las columnas de U y V son únicas a menos del signo (es decir, un factor $e^{i\theta}$).

2.2.3 Propiedades de la matriz

Una vez obtenida la SVD es posible acceder a una gran cantidad de información de la matriz A las cuales se proporcionarán en esta sección. También podemos resolver sistemas de ecuaciones que consideren a la matriz A como la matriz de coeficientes y es importante en algunos problemas de optimización, estos problemas nos ayudarán a obtener una mejor aproximación de “low-rank”.

Entre las propiedades que se pueden extraer de la SVD se tiene las siguientes (ver [29]):

- El rank de A es r , el número de valores singulares distintos de cero.
- $R(A) = \{u_1, \dots, u_r\}$ y $N(A) = \{v_{r+1}, \dots, v_n\}$.
- $\|A\|_2 = \sigma_1$ y $\|A\|_F = \sqrt{\sigma_1^2 + \dots + \sigma_r^2}$.
- Los valores singulares de A distintos de cero son las raíces cuadradas de los valores propios distintos de cero de A^*A o AA^* (estas matrices tienen los mismos valores propios distintos de cero).
- Para $A \in \mathbb{C}^{m \times m}$, $|\det(A)| = \prod_{i=1}^m \sigma_i$.

Ejemplo del cálculo de la SVD

Encontrar la SVD de A , es decir $U\Sigma V^T$, donde $A = \begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{bmatrix}$.

Solución: Primero calculamos los valores singulares σ_i encontrando los valores propios de AA^T .

$$AA^T = \begin{bmatrix} 17 & 8 \\ 8 & 17 \end{bmatrix}.$$

El polinomio característico es $\det(AA^T - \lambda I) = \lambda^2 - 34\lambda + 225 = (\lambda - 25)(\lambda - 9)$, por lo que los valores singulares son $\sigma_1 = \sqrt{25} = 5$ y $\sigma_2 = \sqrt{9} = 3$.

Ahora encontramos los vectores singulares de la derecha (las columnas de V) encontrando un conjunto ortonormal de vectores propios de $A^T A$. También es posible proceder encontrando los vectores singulares de la izquierda (columnas de U) en su lugar. Los valores propios de $A^T A$ son 25, 9 y 0, y dado que $A^T A$ es simétrica, sabemos que los vectores propios serán ortogonales.

- Para $\lambda = 25$, tenemos

$$A^T A - 25I = \begin{bmatrix} -12 & 12 & 2 \\ 12 & -12 & -2 \\ 2 & -2 & -17 \end{bmatrix},$$

que se reduce a $\begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$. Un vector de longitud unitaria en el núcleo de esa matriz

es $v_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}$.

- Para $\lambda = 9$, tenemos

$$A^T A - 9I = \begin{bmatrix} 4 & 12 & 2 \\ 12 & 4 & -2 \\ 2 & -2 & -1 \end{bmatrix},$$

que se reduce a $\begin{bmatrix} 1 & -1 & -\frac{1}{4} \\ 0 & 1 & \frac{1}{4} \\ 0 & 0 & 0 \end{bmatrix}$. Un vector de longitud unitaria en el núcleo de esa

matriz es $v_2 = \begin{bmatrix} \frac{1}{\sqrt{18}} \\ -\frac{1}{\sqrt{18}} \\ \frac{4}{\sqrt{18}} \end{bmatrix}$.

- Para el último vector propio, podríamos calcular el núcleo de $A^T A$ o encontrar un vector unitario perpendicular a v_1 y v_2 . Ser perpendicular a $v_1 = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$ necesitamos $-a = b$. Entonces la condición de que $v_2^T v_3 = 0$ se convierte $\frac{2a}{\sqrt{18}} + \frac{4c}{\sqrt{18}} = 0$ o $-a = 2c$.

Así que $v_3 = \begin{bmatrix} a \\ -a \\ \frac{-a}{2} \end{bmatrix}$ y para que sea de longitud unitaria necesitamos $a = \frac{2}{3}$ entonces

$v_3 = \begin{bmatrix} \frac{2}{3} \\ -\frac{2}{3} \\ -\frac{1}{3} \end{bmatrix}$. Así que se tiene que

$$A = U \Sigma V^T = U \begin{bmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{18}} & -\frac{1}{\sqrt{18}} & \frac{4}{\sqrt{18}} \\ \frac{2}{3} & -\frac{2}{3} & -\frac{1}{3} \end{bmatrix}.$$

Finalmente, podemos calcular U por la fórmula $\sigma u_i = A v_i$ o $u_i = \frac{1}{\sigma} A v_i$. Esto da

$U = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$. Entonces, la SVD es:

$$A = U\Sigma V^T = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{18}} & -\frac{1}{\sqrt{18}} & \frac{4}{\sqrt{18}} \\ \frac{2}{3} & -\frac{2}{3} & -\frac{1}{3} \end{bmatrix}.$$

[Matlab/Octave]. En Matlab R2021a/GNU Octave 2020 (5.2.0), se puede encontrar la SVD de la siguiente manera:

```
>> A=[3 2 2;2 3 -2] % Crea una matriz de datos aleatorios de 3x2
```

```
A =
```

```
3 2 2
2 3 -2
```

```
>> [U,S,V]=svd(A) % Descomposicion en Valores Singulares
```

```
U =
```

```
-0.70711 -0.70711
-0.70711 0.70711
```

```
S =
```

```
Diagonal Matrix
```

```
5.0000 0 0
0 3.0000 0
```

```
V =
```

```
-7.0711e-01 -2.3570e-01 -6.6667e-01
-7.0711e-01 2.3570e-01 6.6667e-01
-5.5511e-17 -9.4281e-01 3.3333e-01
```

Para matrices A no cuadradas, la SVD económica es más eficiente, esto ya que la descomposición de tamaño económico elimina filas o columnas adicionales de ceros de la matriz diagonal de valores singulares, S , junto con las columnas en U o V que multiplican esos ceros en la expresión $A = USV'$. La eliminación de estos ceros y columnas puede mejorar el tiempo de ejecución y reducir los requisitos de almacenamiento sin comprometer la precisión de la descomposición.

```

% Ejemplo anterior utilizando la svd económica
>> [U,S,V]=svd(A,'econ') % SVD economica
U =

-0.70711  -0.70711
-0.70711   0.70711

S =

Diagonal Matrix

5.0000      0
0          3.0000

V =

-7.0711e-01  -2.3570e-01
-7.0711e-01   2.3570e-01
-5.5511e-17  -9.4281e-01

```

En la siguiente sección se presenta la descomposición ortogonal adecuada (POD), que establece los conceptos de reducción del modelo que capitaliza la existencia de dinámicas de bajo rango en el sistema, para obtener funciones de base óptimas que abarcan un subespacio de menor dimensión.

2.3 Descomposición Ortogonal Adecuada (POD)

La motivación para presentar la técnica de descomposición ortogonal adecuada (POD) en el contexto de esta tesis es introducir el concepto del cálculo de modelos de orden reducido y métodos basados en mediciones. Aunque en el enfoque de esta tesis no es en esta técnica, se presenta una breve descripción del método. En esta sección, seguimos de cerca [10, 46].

La POD en términos generales, es una técnica para encontrar de manera óptima una base que abarca un conjunto de datos recopilados a partir de una simulación experimental o numérica de un sistema dinámico. Este método se utiliza a menudo en el desarrollo de modelos de baja dimensión. La idea es, dado un conjunto de datos que se encuentra en un espacio vectorial V , encontrar un subespacio V_r de dimensión fija r tal que se minimice el error de la proyección de tal conjunto en el subespacio

2.3.1 Optimalidad de la base POD

Suponga que tenemos un conjunto de datos $x(t) \in \mathbb{R}^n$, con $0 \leq t \leq T$. Se busca una proyección $P_r : \mathbb{R}^n \rightarrow \mathbb{R}^n$ de rango fijo r que minimice el error total

$$\int_0^T \|x(t) - P_r x(t)\|^2 dt.$$

Para resolver este problema, se introduce la matriz

$$Q = \int_0^T x(t)x(t)^* dt,$$

encontrando los valores y vectores propios de Q , dados por

$$Q\mu_j = \lambda_j\mu_j, \quad \lambda_1 \geq \dots \geq \lambda_n \geq 0.$$

Dado que Q es simétrica y semidefinida positiva entonces todos los valores propios λ_j son reales y no negativos, además los vectores propios μ_j pueden elegirse para que sean ortonormales [29]. Los vectores μ_j se denominan modos de la descomposición ortogonal adecuada (POD). El subespacio óptimo de dimensión r está generado por $\{\mu_1, \dots, \mu_r\}$, y la proyección óptima P_r está dada por

$$P_r = \sum_{j=1}^r \mu_j \mu_j^*.$$

2.3.2 Cálculo de modelos de orden reducido

Habiendo determinado P_r , suponga ahora que se está determinando las soluciones de un sistema descrito por las ecuaciones

$$\dot{\mathbf{x}}(\mathbf{w}, t) = \mathbf{f}(\mathbf{x}(\mathbf{w}, t)), \quad (2.6)$$

donde $\mathbf{x}(\mathbf{w}, t)$ es el estado del sistema en el tiempo t y \mathbf{w} la distribución espacial sobre algún dominio Ω , y \mathbf{f} es un campo vectorial que mapea la variedad uniforme $\mathcal{M} \subset \mathbb{R}^n$ en sí misma. Para capitalizar los modos POD, uno puede formar modelos de orden reducido usando la proyección de Galerkin (ver, por ejemplo, [19]), que especifica $\dot{x}(t) = P_r \mathbf{f}(x(t))$, es decir, proyectando el campo vectorial original \mathbf{f} en el subespacio r -dimensional. Luego escribimos

$$\mathbf{x}_r(t) = \sum_{j=1}^r \xi_j(t) \mu_j. \quad (2.7)$$

Sustituyendo la ecuación 2.7 en 2.6 y multiplicando por μ_j^* , se obtiene

$$\dot{\xi}_j(t) = \mu_j^* f(\mathbf{w}(t)), \quad j = 1, \dots, r$$

el cual es un conjunto de r ecuaciones diferenciales ordinarias (EDO) que describe la evolución de $\mathbf{w}(t)$. En otras palabras, la determinación de soluciones a 2.6 que antes implicaba resolver un conjunto de n EDO, ahora se reduce a un conjunto de r EDO.

2.3.3 Métodos basados en mediciones

Al analizar una serie de tiempo de datos en una cuadrícula espacial, a menudo es beneficioso utilizar métodos basados en mediciones. Los datos del campo vectorial bidimensional o tridimensional en el tiempo t_k se reorganizan luego en un vector de una sola columna. Si consideramos un conjunto discreto de mediciones $\{\mathbf{x}(t_k)\}_{k=0}^p \in \mathbb{R}^n$ generado por algún proceso físico descrito por las ecuaciones $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k)$, el conjunto de mediciones se puede expresar como la siguiente matriz,

$$X = \begin{pmatrix} x_{0,1} & x_{1,1} & \cdots & x_{p,1} \\ x_{0,2} & x_{1,2} & \cdots & x_{p,2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{0,n} & x_{1,n} & \cdots & x_{p,n} \end{pmatrix}.$$

En este caso, la POD se puede formular utilizando la SVD de la matriz X , es decir, mediciones de X ,

$$X = U\Sigma V^T,$$

donde $U \in \mathbb{C}^{n \times n}$, $V \in \mathbb{C}^{p \times p}$ y $\Sigma \in \mathbb{R}^{n \times p}$. U y V son matrices unitarias y Σ es una matriz rectangular diagonal con valores singulares positivos $\{\sigma_1, \dots, \sigma_r\}$, donde r denota el número de valores singulares positivos definidos en 2.2.

Los vectores columna en $U = \{u_1, \dots, u_n\}$ y $V = \{\nu_1, \dots, \nu_p\}$ contienen los vectores propios ortogonales de XX^T y $X^T X$, respectivamente, como en

$$\begin{aligned} X = U\Sigma V^T &\Rightarrow XX^T = U\Sigma\Sigma^T U^T \Rightarrow XX^T U = UD, \\ X^T &= V\Sigma^* U^T \Rightarrow X^T X = V\Sigma^T \Sigma V^T \Rightarrow X^T X V = VD, \end{aligned}$$

donde $D = \Sigma\Sigma^T = \Sigma^T \Sigma = \sum_{k=1}^p \lambda_k$.

Los valores singulares de la matriz de mediciones X se asocian luego con los valores propios de las matrices $X^T X$ y XX^T mediante la relación $\lambda_k = \sigma_k^2$.

Dado que XX^T es simétrica y semidefinida positiva, todos los valores propios λ_k son reales y no negativos, y en virtud de las propiedades de la SVD, los vectores propios en U son ortonormales.

El resultado principal es que el subespacio óptimo de POD de dimensión m se amplía por $\{u_1, \dots, u_m\}$, y la proyección óptima está dada por $P_r = \sum_{k=1}^p \mu_k \mu_k^T$. Los vectores base, reescritos como $U_r = (u_1, \dots, u_r)$, se denominan modos POD, donde $r \leq \min(n, p)$ es el número de valores singulares distintos de cero, correspondientes al rango de X .

2.3.4 Reducción de dimensión

Ahora establecemos nuestro objetivo para determinar un subespacio óptimo de V que tenga la dimensión más baja posible $r \ll m$, mientras se mantiene una buena aproximación al conjunto de datos original. En otras palabras, se busca un sistema de orden reducido tal que

la solución exacta de \mathbf{x}_k pueda aproximarse mediante una combinación lineal de r vectores base.

Como se propone en [43], se mide la aproximación utilizando el contenido de información relativo referido como energía, definida como

$$I(r) = \frac{\sum_{i=1}^r \sigma_i^2}{\sum_{j=1}^m \sigma_j^2}.$$

El objetivo es elegir el r más pequeño de modo que $I(r)$ todavía esté lo suficientemente cerca de 1. Por ejemplo, si el subespacio U_r debe contener un porcentaje β de la información en U , entonces se debe elegir r tal que

$$r = \min_r \left(\frac{\beta}{100} \leq I(r) \right).$$

En esta sección se observó que el método POD identifica una base ortogonal óptima de los modos U y V correlacionados en el tiempo y espacio, respectivamente, capitalizando el método SVD de descomponer una matriz de datos. Es esencialmente una técnica de reducción de modelos y se basa en el supuesto de que la evolución en el tiempo de la dinámica del sistema se rige por un número reducido de modos dominantes.

Asociado con la proyección de Galerkin, la dimensión del sistema de ecuaciones gobernantes que tenemos que resolver para determinar su solución puede reducirse en gran medida. Sin embargo, incluso para una dimensión reducida r , estos sistemas pueden resultar costosos de simular.

Para conectar la POD al método de descomposición en modo dinámico (DMD), veremos que la DMD no solo proporciona los modos del sistema, como POD, sino que también asocia estos modos espaciales correlacionados con una frecuencia temporal y una posible tasa de crecimiento o decadencia. En la siguiente sección se presenta la descomposición en modo dinámico, su definición y algoritmo.

2.4 Descomposición en Modo Dinámico (DMD)

El método DMD tiene una rica historia, que va desde los primeros trabajos de desarrollo de los fundamentos teóricos hasta los trabajos modernos, este método se originó en la comunidad de dinámica de fluidos como una técnica basada en principios para descomponer flujos complejos en una representación simple basada en estructuras coherentes espacio-temporales de bajo rango. Schmid en [54] primero definió el algoritmo DMD y demostró su capacidad para proporcionar información físicamente interpretable a partir de datos de fluidos de alta dimensión, el creciente éxito de DMD se debe al hecho de que es un método libre de ecuaciones y basado en datos, capaz de proporcionar una descomposición precisa de un sistema complejo en estructuras coherentes espacio-temporales que pueden usarse para análisis de diagnóstico, estado futuro a corto plazo, identificación y control. Es importante destacar que Rowley en [50] mostró que DMD está conectada a la dinámica no lineal a través de la teoría del operador de Koopman (ver [36]) y es interpretable usando técnicas de sistemas dinámicos estándar [46]. Específicamente, el algoritmo DMD es una manifestación de la

teoría de Koopman cuando las funciones observables son la identidad o transformaciones lineales del espacio de estados. Por tanto, DMD es una arquitectura algorítmica basada en principios que permite una aproximación explícita del operador de Koopman.

