Universidade Federal da Paraíba
Centro de Informática
Programa de Pós-Graduação em Modelagem Matemática e Computacional

MÉTODOS NUMÉRICOS PARA PREVISÃO EM SISTEMAS DINÂMICOS

Wilson Salustiano Júnior

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Modelagem Matemática e Computacional, UFPB, da Universidade Federal da Paraíba, como parte dos requisitos necessários à obtenção do título de Mestre em Modelagem Matemática e Computacional.

Orientador: Hugo Leonardo Davi de Souza Cavalcante

João Pessoa
Dezembro de 2019

Catalogação na publicação
Seção de Catalogação e Classificação

S181m Salustiano Júnior, Wilson.
Métodos Numéricos para Previsão em Sistemas Dinâmicos / Wilson Salustiano Júnior. - João Pessoa, 2019.
100 f. : il.

Orientação: Hugo Leonardo Davi de Souza Cavalcante.
Dissertação (Mestrado) - UFPB/Informática.

1. Dinâmica Não-linear. 2. Previsão em Sistemas Dinâmicos. 3. Regressão Linear. 4. Aprendizado de máquina. 5. Caos Determinístico. I. Cavalcante, Hugo Leonardo Davi de Souza. II. Título.

UFPB/BC

MÉTODOS NUMÉRICOS PARA PREVISÃO EM SISTEMAS DINÂMICOS

Wilson Salustiano Júnior

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL (PPGMMC) DO CENTRO DE INFORMÁTICA DA UNIVERSIDADE FEDERAL DA PARAÍBA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM MODELAGEM MATEMÁTICA E COMPUTACIONAL.

Examinada por:

Hugo L. D. S. Cavalcante

Prof. Hugo Leonardo Davi de Souza C., D.Sc.

Ana Paula Wyse

Profª. Ana Paula Pintado Wyse, D.Sc.

Gilson Francisco de Oliveira Junior

Prof. Gilson Francisco de Oliveira Junior, D.Sc.

JOÃO PESSOA, PB – BRASIL
DEZEMBRO DE 2019

Resumo da Dissertação apresentada ao PPGMMC/CI/UFPB como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

MÉTODOS NUMÉRICOS PARA PREVISÃO EM SISTEMAS DINÂMICOS

Wilson Salustiano Júnior

Dezembro/2019

Orientador: Hugo Leonardo Davi de Souza Cavalcante

Programa: Modelagem Matemática e Computacional

Muitos comportamentos naturais e artificiais podem ser reproduzidos por sistemas dinâmicos descritos por equações diferenciais, cujas soluções formam séries temporais. Portanto, um problema importante é a determinação de qual modelo matemático é capaz de descrever o comportamento de um dado sistema dinâmico, e como determinar quais são valores dos parâmetros que regem este modelo. A solução desse problema nos permite obter informações sobre a evolução do sistema e a determinação de seus possíveis estados futuros. Estas informações são especialmente importantes em sistemas de difícil previsão, como é o caso de sistemas complexos ou não-lineares, que possuem grande sensibilidade a condições iniciais. Para tentar reproduzir a dinâmica de sistemas cujas equações não são conhecidas, estudaremos o sistema de Lorenz. Temos como objetivo mostrar que o conhecimento de séries temporais de dados oriundos de sistemas dinâmicos podem ser usadas para recuperar a dinâmica que origina o comportamento observado. Em seguida, usando esta dinâmica recuperada, abordaremos o problema da previsão do estado futuro. Desenvolvemos uma técnica de inteligência artificial utilizando funções polinomiais para calcular as velocidades de evolução das variáveis de estado do sistema, com o objetivo de recuperar sua dinâmica, para em seguida realizar a previsão. Veremos que o algoritmo é capaz de se adaptar ao sistema, com parâmetros flexíveis, extraídos a partir dos dados coletados em observações pregressas, mesmo quando estes dados estão contaminados por ruído observacional. Esperamos que o nosso método seja útil além da ciência básica, em várias aplicações, tais como fazer previsões de sistemas físicos, biológicos, financeiros, geográficos, meteorológicos, etc.

Palavras-chave: Dinâmica Não-linear. Previsão em Sistemas Dinâmicos. Regressão Linear. Aprendizado de máquina. Caos Determinístico.

Abstract of Dissertation presented to PPGMMC/CI/UFPB as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

NUMERICAL METHODS FOR FORECASTING IN DYNAMICAL SYSTEMS

Wilson Salustiano Júnior

December/2019

Advisor: Hugo Leonardo Davi de Souza Cavalcante

Program: Computational Mathematical Modelling

Many natural and artificial behaviors can be reproduced by dynamical systems described by differential equations, whose solutions are time series. Therefore, an important problem is how to determine which mathematical model is able of describing the behavior of a given dynamical system, as well as how to determine what are the values of the parameters governing this model. Solving this problem enables one to obtain information about the evolution of the system and to determine its possible future states. This information is especially important when dealing with systems of difficult forecast, as is the case of complex or nonlinear systems, which are extremely sensitive to their initial conditions. To try to reproduce the dynamics of systems whose equations are not known, we will study the Lorenz system. Our goal is to show that knowledge of time series data from dynamical systems can be used to recover the dynamics originating the observed behavior. Then, using this recovered dynamics, we will address the problem of predicting the future state on time series. We have developed a technique of artificial intelligence using polynomial functions to calculate the evolution velocities of the state variables of the system, aiming to recover its underlying dynamics, and then forecast its future evolution. The algorithm is capable of adapting to the system, with flexible parameters extracted from the previously observed data, even when the time series is contaminated by observation noise. We hope that our method will be useful beyond basic science in many applications, such as making predictions of systems that may be physical, biological, financial, geographic, meteorological, etc.