2.4.1 Descripción y definición

El método DMD proporciona una descomposición espacio-temporal de datos en un conjunto de modos dinámicos que se derivan de mediciones de un sistema dado en el tiempo. Las matemáticas detrás de la extracción de información dinámica resuelta en el tiempo están estrechamente relacionadas con la idea del algoritmo de Arnoldi [54], uno de los caballos de batalla de los solucionadores computacionales rápidos. El proceso de recolección de datos involucra dos parámetros:

n : número de puntos espaciales guardados por mediciones de tiempo

m : número de mediciones tomadas.

Definición 2.4. (DMD en [9]) Si se considera un sistema dinámico descrito por la ecuación $\frac{d\mathbf{x}}{dt} = f(\mathbf{x}, t)$ y dos conjuntos de datos,

$$\begin{aligned} X &= [x_1, x_2, \dots, x_{m-1}] \\ X' &= [x_2, x_3, \dots, x_m], \end{aligned}$$

la DMD calcula la descomposición espectral principal (es decir valores y vectores propios) del operador lineal de mejor ajuste A que relaciona los datos:

$$X' \approx AX,$$

el operador A de mejor ajuste establece entonces un sistema dinámico lineal que mejor hace avanzar las mediciones en el tiempo. Si suponemos un muestreo uniforme en el tiempo, esto se convierte en:

$$x_{k+1} \approx Ax_k.$$

Matemáticamente, el operador de mejor ajuste de A se define de la siguiente forma:

$$A = \operatorname{argmin}_A \|X' - AX\|_F = X'X^\dagger, \quad (2.8)$$

donde $\|\cdot\|_F$ es la norma de Frobenius y \dagger denota la pseudoinversa.

2.4.2 El algoritmo de DMD

En la práctica, cuando la dimensión de estado n es grande, la matriz A puede resultar difícil de analizar directamente. En cambio, DMD evita la descomposición propia de A al considerar una representación de rango reducido en términos de una matriz \tilde{A} proyectada por una POD. El algoritmo DMD, está dado por los siguientes pasos [9]:

1. Calculamos la descomposición en valores singulares (SVD) de $X \in \mathbb{C}^{n \times m}$ [60]:

$$X \approx \tilde{U} \tilde{\Sigma} \tilde{V}^*, \quad (2.9)$$

donde $*$ denota la transpuesta conjugada, $\tilde{U} \in \mathbb{C}^{n \times r}$, $\tilde{\Sigma} \in \mathbb{C}^{r \times r}$ y $\tilde{V} \in \mathbb{C}^{m \times r}$ con $r \leq m$. Las columnas de U son ortonormales es decir satisfacen que $\tilde{U}^* \tilde{U} = I$, de manera similar $\tilde{V}^* \tilde{V} = I$. La reducción de SVD en 2.9 se aprovecha en esta etapa del algoritmo para realizar un truncamiento de “low-rank” de los datos. Específicamente, si la estructura de baja dimensión está presente en los datos, los valores singulares de $\tilde{\Sigma}$ disminuirán bruscamente a cero con quizás solo un número limitado de modos dominantes.

2. La matriz A de 2.8 se puede obtener utilizando la pseudoinversa de X a través de la SVD:

$$A = X' \tilde{V} \tilde{\Sigma}^{-1} \tilde{U}^*.$$

En la práctica, es más eficiente computacionalmente calcular \tilde{A} , la proyección $r \times r$ de la matriz completa A , es decir solo estamos interesados en los r valores propios y los vectores propios principales de A , por lo que podemos proyectar A en los modos POD en U :

$$\tilde{A} = U^* X' V \Sigma^{-1} = U^* A U.$$

La matriz \tilde{A} define un modelo lineal de baja dimensión del sistema dinámico en coordenadas POD:

$$\tilde{x}_{k+1} = \tilde{A} \tilde{x}_k.$$

Notar que es posible reconstruir el estado de alta dimensión $x_k = \tilde{U} \tilde{x}_k$.

3. Calculamos la descomposición espectral de \tilde{A} :

$$\tilde{A} W = W D,$$

donde las entradas de la matriz diagonal D son los valores propios de la DMD, que también corresponden a los valores propios de la matriz A completa, las columnas de W son vectores propios de \tilde{A}

4. Podemos reconstruir la descomposición espectral de A a partir de W y D . En particular, los valores propios de A están dados por D y los vectores propios de A (modos DMD) están dados por columnas de la matriz:

$$\Phi = X' \tilde{V} \Sigma^{-1} W.$$

En la siguiente sección se presenta DMD en observables no lineales donde se postulan conceptos fundamentales detrás del operador de Koopman y será de mucha importancia para el desarrollo del experimento computacional 4.2, en el cual se toma la ecuación no lineal de Schrödinger (NLS) para demostrar que la descomposición de Koopman puede mejorar en gran medida la identificación y la predicción del estado futuro de los datos, donde un observable adecuado es importante para que esto suceda.

2.4.3 DMD en observables no lineales

La aproximación del operador de Koopman a través de DMD es de vital importancia para permitir la evaluación del operador a partir de los datos. Se destaca el importante papel desempeñado por los observables y sus variedades de evolución asociadas. En particular, se puede pensar que las ecuaciones diferenciales parciales no lineales evolucionan en variedades que a menudo son difíciles de caracterizar y rara vez se conocen analíticamente. Una elección correcta de observables puede, en algunos casos, linealizar la variedad no lineal. Independientemente, los observables elegidos con criterio pueden ayudar a transformar una ecuación diferencial parcial que evoluciona en una variedad fuertemente no lineal a una variedad débilmente no lineal, lo que permite un rango de aplicabilidad más preciso y más amplio de la aproximación de Koopman.

Haciendo uso de la teoría espectral, podemos proceder a postular el concepto fundamental detrás del operador de Koopman. El trabajo original de Koopman en [36] consideró los sistemas hamiltonianos y formuló el operador de Koopman en tiempo discreto; sin embargo, comenzamos con tiempo continuo y luego derivamos la formulación asociada de tiempo discreto.

Definición 2.5. (Operador de Koopman). Se considera un sistema dinámico de tiempo continuo descrito por la ecuación

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}), \quad (2.10)$$

donde $\mathbf{x} \in \mathcal{M}$ es un estado en una variedad \mathcal{M} suave de n dimensiones. El operador de Koopman \mathcal{K} es un operador lineal de dimensión infinita que actúa sobre todas las funciones observables $g : \mathcal{M} \rightarrow \mathbb{C}$, es decir

$$\mathcal{K}g(\mathbf{x}) = g(\mathbf{f}(\mathbf{x})).$$

Poco después del artículo original, Koopman y von Neumann extendieron estos resultados a sistemas dinámicos con espectros continuos en [37].

Teorema 2.6. (Descomposición en modo Koopman y DMD). Sea Φ_k una función propia de \mathcal{K} con valor propio λ_k , y suponga que $\Phi_k \in \text{span}\{g_j\}$, de modo que

$$\Phi_k(x) = w_1 g_1(\mathbf{x}) + w_2 g_2(\mathbf{x}) + \cdots + w_q g_q(\mathbf{x}) = \mathbf{w} \cdot \mathbf{g}$$

para algunos $\mathbf{w} = [w_1, w_2, \dots, w_q]^T \in \mathbb{C}^q$. Si $\mathbf{w} \in R(Y)$, donde $Y = g(X)$ para la matriz X de la definición 2.4 y R es el rango entonces \mathbf{w} es un vector propio de A_Y con valor propio λ_k de modo que $\tilde{\mathbf{w}}^* A_Y = \lambda_k \tilde{\mathbf{w}}^*$.

2.4.4 El algoritmo de Koopman

A continuación se muestra cómo utilizar los datos y los observables para producir un operador de Koopman y una predicción del estado futuro de la evolución no lineal 2.10. Debería compararse con el algoritmo DMD. Esencialmente, el algoritmo de Koopman simplemente aplica DMD en el espacio de los observables, el cual está dado por los siguientes pasos:

1. Primero, a partir de las matrices de datos X y X' , creamos las matrices de datos de los observables Y y Y' :

$$Y = \begin{bmatrix} | & | & \cdots & | \\ \mathbf{g}(\mathbf{x}_1) & \mathbf{g}(\mathbf{x}_2) & \cdots & \mathbf{g}(\mathbf{x}_{m-1}) \\ | & | & \cdots & | \end{bmatrix}$$

$$Y' = \begin{bmatrix} | & | & \cdots & | \\ \mathbf{g}(\mathbf{x}'_1) & \mathbf{g}(\mathbf{x}'_2) & \cdots & \mathbf{g}(\mathbf{x}'_{m-1}) \\ | & | & \cdots & | \end{bmatrix},$$

donde cada columna es dada por $y_k = \mathbf{g}(\mathbf{x}_k)$ o $y'_k = \mathbf{g}(\mathbf{x}'_k)$

2. Utilizamos el algoritmo DMD para calcular

$$A_Y = Y'Y^\dagger,$$

junto con la contraparte de rango bajo \tilde{A}_Y . Los valores propios y vectores propios de A_Y pueden aproximarse a los valores propios y modos de Koopman, dependiendo del conjunto de observables elegidos.

3. DMD se puede utilizar para calcular los modos Φ_Y , que pueden aproximarse a los modos Koopman (ver sección 2.4.2)

$$\Phi_Y = Y'V\Sigma^{-1}W,$$

donde W proviene del problema de valores propios $\tilde{A}_Y W = W D$ y $Y = U\Sigma V^*$.

4. El estado futuro en el espacio de los observables viene dado por la evolución lineal

$$y(t) = \Phi_Y \text{diag}(\exp(\omega t)) b$$

donde $b = \Phi_Y^+ y_1$ se determina proyectando hacia los datos iniciales observables, y ω es el conjunto de valores propios λ_k generados a partir de la matriz D , donde $\omega_k = \ln(\lambda_k)/\Delta t$.

5. Transformación de observables al nuevo espacio de estado:

$$y_k = \mathbf{g}(\mathbf{x}_k) \implies x_k = \mathbf{g}^{-1}(y_k).$$

Este último paso puede ser trivial si uno de los observables seleccionados para comprender $g(x_k)$ es la propia variable de estado x_k . Si solo se eligen observables no lineales de x_k , entonces el proceso de inversión puede ser difícil.

Este proceso muestra que el algoritmo DMD está estrechamente relacionado con el operador de Koopman. De hecho, es la pieza fundamental para la evaluación práctica del operador de Koopman de dimensión finita.

En el siguiente capítulo se presentan los métodos SDSI y SINDy sus conceptos principales y algoritmos para la identificación dispersa de sistemas lineales y no lineales.

Capítulo 3

Métodos

3.1 Identificación dispersa de dinámicas no lineales (SINDy)

El trabajo realizado en esta sección se basa en los métodos de mínimos cuadrados lineales dispersos descritos en [11]. Dados los avances tanto en la detección comprimida [18] como en la regresión dispersa [30], se ha vuelto computacionalmente factible extraer la dinámica de los sistemas a partir de grandes conjuntos de datos multimodales. Estas técnicas se basan en gran medida en el hecho de que muchos sistemas dinámicos se pueden representar mediante ecuaciones que son dispersas en el espacio de todas las funciones. En esta sección nos enfocamos en sistemas dinámicos que vienen dados por un sistema de ecuaciones diferenciales ordinarias de la forma

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)), \quad (3.1)$$

donde el vector $\mathbf{x}(t) = [x_1(t), \dots, x_n(t)] \in \mathbb{R}^n$ representa el estado del sistema en el tiempo t , y la función no lineal $\mathbf{f}(\mathbf{x}(t))$ representa las restricciones dinámicas que definen las ecuaciones de movimiento del sistema. Es decir, en muchos sistemas de interés, la función \mathbf{f} a menudo consta de solo unos pocos términos, lo que la hace dispersa en el espacio de todas las funciones, luego buscamos aproximar \mathbf{f} mediante un modelo lineal generalizado de la siguiente forma

$$\mathbf{f}(\mathbf{x}) \approx \sum_{i=1}^p \theta_i(\mathbf{x})\xi_i = \Theta(\mathbf{x})\xi,$$

con el menor número posible de términos distintos de cero en ξ . Entonces es posible resolver los términos relevantes que están activos en la dinámica usando regresión dispersa [30]. Ahora, los datos de series de tiempo se recopilan de 3.1 y se forman dos grandes matrices de datos:

$$X = \begin{pmatrix} \mathbf{x}^T(t_1) \\ \mathbf{x}^T(t_2) \\ \vdots \\ \mathbf{x}^T(t_m) \end{pmatrix} = \begin{pmatrix} x_1(t_1) & x_2(t_1) & \cdots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \cdots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \cdots & x_n(t_m) \end{pmatrix}, \quad (3.2)$$

$$\dot{X} = \begin{pmatrix} \dot{\mathbf{x}}^T(t_1) \\ \dot{\mathbf{x}}^T(t_2) \\ \vdots \\ \dot{\mathbf{x}}^T(t_m) \end{pmatrix} = \begin{pmatrix} \dot{x}_1(t_1) & \dot{x}_2(t_1) & \cdots & \dot{x}_n(t_1) \\ \dot{x}_1(t_2) & \dot{x}_2(t_2) & \cdots & \dot{x}_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{x}_1(t_m) & \dot{x}_2(t_m) & \cdots & \dot{x}_n(t_m) \end{pmatrix}, \quad (3.3)$$

donde $X, \dot{X} \in \mathbb{R}^{m \times n}$. En la práctica, esto puede calcularse directamente a partir de los datos en X ; para datos ruidosos, la derivada regularizada de variación total tiende a proporcionar derivadas numéricamente robustas [15]. Alternativamente, es posible formular el algoritmo SINDy para sistemas de tiempo discreto $\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k)$, como en el algoritmo DMD, y evitar las derivadas por completo. Se puede construir una biblioteca de funciones no lineales candidatas $\Theta(X) \in \mathbb{R}^{m \times p}$ a partir de los datos en X . Por ejemplo, $\Theta(X)$ puede constar de términos constantes, polinomiales y trigonométricos:

$$\Theta(X) = \begin{pmatrix} | & | & \cdots & | & \cdots & | & | & | & | & \cdots \\ 1 & X & \cdots & X^d & \cdots & \sin(X) & \cos(X) & \sin(2X) & \cos(2X) & \cdots \\ | & | & & | & & | & | & | & | & \end{pmatrix}$$

Aquí, la matriz X^d denota una matriz con vectores columna dados por todas las posibles series de tiempo de polinomios de d-ésimo grado en el estado \mathbf{x} dado por:

$$X^d = \begin{pmatrix} x_1^d(t_1) & x_1(t_1)x_2(t_1) & \cdots & x_2^d(t_1) & x_2(t_1)x_3(t_1) & \cdots & x_n^d(t_1) \\ x_1^d(t_2) & x_1(t_2)x_2(t_2) & \cdots & x_2^d(t_2) & x_2(t_2)x_3(t_2) & \cdots & x_n^d(t_2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_1^d(t_m) & x_1(t_m)x_2(t_m) & \cdots & x_2^d(t_m) & x_2(t_m)x_3(t_m) & \cdots & x_n^d(t_m) \end{pmatrix}.$$

El sistema dinámico en 3.1 ahora se puede representar en términos de las matrices de datos en 3.2 y 3.3 como

$$\dot{X} = \Theta(X)\Xi, \quad (3.4)$$

donde la matriz desconocida $\Xi = [\xi_1 \ \cdots \ \xi_n] \in \mathbb{R}^{p \times n}$ y cada columna ξ_i es un vector de coeficientes que determinan los términos activos en la i -ésima fila en 3.1. Un modelo parsimonioso proporcionará un ajuste de modelo preciso en 3.4 con el menor número posible de términos. Dicho modelo puede identificarse utilizando una regresión dispersa:

$$\xi_i = \operatorname{argmin}_{\xi'_i} \|\dot{X}_i - \Theta(X)\xi'_i\| + \lambda \|\xi'_i\|_1.$$

El término $\|\cdot\|_1$ promueve la dispersión en el vector de coeficientes ξ_i . Esta optimización se puede resolver mediante el procedimiento de mínimos cuadrados con truncado secuencial (STLS) [11] (ver Alg. 1).

3.1.1 Algoritmo

En esta sección se presenta el algoritmo de mínimos cuadrados con truncado secuencial (STLS), donde las condiciones generales para la unicidad de la solución relajada de la norma

l_1 se proporcionan en [61]. En la práctica, es posible que estas condiciones no se cumplan fácilmente y que se produzcan descubrimientos falsos, aunque pueden evitarse bajo ciertas condiciones [57]. Las condiciones específicas bajo las cuales converge el algoritmo de STLS en SINDy se proporcionan en [67]. La convergencia y recuperación de los datos se han explorado en un marco generalizado para la regresión regularizada relajada dispersa en [68], para la cual SINDy constituye un caso especial. Las condiciones bajo las cuales se puede recuperar la estructura de un modelo a partir de los datos de entrada y salida también se han examinado en el contexto de la identificabilidad [1].

Algorithm 1 STLS: Mínimos cuadrados con truncado secuencial [11].

Entrada: $\Theta \in \mathbb{R}^{m \times p}$, $\dot{X} \in \mathbb{R}^{m \times n}$, $\lambda > 0$, $N \in \mathbb{Z}^+$

Salida: $\Xi = \text{STLS}(\Theta, \dot{X}, \lambda, N)$

```

1:  $\Xi^0 \leftarrow \Theta^\dagger \dot{X}$  //  $\dagger$  denota la pseudoinversa
2: mientras no converge hacer
3:    $k \leftarrow k + 1$ 
4:    $I_{\text{small}} \leftarrow (\text{abs}(\Xi) < \lambda)$ 
5:    $\Xi^k(I_{\text{small}}) \leftarrow 0$ 
6:   para todas las variables hacer
7:      $I_{\text{big}} \leftarrow I_{\text{small}}(:, ii)$ 
8:      $\Xi^k(I_{\text{big}}, ii) \leftarrow (\Theta(:, I_{\text{big}}, ii))^\dagger \dot{X}(:, ii)$ 
9:   fin para
10: fin mientras
11: devolver  $\Xi$ 

```

El parámetro λ se selecciona para identificar el modelo óptimo de Pareto que mejor equilibra la complejidad del modelo con la precisión. Se realiza un barrido aproximado de λ para identificar el orden aproximado de magnitud donde se eliminan los términos y donde el error comienza a aumentar. Luego, este barrido de parámetros se puede refinar y los modelos en el frente de Pareto se evalúan utilizando criterios de información [44].

3.2 Identificación de sistemas dinámicos dispersos (SDSI)

El trabajo realizado en esta sección se basa en los resultados presentados por Vides en [65]. A continuación se muestran algunos métodos de aproximación de bajo rango con aplicaciones a la solución de problemas de mínimos cuadrados lineales dispersos los cuales serán de gran ayuda para el desarrollo de este trabajo.

3.2.1 Aproximación de bajo rango y solucionadores de mínimos cuadrados lineales dispersos

Definición 3.1. Dado $\delta > 0$ y una matriz $A \in \mathbb{C}^{m \times n}$, escribiremos $\text{rk}_\delta(A)$ para denotar el entero no negativo determinado por la expresión

$$\text{rk}_\delta(A) = \sum_{i=1}^{\min\{m,n\}} W_\delta(\sigma_i(A))$$

donde W_δ se definió en la sección 2.1 y los números $\sigma_j(A)$ representan los valores singulares correspondientes a una descomposición en valores singulares de tamaño económico de la matriz A (es decir, un tamaño menor o igual a $m \times n$).

Lema 3.2. a) Se tiene que $\text{rk}_\delta(A^T) = \text{rk}_\delta(A)$ para cada $\delta > 0$ y cada $A \in \mathbb{C}^{m \times n}$.

b) Dado $\delta > 0$ y $A \in \mathbb{C}^{m \times n}$ tendremos que $\text{rk}_\delta(A) \leq \text{rk}(A)$.

Teorema 3.3. Dado $\delta > 0$ y $x_1, x_2, \dots, x_m, y \in \mathbb{C}^n$, sea

$$X = \begin{bmatrix} | & & | \\ x_1 & \cdots & x_m \\ | & & | \end{bmatrix}.$$

Si $\text{rk}_\delta(X) > 0$ y si establecemos $r = \text{rk}_\delta(X)$ y $\sigma_{n,m}(r) = \sqrt{r(\min\{m,n\} - r)}$ entonces, existe una proyección ortogonal R de rango r , con r vectores $x_{j_1}, \dots, x_{j_r} \in \{x_1, \dots, x_m\}$ y r escalares $c_1, \dots, c_r \in \mathbb{C}$ tal que se tiene:

$$\|X - RX\|_F \leq \left(\frac{\sigma_{n,m}(r)}{\sqrt{r}}\right)\delta,$$

y

$$\left\|y - \sum_{k=1}^r c_k x_{j_k}\right\| \leq \left(\sum_{k=1}^r |c_k|^2\right)^{\frac{1}{2}} \sigma_{n,m}(r)\delta + \|(I_n - R)y\|.$$

Como implicación del teorema anterior se puede obtener el siguiente corolario.

Corolario 3.4. Dado $\delta > 0, y \in \mathbb{C}^m$ y $A \in \mathbb{C}^{m \times n}$. Si $\text{rk}_\delta(X) > 0$ y si hacemos $r = \text{rk}_\delta(A)$ y $\sigma_{n,m}(r) = \sqrt{r(\min\{m,n\} - r)}$ entonces, existe $x \in \mathbb{C}^n$ y una proyección ortogonal R de rango r que no depende de y , tal que $\|Ax - y\| \leq \|x\|\sigma_{n,m}(r)\delta + \|(I_m - R)y\|$ y x tiene como máximo r entradas distintas de cero.

Dado $\delta > 0$, y dos matrices $A \in \mathbb{C}^{m \times n}$ y $Y \in \mathbb{C}^{m \times p}$, escribiremos $AX \approx_\delta Y$ para representar el problema de encontrar $X \in \mathbb{C}^{n \times p}, \alpha, \beta \geq 0$ y un proyector ortogonal R tal que $\|AX - Y\|_F \leq \alpha\delta + \beta\|(I_m - R)Y\|_F$. A la matriz X la llamaremos solución del problema $AX \approx_\delta Y$.

Teorema 3.5. Dado $\delta > 0$, y dos matrices $A \in \mathbb{C}^{m \times n}$ y $Y \in \mathbb{C}^{m \times p}$. Si $\text{rk}_\delta(X) > 0$ y si establecemos $r = \text{rk}_\delta(A)$, entonces, hay una solución X al problema $AX \approx_\delta Y$ con como máximo rp entradas distintas de cero.

Para más detalles sobre las demostraciones y otras nociones del lema, corolario y teoremas vistos en esta sección, se remite al lector a consultar [65].

Aunque los algoritmos de solución de mínimos cuadrados lineales dispersos y los resultados teóricos de este trabajo se basan en principios similares a los considerados por Bousidis y Magdon-Ismael en [5] y por Brunton, Proctor y Kutz en [11], una de las principales diferencias del enfoque considerado en este trabajo con el enfoque utilizado en [5], es que dada una ecuación matricial lineal de mínimos cuadrados $X = \arg \min_Y \|AY - B\|_F$, aunque la aproximación $A_\delta = RA$ de rango $\text{rk}_\delta(A)$ correspondiente a la matriz de coeficientes A , se calcula de forma genérica utilizando la proyección ortogonal R determinada por el teorema 3.3, que a su vez puede calcularse utilizando una SVD truncada de tamaño económico para A , con error de aproximación $\varepsilon = O(\delta) > 0$, el proceso de selección de cada conjunto ordenado de las columnas de A_δ correspondiente a cada columna de la representación aproximada dispersa \bar{X} de la solución de mínimos cuadrados de referencia X , no es aleatorio sino B -dependiente, y la principal diferencia entre el enfoque implementado en este trabajo y el implementado en [11] es que el proceso de aproximación dispersa utilizado aquí se basa en soluciones de referencia de problemas de mínimos cuadrados que involucran submatrices de la aproximación de bajo rango A_δ de la matriz de coeficientes A , en lugar de submatrices de la matriz original A .

Más específicamente, si A tiene n columnas, una vez calculada la aproximación de bajo rango A_δ de A según el teorema 3.3 y el teorema 3.5, para cada columna x_j de una solución inicial de mínimos cuadrados de referencia X del problema $X = \arg \min_Y \|A_\delta Y - B\|_F$, se puede establecer un umbral $\varepsilon > 0$, encontrar un número entero $N_j(\varepsilon) \leq \text{rk}_\delta(A)$ y calcular una permutación $\rho_j : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ basado en los módulos $|x_{i,j}|$ de las entradas de x_j de acuerdo con las siguientes condiciones

$$\begin{aligned} |x_{\rho_j(1),j}| &\geq |x_{\rho_j(2),j}| \geq \dots \geq |x_{\rho_j(N_j(\varepsilon)),j}| > \varepsilon \\ \varepsilon &\geq |x_{\rho_j(N_j(\varepsilon)+1),j}| \geq \dots \geq |x_{\rho_j(n-1),j}| \geq |x_{\rho_j(n),j}|. \end{aligned}$$

Para cada subconjunto ordenado $\tilde{a}_{\rho_j(1)}, \dots, \tilde{a}_{\rho_j(N_j(\varepsilon))}$ de columnas de A_δ , podemos resolver los problemas

$$w_j = \operatorname{argmin}_y \left\| \begin{bmatrix} | & & | \\ \tilde{a}_{\rho_j(1)} & \cdots & \tilde{a}_{\rho_j(N_j(\varepsilon))} \\ | & & | \end{bmatrix} y - b_j \right\|$$

para cada $j = 1, \dots, p$.

Si definimos p vectores $\tilde{x}_1, \dots, \tilde{x}_p$ según las siguientes asignaciones

$$\tilde{x}_{\sigma_j(k),j} = \begin{cases} w_{k,j}, & 1 \leq k \leq N_j(\varepsilon) \\ 0, & N_j(\varepsilon) < k \leq n \end{cases}$$

donde $\tilde{x}_{i,j}$ y $w_{k,j}$ denotan las entradas de cada par de vectores \tilde{x}_j y w_j , respectivamente,

obtenemos una nueva solución dispersa aproximada

$$\tilde{X} = \begin{bmatrix} | & & | \\ \tilde{x}_1 & \cdots & \tilde{x}_p \\ | & & | \end{bmatrix}$$

al problema $\bar{X} = \arg \min_Y \|A_\delta Y - B\|_F$. Usando \tilde{X} como una nueva solución de referencia, se puede repetir este proceso un número preestablecido de veces, o hasta que se cumpla algún criterio de parada (ver algoritmo 1 en [65]), para encontrar una representación más dispersa de la solución de referencia inicial X .

Los resultados presentados en esta sección se pueden observar en la resolución de mínimos cuadrados lineales dispersos descrito en la sección 3.2.3.

3.2.2 Métodos de aproximación de rango bajo para la identificación de modelos dispersos

El trabajo realizado en esta sección se basa en los resultados presentados por Vides en [65].

Identificación dispersa de operadores de transición para series temporales con simetrías. Dada una sucesión $\{x_t\}_{t \geq 1} \subset \mathbb{C}^n$, decimos que $\{x_t\}_{t \geq 1}$ es una serie de tiempo de un sistema (Γ, Υ) , si $\{x_t\}_{t \geq 1} \subset \Gamma$ y $\Upsilon(x_t) = x_{t+1}$ para cada $t \geq 1$. Si además existe un grupo finito $G_N = \{g_1, \dots, g_N\} \subset \mathbb{U}(n) = \{X \in \mathbb{C}^{n \times n} : X^*X = XX^* = I_n\}$ tal que

$$\Upsilon(g_j x_t) = g_j \Upsilon(x_t),$$

para cada $t \geq 1$ y cada $g_j \in G_N$, diremos que el sistema (Γ, Υ) es G_N equivalente y que la sucesión $\{x_t\}_{t \geq 1}$ es una serie de tiempo con simetrías.

Diremos que una matriz $S \in \mathbb{C}^{n \times n}$ es simétrica respecto a un grupo finito $G_N = \{g_1, \dots, g_N\} \subset \mathbb{U}(n)$ si

$$g_j S = S g_j$$

para cada $g_j \in G_N$. Se escribirá $\mathbb{S}(n)^{G_N}$ para denotar el conjunto de todas las matrices en $\mathbb{C}^{n \times n}$ que son simétricas con respecto al grupo G_N .

Dado un entero $L \geq 1$, un grupo finito $G_N \subset \mathbb{U}(n)$ con $\#(G_N) = N$, y una muestra $\Gamma_T = \{x_t\}_{t=1}^T$ de una serie de tiempo $\{x_t\}_{t \geq 1}$ en el espacio de estado $\Gamma \subset \mathbb{C}^n$ de algún sistema G_N equivalente (Γ, Υ) , escribiremos $\mathbb{H}_L(\Gamma_T, G_N)$ para denotar la matriz de bloques estructurada con bloques de matriz tipo Hankel que está determinada por la siguiente expresión

$$\mathbb{H}_L(\Gamma_T, G_N) = [(I_L \otimes g_1)\mathbb{H}_L(\Gamma_T) \quad \cdots \quad (I_L \otimes g_N)\mathbb{H}_L(\Gamma_T)].$$

OBS. Ya que $G_N \subset \mathbb{U}(n)$ es un grupo, uno de los elementos de G_N es igual a I_n , por lo tanto, uno de los bloques de la matriz de $\mathbb{H}_L(\Gamma_T, G_N)$ es igual a $I_L \otimes I_n \mathbb{H}_L(\Gamma_T) = \mathbb{H}_L(\Gamma_T)$.

A continuación se define el principal problema de identificación del modelo disperso para series de tiempo con simetrías donde se aplica la idea planteada por Vides en [65].

Problema de identificación de modelos dispersos para series de tiempo con simetrías

Dado $\delta > 0$, un entero $L > 0$, un grupo finito $G_N \subset \mathbb{U}(n)$ con $\#(G_N) = N$, y una muestra $\Gamma_T = \{x_t\}_{t=1}^T$ de una serie de tiempo $\{x_t\}_{t \geq 1} \subset \mathbb{C}^n$ de un sistema G_N equivalente (Γ, Υ) con operador de transición Υ a ser identificado. Sea $\Gamma_0 = \{x_1, \dots, x_{T-1}\}$, $\Gamma_1 = \{x_2, \dots, x_T\}$, $\mathbf{H}_{L,k} = \mathbb{H}_L(\Gamma_k, G_N)$ para $k = 0, 1$ y $G_N = \{I_L \otimes g_j : g_j \in G_N\}$. Determine si es posible calcular una matriz dispersa \bar{A}_T , una matriz $A_T \in \mathbb{S}(nL)^{\tilde{G}_N}$, una proyección ortogonal R y tres números no negativos d, e y f tales que si establecemos

$$\begin{aligned} v &= d\delta + \sqrt{nL} \|\mathbf{H}_{L,1}(I_{N(T-L)} - R)\|_F, \\ \varepsilon &= e\delta + f \|\mathbf{H}_{L,1}(I_{N(T-L)} - R)\|_F, \end{aligned}$$

entonces

$$\begin{aligned} \|\mathbf{H}_{L,1} - \bar{A}_T \mathbf{H}_{L,0}\|_F &\leq v, \\ \|\Upsilon(x_t) - P_L \bar{A}_T^t X_1\| &\leq \varepsilon, \\ \|\Upsilon(g_i x_t) - P_L \bar{A}_T^t (I_L \otimes g_i) X_1\| &\leq \varepsilon, \\ \|\mathbf{H}_{L,1} - A_T \mathbf{H}_{L,0}\|_F &\leq v, \\ \|\Upsilon(x_t) - P_L A_T^t X_1\| &\leq \varepsilon, \\ \|\Upsilon(g_i x_t) - P_L A_T^t (I_L \otimes g_i) X_1\| &\leq \varepsilon, \end{aligned}$$

para cada $t = 1, \dots, T-L$ y cada $g_j \in G_N$, con $P_L = \bar{e}_{1,L}^T \otimes I_n$ y $X_1 = [x_1 \ \dots \ x_L]^T$.

Definición 3.6. Escribimos $\text{SDSI}(\Gamma_T, G_N, L, \delta)$ para denotar el conjunto de 6-tuplas de soluciones $(\bar{A}_T, A_T, R, d, e, f)$ al problema 3.2.2 basado en los datos Γ_T, G_N, L, δ .

Teorema 3.7. Dado $\delta > 0$, un grupo finito $G_N \subset \mathbb{U}(n)$ con $\#(G_N) = N$, y una muestra $\Gamma_T = \{x_t\}_{t=1}^T$ de una serie de tiempo $\{x_t\}_{t \geq 1} \subset \mathbb{C}^n$ de un sistema G_N equivalente (Γ, Υ) con el operador de transición Υ a identificar. Si existe un entero $L > 0$ tal que $\text{rk}_\delta(H_L(\{x_t\}_{t=1}^T, G_N)) > 0$ entonces existe $(\tilde{A}_T, A_T, R, d, e, f) \in \text{SDSI}(\Gamma_T, G_N, L, \delta)$.

Para ver los detalles de la demostración del teorema anterior, se remite al lector a consultar [65].

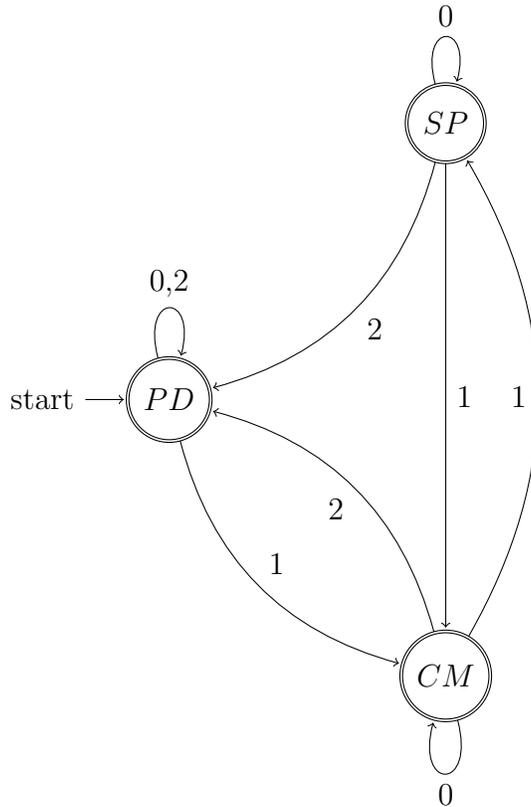
Podemos aplicar el teorema 3.7 para la identificación de modelos dispersos, especialmente cuando se puede estimar un modelo predictivo disperso $\bar{\Upsilon}$ para una serie de tiempo $\{x_t\}_{t \geq 1}$ de un sistema (Γ, Υ) para un horizonte de tiempo dado $T > 0$, basado en una muestra $\Gamma_S = \{x_t\}_{t=1}^S \subset \{x_t\}_{t \geq 1}$ para algunos $S \ll T$. Ejemplos importantes de sistemas que satisfacen la consideración anterior son los sistemas periódicos y eventualmente periódicos, como los considerados en [20] y [23], respectivamente.

Los resultados e ideas presentados en esta sección pueden traducirse en algoritmos de identificación dispersos como el algoritmo 2.

3.2.3 Algoritmos

Los algoritmos y trabajos realizado en esta sección se basan en los resultados presentados por Vides en [64, 65]. Uno de los propósitos de esta tesis es proporcionar herramientas de

identificación de sistemas dinámicos dispersos (SDSI) enfocadas en enfermedades infecciosas se busca que se puedan utilizar para construir marcos colaborativos de métodos teóricos y computacionales que se pueden aplicar en un contexto multidisciplinario, donde se requiere la identificación de sistemas. Un ejemplo de los marcos colaborativos antes mencionados puede ser descrito por el autómata ilustrado a continuación en el cual se describe autómatas de estado finito de procesos SDSI genéricos.



Los bloques PD, CM y SP del sistema descritos anteriormente corresponden a las etapas de procesamiento de datos, cómputo del modelo y simulación predictiva de un proceso de identificación de un sistema disperso genérico, respectivamente, mientras que las etiquetas 0, 1 y 2 corresponden a los estados cómputo en curso, cómputo completado y se requieren más datos, respectivamente.

En esta tesis nos centramos en los algoritmos de resolución de mínimos cuadrados lineales dispersos y en las invariantes topológicas aproximadas en forma de números fácilmente computables, que se pueden utilizar en el bloque de modelado CM del autómata presentado para la identificación del modelo disperso. Entre otros casos, las señales de control para autómatas como en el caso anterior pueden ser proporcionadas por un experto interesado en la identificación dinámica de algún sistema en particular. Aunque los programas en [64] pueden usarse, adaptarse o modificarse para trabajar en cualquiera de los dos casos considerados anteriormente, los programas y experimentos incluidos como parte del trabajo de esta tesis están desarrollados teniendo en cuenta el primer caso. Los esquemas de inteligencia artificial se pretenden explorar más a fondo en el futuro.

Aunque los resultados de esta tesis se centran en la identificación de modelos dispersos en un enfoque de enfermedades infecciosas, además de los programas correspondientes a solucionadores de mínimos cuadrados lineales dispersos e identificadores de grado y rango aproximados basados en los resultados de las secciones 3.2.1 y 3.2.2, algunos programas para lectura y escritura de datos, la generación sintética de señales y la simulación predictiva también se incluyen como parte del conjunto de herramientas SDSI disponible en [64].

Algoritmo solucionador de mínimos cuadrados lineales dispersos

Una aplicación de los resultados e ideas presentados en la sección 3.2.1 es el algoritmo 1 en [65], el cual es solucionador de mínimos cuadrados lineales dispersos. Los problemas de mínimos cuadrados $c = \operatorname{argmin}_{\bar{c} \in \mathbb{C}^K} \|\bar{A}\bar{c} - y\|$ que se resolverán como parte del proceso correspondiente al algoritmo 1 en [65] se puede resolver con cualquier solucionador de mínimos cuadrados eficiente disponible en el lenguaje o programa donde se implementa el algoritmo de solución de mínimos cuadrados lineales dispersos.

Algoritmo de identificación de modelo de serie de tiempo disperso

Dada una serie de tiempo $\{x_t\}_{t \leq 1}$ de un sistema (Γ, Υ) con el operador de transición Υ a identificar, podemos abordar el cálculo de las aproximaciones locales de Υ basadas en una muestra de datos $\Gamma_T = \{x_t\}_{t=1}^T \subset \Gamma$ utilizando el algoritmo 2.

Para este trabajo, cuando los datos de la serie de tiempo $\Gamma_T = \{x_t\}_{t=1}^T \subset \mathbb{C}^n$ de un sistema dado (Γ, Υ) con el operador de transición Υ a identificar, no se muestrean uniformemente en el tiempo, la muestra Γ_T se preprocesa aplicando métodos de interpolación spline local para obtener una estimación uniforme en el tiempo $\tilde{\Gamma}_T = \{\tilde{x}_t\}_{t=1}^T \subset \mathbb{C}^n$ para Γ_T . El programa de Matlab `DataSpliner.m` es un ejemplo de una implementación computacional de este procedimiento de interpolación y se incluye como parte de los programas en el conjunto de herramientas SDSI disponible en [64].

Dado un grupo finito $G_N \subset \mathbb{U}(n)$, un sistema G_N equivalente (Γ, Υ) y una muestra de datos $\Gamma_T \subset \Gamma$, si los elementos P, \bar{A}, A, X_1 se calculan usando el algoritmo 2 con el ajuste $(P, \bar{A}, A, X_1) = \text{SDSI}(G_N, \bar{\Gamma}_T, L, \delta, \varepsilon)$ para algunos $\delta, \varepsilon > 0$ adecuados, se pueden construir dos modelos predictivos $\mathbf{P}_{\bar{A}}, \mathbf{P}_A$ para la evolución temporal del sistema, determinados por la relación de recurrencia $x_{t+1} = \Upsilon(x_t), t \geq 1$, usando esquemas de la forma $\mathbf{P}_{\bar{A}}(t) = P\bar{A}^t X_1$ y $\mathbf{P}_A(t) = PA^t X_1$, para $t \geq 1$.

El algoritmo presentado a continuación se extraen de [65], en este trabajo se muestra la factibilidad y rapidez del mismo aplicado a sistemas epidemiólogos lineales y no lineales.

Algorithm 2 SDSI: Algoritmo para la identificación de sistemas dinámicos dispersos [65].

Entrada: $G_N \subset \mathbb{U}(n)$, $\tilde{\Gamma}_T = \{x_t\}_{t=1}^T \subset \mathbb{C}^n$, $L \in \mathbb{Z}^+$, $\delta > 0$, $\varepsilon > 0$

Salida: $(P_L, \tilde{A}_T, A_T, X_1) = \text{SDSI}(G_N, \Gamma_T, L, \delta, \varepsilon)$

- 1: Calcular $L = \max\{\text{drk}_{\delta, G_N}(\tilde{\Gamma}_T), L\}$
 - 2: $\tilde{\Gamma}_0 \leftarrow \{x_t\}_{t=1}^{T-1}$, $\tilde{\Gamma}_1 = \{x_t\}_{t=2}^T$
 - 3: $\mathbf{H}_{L,k} \leftarrow \mathbb{H}_L(\tilde{\Gamma}_k, G_N)$, $k = 0, 1$
 - 4: Resolver $\mathbf{H}_{L,0}^T C \approx_{\delta} \mathbf{H}_{L,1}^T$ aplicando el algoritmo 1 en [65] con el ajuste $C = \text{SLRSolver}(\mathbf{H}_{L,0}^T, \mathbf{H}_{L,1}^T, \delta, nL, \varepsilon)$
 - 5: $\tilde{A}_T \leftarrow C^T$
 - 6: $A_T \leftarrow \frac{1}{N} \sum_{i=1}^N (I_L \otimes g_i^*) \tilde{A}_T (I_L \otimes g_i)$
 - 7: $X_1 \leftarrow [x_1^T \cdots x_L^T]^T$
 - 8: $P_L \leftarrow \tilde{e}_{1,L}^T \otimes In$
 - 9: **devolver** $(P_L, \tilde{A}_T, A_T, X_1)$
-

Ejemplo Computacional

Un ejemplo de lo mencionado anteriormente extraído de los resultados presentados por Vides en [65], en el cual consideremos una muestra $\Gamma_{257} = \{s_k : 1 \leq k \leq 257\}$ de una señal escalar en tiempo discreto $S = \{s_k : k \in \mathbb{Z}^+\}$ con grupo de simetría trivial $G_1 = \{1\}$, que está determinada para cada $k \in \mathbb{Z}^+$ por la expresión

$$s_k = \sum_{j=0}^{\infty} \min\{t_k - j, 1 - t_k + j\} (W_j(t_k) - W_{j+1}(t_k)),$$

con $t_k = \frac{k-1}{32}$. Agregando ruido a la muestra Γ_{257} usando una sucesión de números pseudoaleatorios normalmente distribuidos $\mathbf{RN}_{257} = \{r_k : |r_k| = O(1 \times 10^{-3}), 1 \leq k \leq 257\}$, obteniendo una versión ruidosa $\tilde{\Gamma}_{257} = \{s_k + r_k : 1 \leq k \leq 257\}$ de la muestra original Γ_{257} .

Si se considera una submuestra $\tilde{\Gamma}_{70} = \{s_k + r_k : 1 \leq k \leq 70\} \subset \tilde{\Gamma}_{257}$ Calculando $\text{drk}_{\delta, G_1}(\tilde{\Gamma}_{70})$ con $\delta = 1 \times 10^{-2}$ con el programa de Matlab `drk.m` en [64] obtenemos $\text{drk}_{\delta, G_1}(\tilde{\Gamma}_{70}) = 17$. Ahora podemos calcular un modelo predictivo

$$s_{k+1} = c_1 s_k + c_2 s_{k-1} + \cdots + c_L s_{k-L+1} \approx P_L A_L^k \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_L \end{bmatrix},$$

para $k \geq L = \text{drk}_{\delta, G_1}(\tilde{\Gamma}_{70}) = 17$, así que se obtiene el siguiente modelo

$$s_{k+1} = 1.00003s_k - 0.99815s_{k-15} + 0.99811s_{k-16}, \quad \text{para } k \geq 17.$$

Las señales identificadas para diferentes valores del parámetro de retardo L se muestran en la figura 3.1

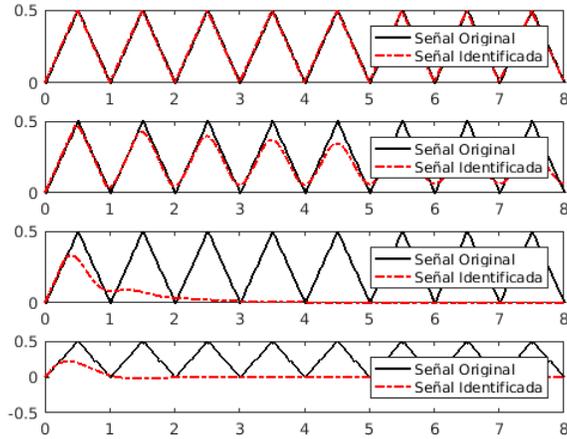


Figure 3.1: Señales identificadas para diferentes valores de retardo. $L = 17$ (primera fila superior). $L = 16$ (segunda fila). $L = 10$ (tercera fila). $L = 5$ (última fila). Fuente: [64]

Los coeficientes c_k correspondientes a los modelos identificados para diferentes valores del parámetro de retardo L se muestran en la figura 3.2. El error cuadrático medio (RMSE) correspondiente a diferentes valores del parámetro de retardo L se documentan en la tabla 3.1

Valor de retraso L	RMSE
5	0.2797
10	0.2659
16	0.0674
17	0.0028

Table 3.1: Errores cuadráticos medios correspondientes a cada valor de retardo. Fuente: [64]

La configuración computacional utilizada para este ejemplo está documentada en el programa `Example1.m` en [64]. Para cada valor de retardo, los parámetros del modelo correspondiente se calcularon utilizando el programa `SpSolver.m` en [64] basado en el algoritmo 1 en [65], con la misma tolerancia $\delta > 0$, para exponer la posible obstrucción de aproximación topológicamente controlada identificada por el número $\text{drk}_{\delta, G_1}(\tilde{\Gamma}_{70})$.

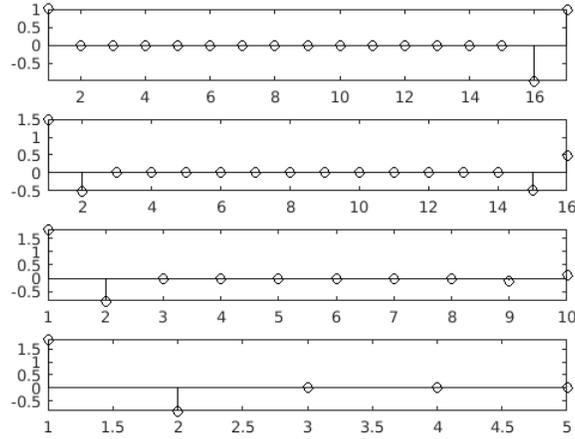


Figure 3.2: Coeficientes identificados para diferentes valores de retardo. $L = 17$ (primera fila superior). $L = 16$ (segunda fila). $L = 10$ (tercera fila). $L = 5$ (última fila). Fuente: [64]

Como consecuencia del teorema 3.7 y de las observaciones anteriores, se puede ver que dado un grupo finito $G_N \subset \mathbb{U}(n)$ y una muestra $\Gamma_T = \{x_t\}_{t=1}^T$ de una serie de tiempo $\{x_t\}_{t \geq 1} \subset \mathbb{C}^n$ de un sistema G_N equivalente (Γ, Υ) , si el número $L = \text{drk}_{\delta, G_N}(\Gamma_T)$ es positivo, podemos usar este número para estimar una condición necesaria para la computabilidad de una solución dispersa al problema

$$\mathbf{H}_{L,0}^T A_T^T \approx_{\delta} \mathbf{H}_{L,1}^T,$$

donde $\mathbf{H}_{L,k} = H_L(\Gamma_k, G_N)$ para $k = 0, 1$, con $\Gamma_0 = \{x_t\}_{t=1}^{T-1}$ y $\Gamma_1 = \{x_t\}_{t=2}^T$. En particular, cuando $\bar{L} = \text{drk}_{\delta, G_N}(\Gamma_T)$ es positivo, el valor de retardo $L = \bar{L}$ proporcionaría un buen punto de partida para un método de identificación de sistema disperso adaptativo, si el error de predicción aún no es lo suficientemente pequeño para el valor de retardo $L = \bar{L}$, un algoritmo de control puede seguir aumentando el valor del parámetro de retardo hasta que se alcanza el error de predicción o se alcanza algún límite prescrito para el valor de retardo. Para los experimentos documentados en este trabajo, hemos utilizado técnicas de autocorrelación de señales escalares estándar para estimar los límites admisibles para los valores de retardo, en el conjunto de herramientas identificación de sistemas dinámicos dispersos (SDSI) disponible en [64]. Estas ideas se implementan en el programa de Matlab `LagEstimate.m` basado en la función de Matlab `xcorr.m`.

3.3 Modelo basado en datos para CPM

En esta sección describimos arquitecturas de control basadas en datos que combinan el descubrimiento de modelos basados en datos con estrategias avanzadas de control basadas en modelos. En este trabajo, usamos la arquitectura SINDy-CPM para identificar rápidamente un modelo de orden bajo que se usa con control predictivo de modelos. Consideramos un sistema dinámico no lineal de la forma:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad (3.5)$$

donde $\mathbf{x} \in \mathbb{R}^n$, entrada de control $\mathbf{u} \in \mathbb{R}^q$ y dinámica $\mathbf{f}(\mathbf{x}, \mathbf{u}) : \mathbb{R}^n \times \mathbb{R}^q \rightarrow \mathbb{R}^n$. En esta sección, describimos SINDy con control y CPM. Este enfoque se utilizará en el siguiente capítulo para proporcionar un experimento para calcular SINDy-CPM en un problema de control de enfermedades infecciosas.

Identificación dispersa de dinámica no lineal con control

Los algoritmos SINDy [11] y SINDy con control identifican un sistema dinámico no lineal disperso a partir de datos, basándose en la suposición de que muchos sistemas tienen relativamente pocos términos activos en la dinámica. SINDy con control utiliza regresión sparse para identificar estos pocos términos activos de una biblioteca $\Theta(\mathbf{x}, \mathbf{u})$ de términos candidatos del modelo lineal y no lineal en el estado \mathbf{x} y la señal de entrada \mathbf{u} , que se requieren para aproximar la función \mathbf{f} en la ecuación 3.5. SINDy junto con el algoritmo de control se muestran en el experimento en un modelo de una enfermedad infecciosa. Para evaluar Θ , primero se tiene m mediciones del estado \mathbf{x} y la señal de entrada \mathbf{u} en el tiempo y las organizamos en dos matrices:

$$X = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_m]^T \text{ y } U = [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_m]^T.$$

Después de recopilar las mediciones, podemos evaluar la biblioteca de funciones no lineales candidatas:

$$\Theta(X, U) = [1 \quad X \quad U \quad (X \otimes X) \quad (X \otimes U) \quad (U \otimes U) \quad \cdots]^T,$$

donde $X \otimes U$ define el vector de todas las combinaciones de productos de los componentes en \mathbf{x} y \mathbf{u} . Esta biblioteca puede incluir cualquier función que pueda describir los datos. La estrategia recomendada es comenzar con una opción básica, como polinomios de bajo orden, y luego aumentar la complejidad y el orden de la biblioteca hasta obtener modelos dispersos y precisos.

Además de evaluar la biblioteca, debemos calcular las derivadas temporales del estado $\dot{X} = [\dot{\mathbf{x}}_1 \quad \dot{\mathbf{x}}_2 \quad \cdots \quad \dot{\mathbf{x}}_m]^T$, normalmente mediante diferenciación numérica. El sistema en la ecuación 3.5 puede entonces escribirse en términos de estas matrices de datos como:

$$\dot{X} = \Theta(X, U)\Xi.$$

Muchos sistemas dinámicos tienen relativamente pocos términos activos en las ecuaciones gobernantes. Por lo tanto, los coeficientes Ξ son en su mayoría dispersos y podemos emplear una regresión sparse para identificar la matriz dispersa de coeficientes Ξ que significa la menor cantidad de no linealidades en nuestra biblioteca que da como resultado un buen ajuste del modelo:

$$\xi_k = \operatorname{argmin}_{\bar{\xi}_k} \frac{1}{2} \|\dot{X}_k - \Theta(X, U)\bar{\xi}_k\|_2^2 + \lambda \|\bar{\xi}_k\|_0.$$

\dot{X}_k es la k -ésima fila de \dot{X} , ξ_k es la k -ésima fila de Ξ y λ es el parámetro que promueve la dispersión. El término $\|\cdot\|_0$ promueve la dispersión en el vector de coeficientes ξ_k . Para resolver esta optimización, utilizamos el algoritmo 1.

Control predictivo de modelos

Este trabajo se explora el uso de modelos SINDy o SDSI para el control predictivo de modelos. CPM es un control efectivo basado en modelos, que ha revolucionado el panorama del control industrial [13]. CPM permite el control de sistemas fuertemente no lineales con restricciones, retrasos de tiempo, dinámica de fase no mínima e inestabilidad. Estos sistemas son difíciles de controlar utilizando enfoques lineales tradicionales [45].

En CPM, calculamos una secuencia de control $\mathbf{u}(\mathbf{x}_j) = \{\mathbf{u}_{j+1}, \dots, \mathbf{u}_{j+m_c}\}$, dada la estimación o medición del estado actual \mathbf{x}_j , mediante una optimización restringida sobre un horizonte en retroceso $T_c = m_c \Delta t$, siendo Δt el paso de tiempo del modelo y m_c el número de pasos de tiempo. En cada paso de tiempo, repetimos la optimización, actualizamos la secuencia de control sobre el horizonte de control y aplicamos la primera acción de control \mathbf{u}_{j+1} al sistema. La secuencia de control óptima $\mathbf{u}(\mathbf{x}_j)$ se obtiene minimizando una función de costo J sobre un horizonte de predicción $T_p = m_p \Delta t \geq T_c$. La función de costo es:

$$J = \sum_{k=0}^{m_p-1} \|\bar{\mathbf{x}}_{j+k} - \mathbf{r}_k\|_{\mathbf{Q}}^2 + \sum_{k=1}^{m_c-1} (\|\mathbf{u}_{j+k}\|_{\mathbf{R}_u}^2 + \|\Delta \mathbf{u}_{j+k}\|_{\mathbf{R}_{\Delta u}}^2),$$

sujeto a la dinámica y restricciones de tiempo discreto. La función de costo J penaliza las desviaciones del estado predicho $\bar{\mathbf{x}}$ de la trayectoria de referencia \mathbf{r} , el gasto de control \mathbf{u} y la tasa de cambio de la señal de control $\Delta \mathbf{u}$. Cada término está ponderado por las matrices \mathbf{Q} , \mathbf{R}_u y $\mathbf{R}_{\Delta u}$, respectivamente. Para permitir que este ciclo de optimización se ejecute en tiempo real, CPM se basa en modelos eficientes y computación de alto rendimiento.

La ventaja de CPM radica en el ajuste simple e intuitivo y la capacidad de controlar fenómenos complejos, especialmente con restricciones conocidas y múltiples condiciones de operación. También funciona para sistemas con retrasos de tiempo y brinda la flexibilidad para formular y adaptar un objetivo de control. El mayor desafío de CPM radica en el desarrollo de un modelo adecuado a través de la identificación del sistema [33]. Los modelos no lineales basados en el aprendizaje automático se utilizan cada vez más. Sin embargo, estas técnicas a menudo se basan en el acceso a conjuntos de datos masivos, tienen una capacidad limitada para generalizar, no incorporan fácilmente restricciones físicas conocidas y requieren cálculos costosos y lentos. En cambio, Kaiser en [33] mostró que los modelos simples obtenidos a través de DMD con control [12] y SINDy con control [33] funcionan casi tan bien con CPM en un modelo no lineal completo, y pueden entrenarse en un período de tiempo sorprendentemente corto.

Capítulo 4

Experimentos y Resultados

4.1 Experimento 1

Con lo mencionado en la sección 2.4.2 se procede a dar un experimento computacional para ilustrar el algoritmo DMD, en el cual se realiza una comparación de DMD con el análisis de componentes principales (PCA).

Mediante la técnica descrita en la sección 2.4.2, se considera el experimento de dos señales espacio-temporales mixtas. En particular, el objetivo es demostrar la capacidad de DMD para descomponer eficientemente la señal. Las dos señales de interés son:

$$\begin{aligned} f(x, t) &= f_1(x, t) + f_2(x, t) \\ &= \operatorname{sech}(x - \pi)e^{i2.3t} + 2\operatorname{sech}(x)\tanh(x)e^{i2.8t}. \end{aligned}$$

Las señales espacio-temporales individuales $f_1(x, t)$ y $f_2(x, t)$ se ilustran en la Figura 4.1 (a) - (b). Las dos frecuencias presentes son $\omega_1 = 2.3$ y $\omega_2 = 2.8$, que tienen estructuras espaciales distintas. La señal mixta $f(x, t) = f_1(x, t) + f_2(x, t)$ se ilustra en la Figura 4.1 (c). Para lo antes mencionado se muestra algunas partes del algoritmo de DMD que genera una aproximación numérica de la señal mixta, esto mediante GNU Octave con algunas técnicas computacionales empleadas en [63], a continuación se muestra algunas partes del código desarrollado en GNU Octave para este experimento numérico.

```
>> n=400; m=200;
>> [Phi, omega, lambda, b, Xdmd] = DMDE1(n, m)

% ALGUNAS PARTES DEL ALGORITMO DMD EN GNU Octave
[U, S, V]=svd(X1, 'econ'); % Paso 1
Atilde=Ur'*X2*Vr/Sr; % Paso 2
[W, D]=eig(Atilde); % Paso 3
Phi=X2*Vr/Sr*W; % Paso 4
Xdmd=Phi*dintiem;
```

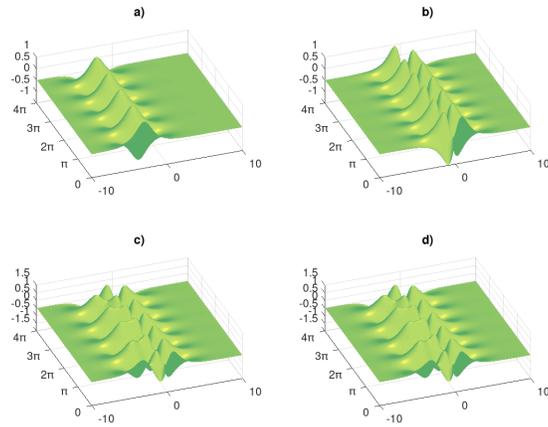


Figure 4.1: Experimento 1. Dinámica espacio-temporal de dos señales (a) $f_1(x, t)$, (b) $f_2(x, t)$ y la mezcla en (c) $f(x, t) = f_1(x, t) + f_2(x, t)$. La función $f(x, t)$ se puede representar en su lugar por $x(t)$. Se calculó la DMD de x , y en la imagen (d) se muestra una reconstrucción aproximada de rango 2 de la señal $x_{DMD}(t)$. La reconstrucción es casi perfecta, con los modos y espectros DMD que coinciden estrechamente con las señales $f_1(x, t)$ y $f_2(x, t)$. Fuente: Elaboración propia.

En la figura 4.2 los valores singulares en (a) y (b) muestran que un truncamiento de rango 2 es apropiado. Las imágenes (d) - (f) comparan los aspectos temporales y espaciales de los dos modos, donde los modos verdaderos se grafican junto con los modos extraídos por DMD y PCA de los datos. DMD produce resultados que están exactamente (con precisión numérica) alineados con la verdadera solución. Los modos PCA se desvían de la verdadera solución. La diferencia de la norma-2 entre los dos modos espaciales verdaderos y los modos extraídos por DMD y PCA se muestran en (c). Los modos DMD coinciden con los modos verdaderos.

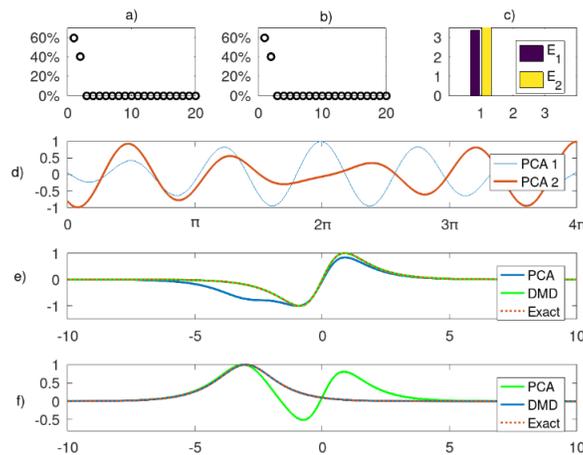


Figure 4.2: En las imágenes se muestra una comparación de DMD con PCA aplicada a los datos del Experimento 1. Fuente: Elaboración propia.

4.2 Experimento 2

Para el siguiente experimento numérico se ilustra el algoritmo de Koopman en relación con el algoritmo DMD, considerando la ecuación no lineal de Schrödinger (NLS)

$$i\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial^2 u}{\partial \xi^2} + |u|^2 u = 0, \quad (4.1)$$

donde $u(\xi, t)$ es una función espacio-tiempo, donde la ecuación se puede reescribir en la forma

$$\frac{\partial u}{\partial t} = \frac{i}{2}\frac{\partial^2 u}{\partial \xi^2} + i|u|^2 u,$$

para que se pueda aplicar una solución del método espectral. La transformación de Fourier en ξ , denotada por el símbolo del sombrero, da la ecuación diferencial en las variables de dominio de Fourier \tilde{u} :

$$\frac{d\tilde{u}}{dt} = -\frac{ik^2}{2}\tilde{u} + i\widehat{|u|^2 u}.$$

Al discretizar en la variable espacial, podemos desarrollar esta ecuación con un algoritmo estándar de pasos en el tiempo, se resuelve con Runge-Kutta de cuarto orden, suponemos que hay un total de 20 porciones de datos en el tiempo en el intervalo $t \in [0, \pi)$ donde las variables de estado son una discretización n -dimensional de $u(\xi, t)$ de modo que $u(\xi, t_k) \rightarrow \xi_k$, donde $n = 512$. Estas ξ_k son las columnas de la matriz de datos generada. En este ejemplo en específico, analizamos la solución de dos solitones que tiene la condición inicial $u(\xi, 0) = 2\text{sech}(\xi)$, el resultado de esta simulación se muestra en la Figura 4.3 (a). El algoritmo DMD descrito en las secciones anteriores nos proporciona una aproximación de bajo rango ($r = 10$). La Figura 4.3 (b) muestra la aproximación DMD lograda. Para lo antes mencionado se muestra algunas partes del algoritmo de DMD y algoritmo de Koopman que genera una aproximación numérica a la solución de 4.1, esto mediante GNU Octave con técnicas computacionales utilizadas en [63].

El algoritmo se ejecutó con los valores:

```
L=30; n=512; % Longitud del dominio y # puntos
% Discretización en el tiempo
slices=20;
t=linspace(0,pi,slices+1); dt=t(2)-t(1);
% Condiciones iniciales
N=2;
u=N*(sech(x)).';
ut=fft(u);
% Resuelve con Runge - Kutta
[t,utsol]=ode45('DMD_soliton',t,ut,[],k);

% ALGUNAS PARTE DEL ALGORITMO DMD EN GNU Octave

%%%%%%%%%%%%%% Koopman %%%%%%%%%%%%%%%
```

```

[U2,Sigma2,V2] = svd(Y1,'econ'); % Paso 1
Atilde = U'*Y2*V/Sigma; % Paso 2
[W,D] = eig(Atilde); % Paso 3
Phi2=Y2*V/Sigma*W; % Paso 4

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

u_dmd2 = Phi2*u_modes;

```

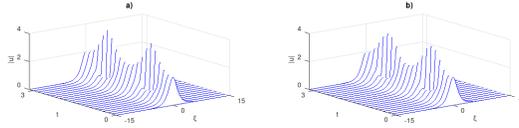


Figure 4.3: Experimento 2 de la ecuación no lineal de Schrödinger (NLS) 2.9 donde en la imagen (a) se muestra la simulación completa. La aproximación se muestra en la imagen (b) usando el algoritmo de DMD. Fuente: Elaboración propia.

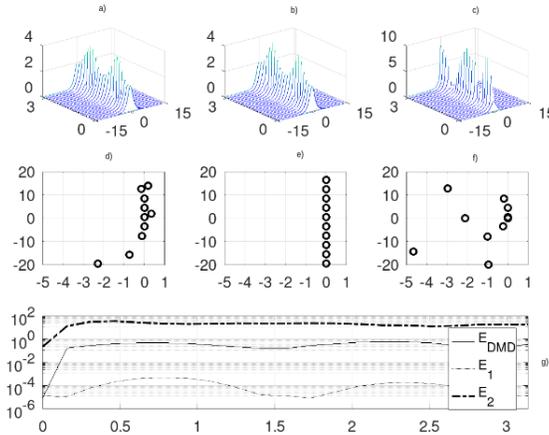


Figure 4.4: En la imagen se muestran las funciones propias de Koopman y distribución de valores propios para los tres observables considerados en los incisos (a)-(c) [fila superior], el algoritmo de DMD es dado por $\mathbf{g}_{DMD}(\mathbf{x}) = \mathbf{x} = u(\xi, t_k)$, se tiene los observables $\mathbf{g}_1(x) = [\mathbf{x}, |\mathbf{x}|^2 \mathbf{x}]^T$ y $\mathbf{g}_2(\mathbf{x}) = [\mathbf{x}, |\mathbf{x}|^2]^T$. Tener en cuenta que el observable $\mathbf{g}_1(\mathbf{x})$ [segunda imagen de la primera fila (b), segunda imagen de la segunda fila (e)] nos da la mejor aproximación al espectro de valores propios puramente imaginarios. En el análisis de errores de la reconstrucción de Koopman de los datos simulados, los tres observables considerados se comparan con la solución verdadera, el observable correspondiente a la teoría de DMD está dado por $\mathbf{g}_{DMD}(\mathbf{x})$, y produce el error E_{DMD} , los observables $\mathbf{g}_1(\mathbf{x})$ y $\mathbf{g}_2(x)$ producen errores E_1 y E_2 , respectivamente. Fuente: Elaboración propia.

4.3 Experimento 3

Primero generamos los datos usando un modelo epidemiológico SEIR [susceptible-expuesto-infectado-retirado(individuo recuperado o muerto)] [2, 14, 17]. El modelo SEIR es un sistema no lineal de ecuaciones diferenciales que describe la dinámica de transición entre cuatro compartimentos: susceptibles $S(t)$ (individuos en riesgo de contraer la enfermedad), expuestos $E(t)$ (individuos infectados pero aún no que capaces de producir una infección), infectados $I(t)$ (individuos capaces de transmitir la enfermedad), y retirados $R(t)$ (individuos recuperados o muertos).

$$\begin{aligned}
 \dot{S}(t) &= -\beta S(t)I(t) \\
 \dot{E}(t) &= \beta S(t)I(t) - \kappa E(t) \\
 \dot{I}(t) &= \kappa E(t) - \gamma I(t) \\
 \dot{R}(t) &= \gamma I(t)
 \end{aligned} \tag{4.2}$$

Las personas susceptibles contraen el virus a una tasa de $\beta I(t)$, siendo β la tasa de transmisión por día. El período de incubación $\frac{1}{\kappa}$ determina el tiempo para pasar del compartimento expuesto al infectado. El período de recuperación $\frac{1}{\gamma}$ determina el tiempo para progresar desde el compartimento infectado al recuperado. El número básico de reproducción $R_0 = \frac{\beta}{\gamma}$ es una estimación importante del crecimiento de la pandemia (COVID-19). R_0 indica cuántas infecciones genera en promedio un individuo infectado al inicio de una pandemia, cuando la mayoría de los individuos son susceptibles. Si $R_0 \leq 1$, los casos infectados están disminuyendo y la propagación de la enfermedad eventualmente llega a cero.

Para controlar la propagación de una enfermedad infecciosa, el número de reproducción puede reducirse mediante intervenciones, como la restricción de viajes, el confinamiento en el hogar o el distanciamiento social [56], [26]. Para mitigar la propagación, es necesario de reducir R_0 y para suprimir la propagación, además de reducir R_0 por debajo de uno. Sin embargo, controlar R_0 puede tener un costo social y económico significativo, lo que resulta en un problema de optimización desafiante para diseñar políticas de intervención. Stewart, Heusden y Dumont informaron recientemente una discusión en profundidad sobre el control de retroalimentación de enfermedades infecciosas en el contexto de COVID-19 (ver [56]).

En este experimento, usamos SDSI y SINDy con el control predictivo de modelos (CPM), para controlar la propagación de una enfermedad infecciosa arbitraria mediante el control directo de la tasa de transmisión $\beta = u(t)$. Usando CPM, podemos definir restricciones sobre la capacidad máxima de atención médica y adaptar un objetivo de control considerando el costo social y el costo económico especificando los pesos de la función de costo \mathbf{Q} , \mathbf{R}_u y $\mathbf{R}_{\Delta u}$. El principal desafío de CPM es el desarrollo de un modelo preciso y computacionalmente eficiente. Para identificar el sistema dinámico no lineal que describe la propagación de la enfermedad infecciosa, utilizamos los algoritmos SDSI y SINDy con control, suponiendo que el sistema dinámico tiene relativamente pocos términos activos. Debido a que estos modelos son dispersos por construcción, evitan el sobreajuste y son más eficientes computacionalmente que muchos otros modelos, por lo que pueden usarse en tiempo real y pueden identificarse a partir de cantidades relativamente pequeñas de datos de entrenamiento, en comparación con las redes neuronales. El entrenamiento y la validación del modelo, junto con el rendimiento

de los modelos para CPM, se muestran en la figura 4.6. A modo de comparación, se presentan CPM basado en SINDy y CPM basado en DMD.

El estado de nuestro sistema es $\mathbf{x} = [S, E, I, R]$, y suponemos que podemos controlar la tasa de transmisión β mediante intervenciones específicas, por lo que la entrada de control es $u(t) = \beta(t) = \beta_0 - \beta_c(t)$. Los parámetros constantes de la verdadera dinámica SEIR se establecen en $\beta_0 = 0.5$, $\gamma = 0.2$ y $\kappa = 0.2$. Primero, generamos los datos de entrenamiento con una entrada de control discreta durante 100 días. La entrada de control es una señal binaria pseudoaleatoria (PRBS) [40], que es una señal determinista con propiedades similares al ruido blanco. Esto representa diferentes intervenciones potenciales con el respectivo impacto en la tasa de transmisión. El código presentado a continuación genera los datos para la dinámica SEIR forzada.

```
% Generación de los datos
p.b = 0.5; p.g = 0.2; p.k = 0.2; % beta0, gamma, k
n = 4; % Número de estados
x0 = [0.996, 0.002, 0.002]; % Condiciones iniciales
tspan = 0:0.1:100;
u = prbs(tspan); % Función de forzamiento PRBS
[t,x]=ode45(@(t,x) SEIR(t,x,u,p),tspan,x0);
% Cálculo de las derivadas verdaderas
for i=1:length(x)
    dx(i,:) = SEIR(0,x(i,:),u(i),p);
end
%% Modelo SEIR
function dx = SEIR(t,x,u,p)
    S = x(1); E = x(2); I = x(3); R = x(4);
    dx = [-u*S*I; u*S*I-p.k*E; p.k*E-p.g*I; p.g*I];
end
```

Después de recopilar los datos, ejecutamos SDSI o SINDy con el algoritmo de control para identificar un modelo no lineal disperso. Primero, construimos la biblioteca de funciones candidatas Θ . Aquí, usamos polinomios hasta el tercer orden. Establecemos el hiperparámetro de esparsificación en $\lambda = 0.1$ y ejecutamos el algoritmo SINDy utilizando el enfoque de mínimos cuadrados con truncado secuencial. El código presentado a continuación ejecuta SINDy con las funciones `poolData` y `sparsifyDynamics` que se presentan en [38], de manera similar con SDSI ejecutando el código `SpSolver.m` que se presenta en [64], con la técnica de mínimos cuadrados lineales dispersos.

```
% Build library and compute sparse regression
Theta = poolData(x,n,3); % Up to 3rd order polynom
lambda = 0.1; % Sparsification hyperparameter
Xi = sparsifyDynamics(Theta,dx,lambda,n); % STLS
Xi2 = SpSolver(Theta,dx,100,tol,1e-6,1e-1)
```

=====

Cálculo de la identificación dispersa con SINDy:
 Elapsed time is 0.080979 seconds.

=====
 Cálculo de identificación dispersa con SpSolver:
 Elapsed time is 0.060254 seconds.

=====
 Dinámica identificada SDSI

Xi2 =

```
(33,1)    -1.0000
(3,2)     -0.2000
(33,2)    1.0000
(3,3)     0.2000
(4,3)    -0.2000
(4,4)     0.2000
```

=====
 Dinámica identificada SINDy

Xi =

```
(33,1)    -1.0000
(3,2)     -0.2000
(33,2)    1.0000
(3,3)     0.2000
(4,3)    -0.2000
(4,4)     0.2000
```

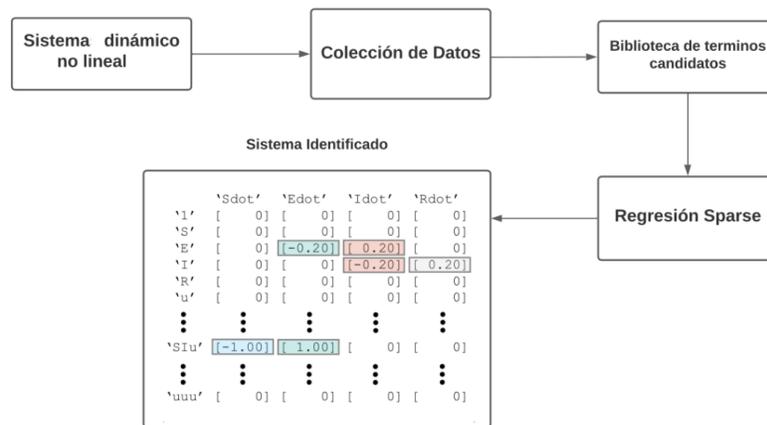


Figure 4.5: Esquema del algoritmo SINDy o SDSI con control. Se muestra el proceso para la identificación del sistema. Fuente: Elaboración propia.

El objetivo del control es minimizar el número de individuos infectados con el menor gasto de control. Además, dada la capacidad máxima del sistema de atención médica (por ejemplo, la cantidad de camas de UCI), definimos una restricción en la cantidad de personas infectadas. Aquí, la referencia es $\mathbf{r} = (0, 0, 0, 0)$ y las matrices de peso son $\mathbf{Q} = \text{diag}(0, 0, 1, 0)$ y $\mathbf{R}_u = \mathbf{R}_{\Delta u} = 0.1$. La entrada de control se limita a $u = [0.15, 0.5]$, el control se actualiza una vez por semana y el horizonte de control y predicción es $m_p = m_c = 14$ días. La restricción sobre el número máximo de personas infectadas se establece en el 5% de la población total.

En la figura 4.6, se muestran los resultados de la intervención de enfermedades infecciosas del CPM. Usamos SINDy para identificar modelos no lineales y comparar SINDy-CPM con DMD-CPM lineal. A la izquierda de la figura 4.6 en el panel a), mostramos la identificación (entrenamiento, fondo gris) y predicción (prueba, fondo blanco) para DMD y SINDy. Las entradas de control son las mismas PRBS para SINDy y DMD, que se muestran en la parte inferior. Vemos que el modelo SINDy predice perfectamente la verdadera dinámica SEIR (solo mostramos la población infecciosa para mayor claridad). El modelo DMD lineal es inestable, diverge después de 150 días y tiene un rendimiento predictivo bajo en comparación con el modelo SINDy. Por lo tanto, podemos suponer que el modelo DMD funcionará mal en CPM.

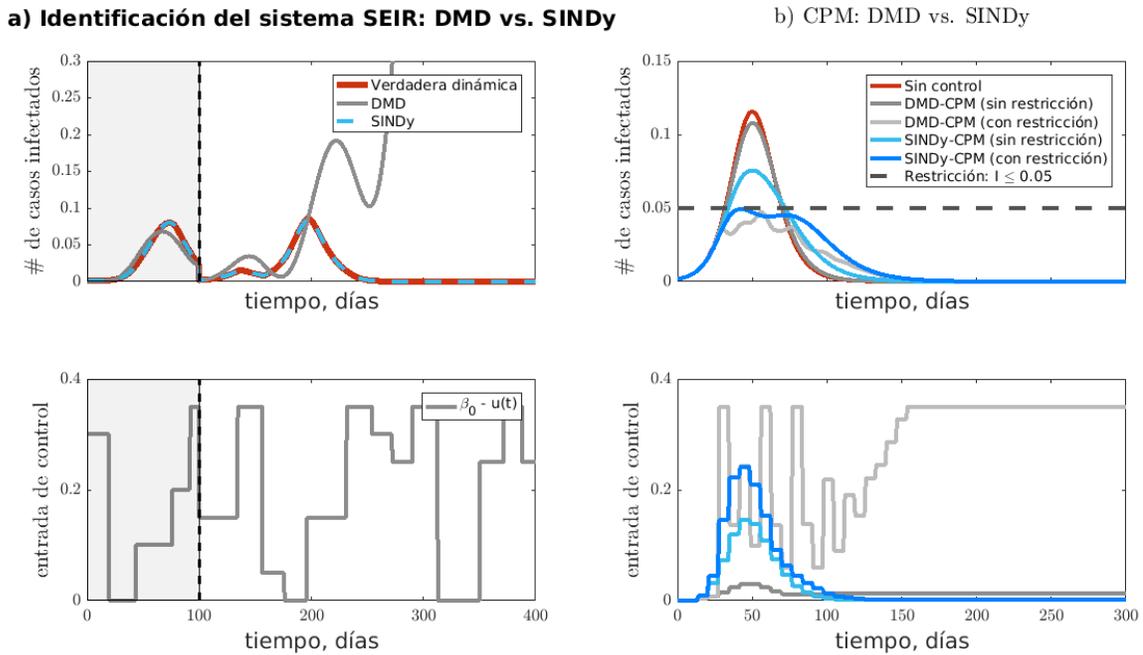


Figure 4.6: Experimento 3. SINDy-CPM vs. DMD-CPM para el problema de control de enfermedades infecciosas. Fuente: Elaboración propia.

En el panel b) de la figura 4.6 mostramos el desempeño del SINDy-CPM y lo comparamos con el DMD-CPM. Ejecutamos ambos métodos con y sin restringir el número máximo de individuos infecciosos (restricción: $I \leq 0.05$, o 5% de la población total). Primero, observamos que el SINDy-CPM sin la restricción puede reducir el número de casos infecciosos de 11,5% a 7,5%. Después de agregar la restricción, el SINDy-CPM puede reducir con éxito la cantidad de casos infecciosos por debajo del 5%. Observamos que el modelo DMD predice el efecto

de la actuación: el modelo sugiere que más intervenciones no reducirían significativamente el número de casos y, por lo tanto, mantiene el control y las intervenciones en un nivel más bajo. Para el caso restringido, DMD-CPM puede reducir los casos por debajo del 5%. La estrategia es más descoordinada y de mayor costo en comparación con el SINDy-CPM. Sin embargo, dada la poca precisión predictiva del modelo DMD, el DMD-CPM funciona sorprendentemente bien, al menos durante el primer tercio de la propagación de la enfermedad.

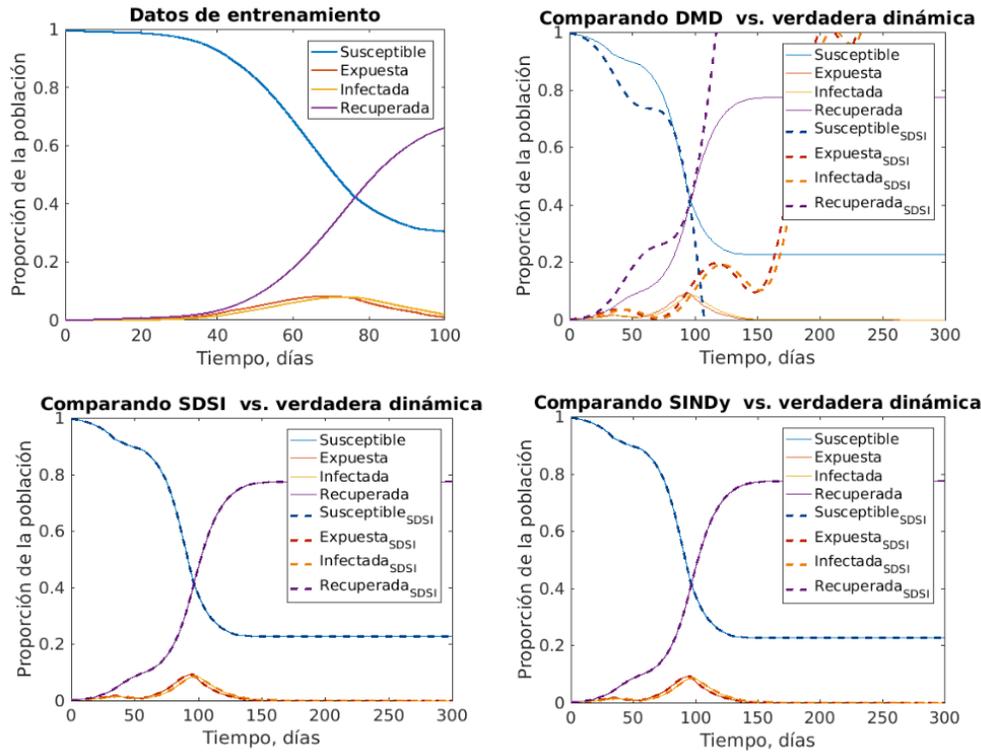


Figure 4.7: Experimento 3. Comparación de la dinámica verdadera versus cada uno de los modelos de identificación, podemos notar que tanto SDSI y SINDy logran una buena identificación de la dinámica verdadera, en el caso de SDSI se tiene que el valor del invariante topológico es igual a cero ($\text{drk}_{*,\delta}(\Gamma_T) = 0$). Fuente: Elaboración propia.

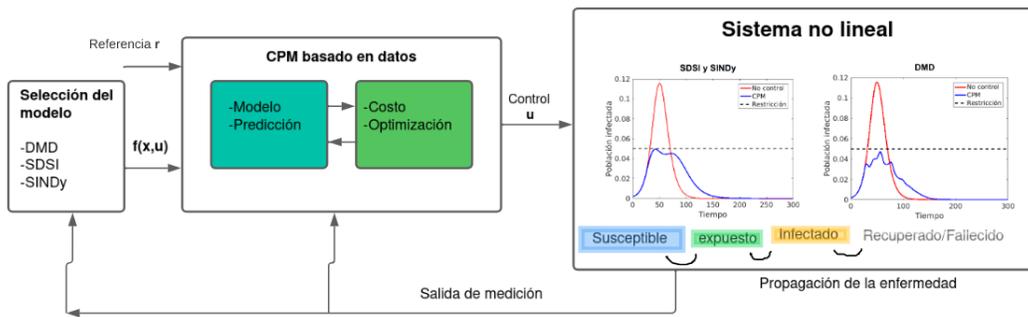


Figure 4.8: Experimento 3. Esquema del marco CPM basado en datos. Uso de DMD lineal, SDSI y SINDy no lineal disperso el control predictivo. Fuente: Elaboración propia.

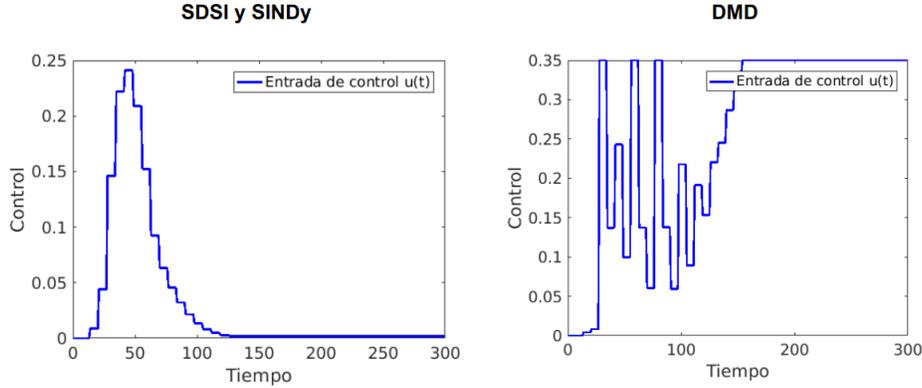


Figure 4.9: Entrada de control. Fuente: Elaboración propia.

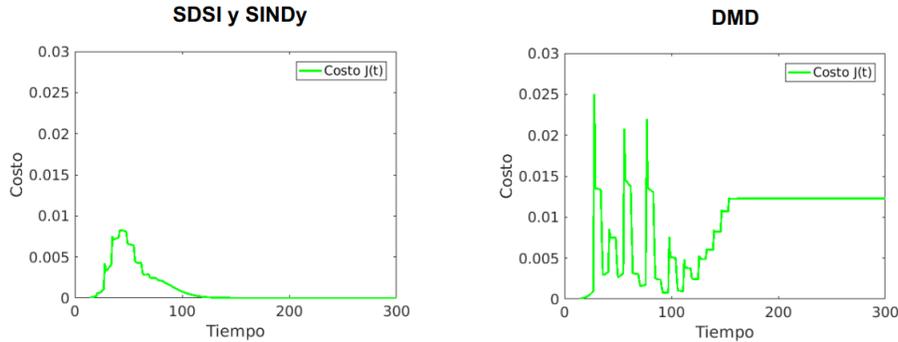


Figure 4.10: Costo. Fuente: Elaboración propia.

Podemos concluir que los modelos DMD pueden ser de gran utilidad, incluso si su rendimiento predictivo es pobre. Además, estos modelos pueden entrenarse con muy pocos datos. Por lo tanto, pueden ser útiles en el límite de datos bajos hasta que se recopilen suficientes datos para identificar modelos SDSI o SINDy precisos para el control. Como se concluyó en [33], SINDy-CPM proporciona un control no lineal efectivo y eficiente. En esta tesis se muestra que SDSI-CPM compite de manera favorable con SINDy-CPM, logrando resultados eficiente en la identificación de sistemas, con DMD como un recurso provisional hasta que se pueda identificar un modelo SDSI o SINDy.

4.4 Experimento 4

En esta sección se aplican algunas ideas de Boyd y Vandenberghe [6] sobre dinámica epidémica, describimos un experimento donde la dinámica de la infección y la propagación de una epidemia se pueden modelar utilizando un sistema dinámico lineal. En el experimento presentado en esta sección una enfermedad se introduce en una población, en cada período (digamos, días) contamos la fracción de la población que se encuentra en cuatro estados diferentes de infección:

- Susceptible. Estos individuos pueden adquirir la enfermedad al día siguiente.

- Infectado. Estos individuos tienen la enfermedad.
- Recuperado (e inmune). Estos individuos tenían la enfermedad y sobrevivieron, y ahora tienen inmunidad.
- Fallecido. Estas personas tenían la enfermedad y, lamentablemente, murieron a causa de ella.

El modelo mencionado anteriormente también se le conoce como modelo SIRD (Susceptible, Infected, Recovered, Deceased). A continuación se muestra un modelo simple que expresa como un sistema dinámico lineal de forma matemática predice cómo las fracciones del estado de una enfermedad evoluciona con el tiempo. El modelo supone que sucede lo siguiente cada día.

- El 6% de la población susceptible adquirirá la enfermedad. (El otro 94% seguirá siendo susceptible).
- El 1% de la población infectada morirá a causa de la enfermedad, el 16% se recuperará y adquirirá inmunidad, y el 3% se recuperará y no adquirirá inmunidad (y por lo tanto, se volverá susceptible). El 80% restante seguirá infectado.
- Los que se han recuperado con inmunidad y los que han muerto permanecen en esos estados.

Primero determinamos $(x_{t+1})_1$, la fracción de individuos susceptibles al día siguiente. Estos incluyen las personas susceptibles de hoy, que no se infectaron, que es $0.94(x_t)_1$, más las personas infectadas de hoy que se recuperaron sin inmunidad, que es $0.03(x_t)_2$. Todos juntos tenemos $(x_{t+1})_1 = 0.94(x_t)_1 + 0.03(x_t)_2$. También se tiene $(x_{t+1})_2 = 0.80(x_t)_2 + 0.06(x_t)_1$; el primer término cuenta los que están infectados y permanecen infectados, y el segundo término cuenta los que son susceptibles y adquieren la enfermedad. Argumentos similares dan $(x_{t+1})_3 = (x_t)_3 + 0.16(x_t)_2$, y $(x_{t+1})_4 = (x_t)_4 + 0.01(x_t)_2$. Al juntar lo antes mencionado se tiene

$$x_{t+1} = \begin{bmatrix} 0.94 & 0.03 & 0 & 0 \\ 0.06 & 0.80 & 0 & 0 \\ 0 & 0.16 & 1 & 0 \\ 0 & 0.01 & 0 & 1 \end{bmatrix} x_t,$$

el cual es un sistema dinámico lineal invariante en el tiempo y de la forma $x_{t+1} = Ax_t$. Se muestra algunas partes del código desarrollado en GNU Octave para este experimento numérico con técnicas computacionales utilizadas en [63] y [64].

```
%%%%%%%%%%%%% DATOS %%%%%%%%%%%%%%
```

```
A = [.94 .03 0 0;.06 .80 0 0;0 .16 1 0;0 .01 0 1];
X=[1 0 0 0]';
for k=1:300, X(:,k+1)=A*X(:,k);end
X0 = X(:,1:(end-1));
```

```

X1 = X(:,2:end);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SDSI %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Ar1=SpSolver(X0.',X1.',10,1e-6).';
norm(Ar1*X0-X1,'fro')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SINDy %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Ar2=SINDy(X0.',X1.',10,1e-6).';
norm(Ar2*X0-X1,'fro')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Optimizacion regular %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

phi =@(x)norm(X1-reshape (x,4,4)*X0,'fro')^2;
g=@(x) [sum(x(1:4))-1;sum(x(5:8))-1;sum(x(9:12))-1;sum(x(13:16))-1;...
...x([3:4 9:10 12 13:15])];
A0 = full(sprand(16,1,.6));
[y, OBJ, INFO, ITER, NF, LAMBDA] = sqp (A0, phi, g, [], 0, [], 200, 1e-4);
Ar3=reshape (y,4,4);
norm(Ar3*X0-X1,'fro')

```

En el código anterior se utiliza la programación cuadrática secuencial (SQP), que es un método iterativo para la optimización no lineal restringida. Los métodos SQP se usan en problemas matemáticos en los que la función objetivo y las restricciones son dos veces diferenciables continuamente.

Los métodos SQP resuelven una secuencia de subproblemas de optimización, cada uno de los cuales optimiza un modelo cuadrático del objetivo sujeto a una linealización de las restricciones. Si el problema no tiene restricciones, entonces el método se reduce al método de Newton para encontrar un punto donde el gradiente del objetivo desaparece. Si el problema solo tiene restricciones de igualdad, entonces el método es equivalente a aplicar el método de Newton a las condiciones de optimalidad de primer orden, o condiciones de Karush-Kuhn-Tucker, del problema.

La figura 4.11 de la dinámica epidémica verdadera [primera imagen], muestra la evolución de los cuatro grupos desde la condición inicial $x_0 = (1, 0, 0, 0)$. La simulación muestra que después de alrededor de 100 días, el estado converge a uno con un poco menos del 10% de la población fallecida y el resto de la población inmune.

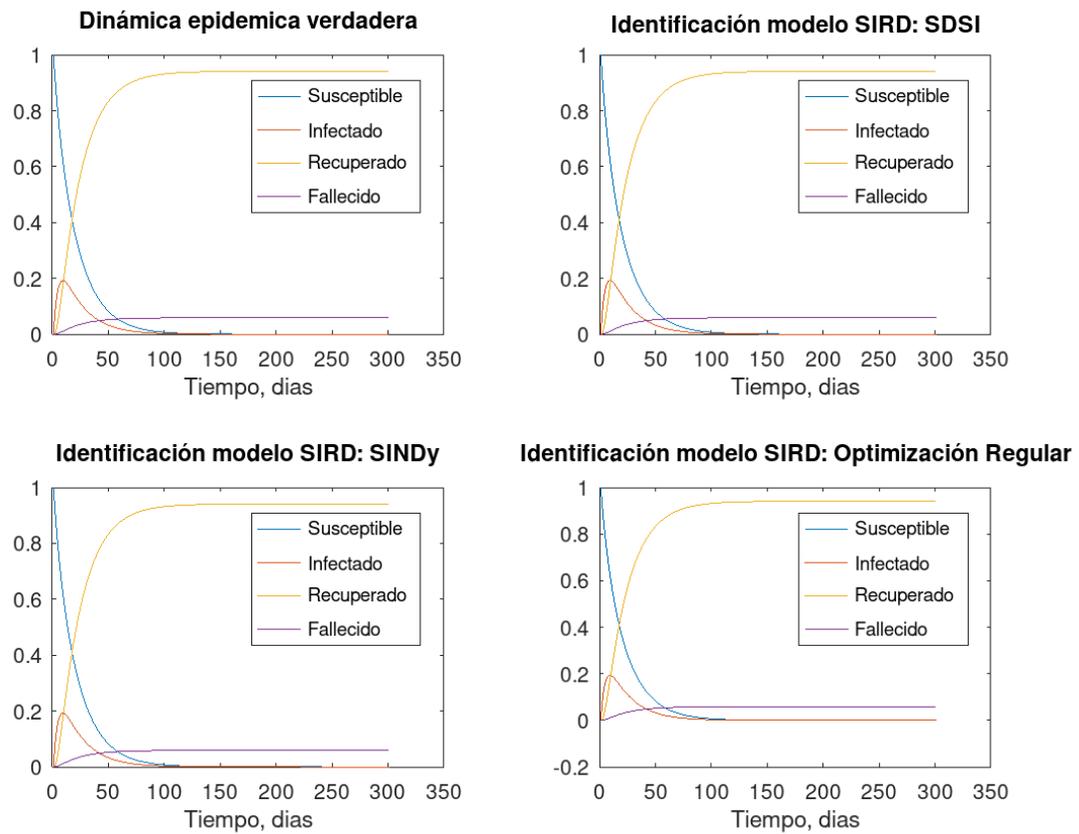


Figure 4.11: Experimento 4. Identificación del modelo SIRD mediante SINDy, SDSI y optimización regular. Fuente: Elaboración propia.

```

=====
Cálculo del modelo SDSI
Error de aproximación SDSI:
ans = 1.8696e-15
=====
Cálculo del modelo SINDy
Error de aproximación SINDy:
ans = 2.6835e-15
=====
Cálculo del modelo de optimización regular
Error de aproximación regular:
ans = 0.0000025875
=====

```

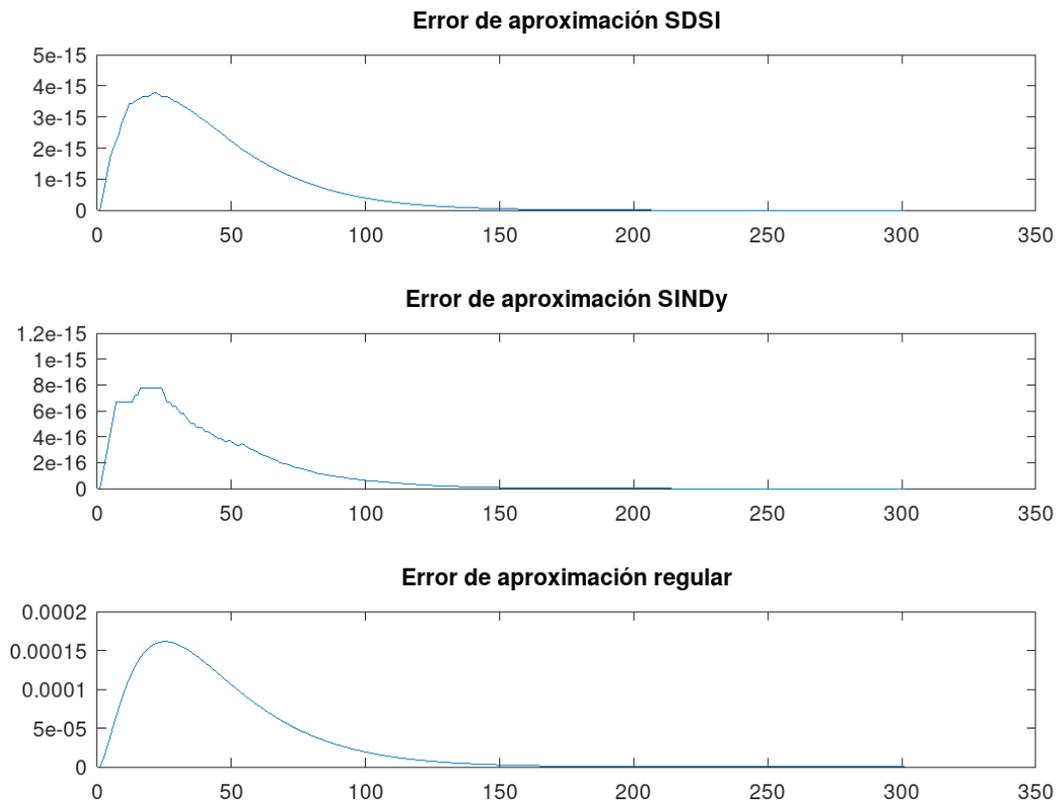


Figure 4.12: Experimento 4. Modelo SIRD, errores de aproximacion. Fuente: Elaboración propia.

Podemos notar que el método propuesto en este experimento de optimización regular logra capturar de buena manera cierta característica del modelo tal como se muestra a continuación con los valores propios

=====

```
eig(A)
```

```
ans =
```

```
1.00000
```

```
1.00000
```

```
0.95185
```

```
0.78815
```

```
>> eig(Ar1)
```

```
ans =
```

```
0.00000
```

```

1.00000
0.78815
0.95185

>> eig(Ar2)
ans =

7.8815e-01
9.5185e-01
1.0000e+00
5.5511e-17

>> eig(Ar3)
ans =

0.78815
0.95185
1.00000
1.00000
=====

```

Los resultados mostrados en este capítulo nos permiten concluir que los modelos utilizados en los distintos experimentos pueden entrenarse con muy pocos datos. Por lo tanto, pueden ser útiles en el límite de datos bajos hasta que se recopilen suficientes datos para identificar modelos SDSI y SINDy preciso para el control. SDSI-CPM y SINDy-CPM proporciona un control no lineal efectivo y eficiente, con DMD como un recurso provisional hasta que se pueda identificar modelos SDSI y SINDy. En el siguiente capítulo se presenta las conclusiones más relevantes realizadas de este trabajo para la identificación dispersa de sistemas lineales y no lineales mediante los métodos SDSI y SINDy.

Capítulo 5

Conclusiones y Trabajos futuros

5.1 Conclusiones

En esta sección se presentan las conclusiones más relevantes de este trabajo.

- La recopilación de métodos del capítulo 3 en forma teórica y de algoritmos, como lo descrito en el capítulo 4 en forma de experimentos y resultados computacionales, se pueden utilizar de manera efectiva para la identificación dispersa de modelos dinámicos epidemiológicos que se pueden usar para calcular simulaciones numéricas predictivas basadas en datos.
- Se mostró cómo SINDy o SDSI con control se pueden combinar con CPM para el control de enfermedades infecciosas. También es interesante comparar DMD-CPM y SINDy-CPM con CPM basado en una red neuronal. Las redes neuronales requieren significativamente más datos de entrenamiento y tienen un mayor tiempo de ejecución en comparación con DMD, SINDy y SDSI. Sin embargo, proporcionan una representación más flexible de la dinámica cuando la estructura del modelo varía en el espacio de estado [33].
- Como se mostró en el capítulo 4, SDSI-CPM tiene un potencial significativo para el control de sistemas no lineales. Además, los rápidos tiempos de entrenamiento y ejecución indican que SDSI puede ser útil para la identificación rápida de modelos en respuesta a cambios abruptos en el modelo, y esto justifica una mayor investigación. La capacidad de identificar modelos precisos y eficientes con pequeñas cantidades de datos de entrenamiento puede ser un facilitador clave de la recuperación de datos en escenarios de tiempo crítico, como cambios de modelo que conducen a la inestabilidad.

5.2 Trabajos futuros

Como consecuencia del teorema 3.7, es posible abordar los problemas de identificación de sistemas como problemas de completación de matrices aproximadas a través de la aproximación de matrices de bajo rango, las conexiones correspondientes con la aproximación de

matrices de bajo rango local en el sentido de [39] se consideran como direcciones futuras de este trabajo.

Como observó Koch en [35], al aplicar técnicas de identificación de sistemas dispersos como las presentadas en este trabajo, llegar a un modelo de alta calidad estará fuertemente relacionado con la elección de funciones en la biblioteca del método, lo que puede requerir conocimiento experto o ajuste, y esta consideración se alinea perfectamente con la filosofía detrás del desarrollo de la tecnología de identificación de sistema presentada como parte de los resultados presentados en este trabajo.

Las implementaciones computacionales de los solucionadores dispersos y los algoritmos de identificación presentados en este trabajo para el cálculo de modelos de señales dispersos y compresiones de señales, usando submatrices dispersas rectangulares de matrices wavelet, también presentan posibles trabajos futuros. Las conexiones de los resultados en la sección 3.2.2 con la solución de problemas relacionados con la controlabilidad y que se pueden realizar sistemas de estado finito en información clásica, cuántica y teoría de autómatas en el sentido de [8, 59, 7, 3], se identifican como posibles líneas de exploración.

Se explorarán más aplicaciones de esquemas de identificación de modelos de señales dispersas para automatización industrial y tecnologías de modelado de información en construcciones.

Bibliografía

- [1] ALKHOURY, Z., PETRECZKY, M., AND MERCERE, G. Identifiability of affine linear parameter-varying models. Automatica 80 (2017), 62–74.
- [2] ANDERSON, R., AND MAY, R. Infectious diseases of humans: dynamics and control. Oxford University Press, 1991.
- [3] BLOCH, A., BROCKETT, R., AND RANGAN, C. Finite controllability of infinite-dimensional quantum systems. IEEE Transactions on Automatic Control 55, 8 (2010), 1797–1805.
- [4] BONGARD, J., AND LIPSON, H. Automated reverse engineering of nonlinear dynamical systems. Proceedings of the National Academy of Sciences (2007), 9943–9948.
- [5] BOUTSIDIS, C., AND MAGDON-ISMAIL, M. A note on sparse least-squares regression. Information Processing Letters 114(5) (2014), 273–276.
- [6] BOYD, S., AND VANDENBERGHE, L. Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares. Cambridge University Press, 2018.
- [7] BROCKETT, R. Reduced complexity control systems. IFAC Proceedings Volumes 41, 2 (2008), 1–6.
- [8] BROCKETT, R., AND WILLSKY, A. Finite group homomorphic sequential system. IEEE Transactions on Automatic Control 17(4) (1972), 483–490.
- [9] BRUNTON, S., AND KUTZ, N. On dynamic mode decomposition: Theory and applications. Nonlinear Dynamics 1, 2 (2014), 391–421.
- [10] BRUNTON, S., KUTZ, N., BRUNTON, B., AND PROCTOR, J. Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems. Society for Industrial and Applied Mathematics, 2016.
- [11] BRUNTON, S., KUTZ, N., AND PROCTOR, J. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proceedings of the National Academy of Sciences, 113 (2016), 3932–3937.
- [12] BRUNTON, S., KUTZ, N., AND PROCTOR, J. Dynamic mode decomposition with control. SIAM J Appl. Dyn. Syst 15(1) (2016), 142–161.

- [13] CAMACHO, E., AND ALBA, C. Model predictive control. Springer Science and Business Media, 2013.
- [14] CASTILLO-CHAVEZ, C., AND VELASCO-HERNANDEZ, J. Modeling contact structures in biology, in frontiers in mathematical biology. Springer (1994), 454–491.
- [15] CHARTRAND, R. Numerical differentiation of noisy, nonsmooth data. ISRN Applied Mathematics 2011 (2011), 11.
- [16] CHEN, J., AVRON, H., AND SINDHWANI, V. Hierarchically compositional kernels for scalable nonparametric learning. Journal of Machine Learning Research 18(66) (2017), 1–42.
- [17] CHOWELL, G., HENGARTNER, N., CASTILLO-CHAVEZ, C., FENIMORE, P., AND HYMAN, J. M. The basic reproductive number of ebola and the effects of public health measures: the cases of congo and uganda. Journal of theoretical biology 229, 1 (2004), 119–126.
- [18] DONOHO, D. Compressed sensing. IEEE Transactions on information theory 54(2) (2006), 1289–1306.
- [19] EVANS, L. C. Partial differential equations and monge-kantorovich mass transfer. Current Developments in Mathematics 1997, 1 (1999), 65–126.
- [20] FARHOOD, M., AND DULLERUD, G. Lmi tools for eventually periodic systems. Systems and Control Letters 47 (2002), 417–432.
- [21] FINZI, M., STANTON, S., IZMAILOV, P., AND WILSON, A. G. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. ICML (2020).
- [22] FRANKE, J., AND SELGRADE., J. Attractors for discrete periodic dynamical systems. Journal of Mathematical Analysis and Applications 286(1) (2003), 64–79.
- [23] FRANKE, J., AND SELGRADE, J. Attractors for discrete periodic dynamical systems. Journal of Mathematical Analysis and Applications 286 (2003), 64–79.
- [24] FREEDMAN, M., AND PRESS, W. Truncation of wavelet matrices: Edge effects and the reduction of topological control. Linear Algebra and its Applications 234 (1996), 1–19.
- [25] GOLUB, G. H. Matrix Computations. Johns Hopkins University Press, USA, 1996.
- [26] GRIMM, V., MENGEL, F., , AND SCHMIDT, M. Extensions of the seir model for the analysis of tailored social distancing and tracing approaches to cope with covid-19. Scientific Reports 11, 1 (2021).
- [27] GUCKENHEIMER, J. From data to dynamical systems. Nonlinearity 27 (2014), 41–50.

- [28] GUCKENHEIMER, J., AND HOLMES, P. Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields. Applied Mathematical Sciences (Springer-Verlag, Berlin) Vol. 42., 1983.
- [29] HORN, R. A. Topics in Matrix Analysis. Cambridge University Press., USA, 1986.
- [30] JAMES, G. An introduction to statistical learning. Springer, USA, 2013.
- [31] JORDAN, M., AND MITCHELL, T. Machine learning: Trends, perspectives, and prospects.
- [32] KAHEMAN, K., KUTZ, N., AND BRUNTON, S. Sindy-pi: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. SIAM J Appl. Dyn. Syst 476(2242) (2020), 20200279.
- [33] KAISER, E., KUTZ, J., AND BRUNTON, S. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 474(2219) (2018), 20180335.
- [34] KEVREKIDIS, I. Equation-free, coarse-grained multiscale computation: Enabling microscopic simulators to perform system-level analysis. Communications in Mathematical Science 1 (2003), 715–762.
- [35] KOCH, J. Data-driven modeling of nonlinear traveling waves. Chaos: An Interdisciplinary Journal of Nonlinear Science 31, 4 (2021), 043128.
- [36] KOOPMAN, B. O. Hamiltonian systems and transformation in hilbert space. Proceedings of the National Academy of Sciences 17, 5 (1931), 315–318.
- [37] KOOPMAN, B. O., AND V. NEUMANN, J. Dynamical systems of continuous spectra. Proceedings of the National Academy of Sciences 18, 3 (1932), 255–263.
- [38] KUTZ, N. Data-Driven Modeling Scientific Computation. Methods for Complex Systems Big Data. Oxford University Press, Oxford, 2013.
- [39] LEE, J., KIM, S., LEBANON, G., SINGER, Y., , AND BENGIO, S. Llorma: Local low-rank matrix approximation. J. Mach. Learn. Res. 17, 1 (2016), 442–465.
- [40] LJUNG, L. System Identification: Theory for the User. Prentice-Hall, Englewood Cliffs, 1999.
- [41] LORENZ, E. Deterministic nonperiodic flow. Journal of the Atmospheric Sciences 20(2) (1963), 130–141.
- [42] LORING, T., AND VIDES, F. Computing floquet hamiltonians with symmetries. Journal of Mathematical Physics 61(11) (2020), 113501.
- [43] LY, H., AND TRAN, H. Proper orthogonal decomposition for flow calculations and optimal control in a horizontal cvd reactor. Quarterly of Applied Mathematics 60(4) (2002), 631–656.

- [44] MANGAN, N., KUTZ, N., BRUNTON, S., AND PROCTOR, J. Model selection for dynamical systems via sparse regression and information criteria. Proceedings of the Royal Society A 473 (2017), 1–16.
- [45] MAYNE, D. Model predictive control: Recent developments and future promise. Automatica 50(2) (2014), 2967–2986.
- [46] MEZIĆ, I. Spectral properties of dynamical systems, model reduction and decompositions. Nonlinear Dynamics 41, 3 (2005), 309–325.
- [47] MEZIĆ, I. Analysis of fluid flows via spectral properties of the koopman operator. Annual Review of Fluid Mechanics 45 (2013), 357–378.
- [48] MOSKOVINA, V., AND SCHMIDT, K. Approximate projectors in singular spectrum analysis. SIAM Journal on Matrix Analysis and Applications 24(4) (2003), 932–942.
- [49] PACKARD, N. Geometry from a time series. Phys. Rev. Lett 45 (1980), 712–716.
- [50] ROWLEY, C. W. Spectral analysis of nonlinear flows. J. Fluid Mech 641 (2009), 1–13.
- [51] SALOVA, A., EMENHEISER, J., RUPE, A., CRUTCHFIELD, J. P., AND D’SOUZA, R. M. Koopman operator and its approximations for systems with symmetries. Chaos: An Interdisciplinary Journal of Nonlinear Science 29(9) (2019), 093128.
- [52] SCHAEFFER, H., TRAN, G., WARD, R., AND ZHANG, L. Extracting structured dynamical systems using sparse optimization with very few samples. Multiscale Modeling and Simulation, 18(4) (2020), 1435–1461.
- [53] SCHICHL, H. Models and History of Modeling. Modeling Languages in Mathematical Optimization, 2004.
- [54] SCHMID, P. Dynamic mode decomposition of numerical and experimental data. Journal of Fluid Mechanics 656 (2010), 5–28.
- [55] SCHMIDT, M., AND LIPSON, H. Distilling free-form natural laws from experimental data. Science 324 (2009), 81–85.
- [56] STEWART, G., HEUSDEN, K., , AND DUMONT, G. How control theory can help us control covid-19. IEEE Spectrum 57, 6 (2020), 22–29.
- [57] SU, W., BOGDAN, M., AND CANDES, E. False discoveries occur early on the lasso path. arXivpreprint arXiv:1511.01957 (2016).
- [58] TAKENS, F. Detecting strange attractors in turbulence. Dynamical systems and turbulence, Warwick 1980. Springer (1981), 366–381.
- [59] TARRAF, D. An input-output construction of finite state ρ/μ approximations for control design. IEEE Transactions on Automatic Control 59, 12 (2014), 3164–3177.

- [60] TREFETHEN, L., AND BAU, D. Matrix Computations. Society for Industrial and Applied Mathematics (SIAM), USA, 1997.
- [61] TROPP, J. Just relax: Convex programming methods for identifying sparse signals in noise. IEEE transactions on information theory 52, 3 (2006), 1030–1051.
- [62] VIDES, F. On cyclic finite-state approximation of data-driven systems. IEEEEXPlore CONCAPAN 2019 (2019).
- [63] VIDES, F. Métodos Numéricos y Modelación Computacional. <https://caddslab.github.io/MNMC.pdf>, (CICC-UNAH), Honduras, 2021.
- [64] VIDES, F. SDSI: A toolset with matlab, python and julia programs for approximate sparse system identification. <https://github.com/FredyVides/SDSI>, Honduras, 2021.
- [65] VIDES, F. Sparse system identification by low-rank approximation. arXivpreprint arXiv:2105.07522 (2021).
- [66] WANG, W., LAI, Y.-C., AND GREBOGI, C. Data based identification and prediction of nonlinear and complex dynamical systems. Physics Reports 644 (2016), 1–76.
- [67] ZHANG, L., AND SCHAEFFER, H. On the convergence of the sindy algorithm. Multiscale Modeling and Simulation 17, 3 (2019), 948–972.
- [68] ZHENG, P., ASKHAM, T., BRUNTON, S., KUTZ, N., AND ARAVKIN, A. A unified framework for sparse relaxed regularized regression: Sr3. IEEE Access 7 (2019), 1404–1423.