Keywords: Nonlinear Dynamics. Prediction is Dynamical Systems. Linear Regression. Machine Learning. Deterministic Chaos.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

$E_{\%}$	Erro percentual, p. 44
O	Média de uma distribuição normal, p. 53
$\frac{\partial}{\partial t}$	Derivada parcial com relação a variável t, p. 11
$\frac{d}{dt}$	Derivada temporal, p. 5
λ	expoente de Liapunov, p. 22
$\mathbb{N}$	Conjuntos dos números naturais, p. 26
$\mathbb{R}$	Conjunto dos números reais, p. 5
$\mathbb{Z}$	Conjunto dos números inteiros, p. 9
$\mathbb{Z}_+$	Conjuntos dos números inteiros não-negativos, p. 26
$\mathbf{D}$	Desvio padrão de uma distribuição normal, p. 53
μ	Dimensão de imersão, p. 36
$\sum$	Somatório, p. 28
τ	Passo de reconstrução do atrator, p. 36
$\text{int}(*)$	Arredondamento de um número real para o inteiro mais próximo., p. 66
$d(,\#)$	Distância entre dois vetores, p. 28
t	Variação temporal, p. 5

Capítulo 1

Introdução

Muitos comportamentos naturais e artificiais podem ser reproduzidos matematicamente através de um modelo que determina a evolução temporal do estado do sistema. Estes modelos de sistemas dinâmicos frequentemente são dados por equações diferenciais cujas soluções formam séries temporais (coleção de observações feitas sequencialmente ao longo do tempo) das variáveis dinâmicas observáveis associadas ao estado do sistema.

Muitas vezes é complicado determinar qual modelo matemático é capaz de exibir o comportamento físico desejado, especialmente quando tratamos de sistemas complexos, nos quais há várias partes que interagem de maneira não-linear. Um sistema é dito linear quando satisfaz o principio da superposição, ou seja, possui as propriedades de aditividade e homogeneidade. Caso contrário, o sistema é não-linear.

> A maior parte da vida cotidiana é não-linear e o principio da superposição falha espetacularmente. Se você ouvir suas duas músicas favoritas ao mesmo tempo, você não terá o dobro do prazer! Dentro do campo da física, a não-linearidade é vital para a operação de um laser, a formação de turbulência em um fluido e a supercondutividade das junções de Josephson. (STROGATZ, 1994, p.9, tradução nossa).[1]

Em outras situações ocorre comportamento caótico no qual o estado é fortemente influenciado por pequenas perturbações (ruídos ou pequenos erros) nas condições iniciais, que podem resultar numa grande diferença em tempos posteriores, dificultando a tarefa de prever o seu estado e que faz com que seja incorretamente interpretado como um sistema aleatório. Isso ocorre pela amplificação da incerteza sobre o estado do sistema dada pelas leis que regem a sua evolução. Ademais, há também a presença de ruídos observacionais e dinâmicos, não-estacionariedade (deriva dos valores) de parâmetros, e os efeitos de um possível número elevado de graus

[1] Most of everyday life is nonlinear, and the principle of superposition fails spectacularly . If you listen to your two favorite songs at the same time, you won't get double the pleasure! Within the realm of physics, nonlinearity is vital to the operation of a laser, the formation of turbulence in a fluid, anda the superconductivity of Josephson junctions.

de liberdade na evolução do sistema. As leis dinâmicas que daremos atenção neste trabalho são aquelas de natureza não-linear com análise de contaminações por ruídos observacionais.

A previsibilidade do comportamento de sistemas caóticos é uma área de grande importância, porque muitos fenômenos do mundo real apresentam algum tipo de comportamento complexo, e alguma lei determinística para evolução de seu estado, fazendo com que seja similar ao comportamento caótico. Várias áreas do conhecimento tem interesse no estudo da predição em séries temporais, dentre elas, em engenharia elétrica CAMPOS (2008), clima COSTA *et al.* (2015), finanças CORDEIRO JR. (2007), entre outros. Muitos trabalhos tem sido desenvolvidos com o objetivo de conseguir realizar ou aperfeiçoar métodos de previsão em sistemas caóticos, dentre eles,YANG *et al.* (2012), LU *et al.* (2018), FONTES (2013), PATHAK *et al.* (2018), PECORA *et al.* (2007), WANG *et al.* (2016), DE S. CAVALCANTE *et al.* (2013).

Técnicas modernas de inteligência artificial, tais como ciência de dados, aprendizagem de máquina, amostragem compressiva, e computação em reservatório, tem se revelado eficazes em situações de alta complexidade, resolvendo problemas que constituem desafios para a matemática e computação até há pouco. Por exemplo, em TRAN e WARD (2016) são apresentadas algumas técnicas de otimização baseadas no que chamam de "Theory from Compressive Sensing" para a recuperação de sistemas caóticos provido de dados corrompidos. Em YANG *et al.* (2012) é utilizado expansão em série em variáveis dinâmicas e de tempo como técnica de problema de previsão. Já em LU *et al.* (2018), apresentam uma abordagem de aprendizado de máquina chamada "Reservoir Computing".

Nesse contexto, torna-se interessante desenvolver uma técnica que procure determinar qual conjunto de equações diferenciais pode reproduzir a dinâmica dos dados observados. Uma vez descoberto este sistema equivalente, existe a possibilidade de se resolver as equações para fazer estimativas sobre o seu estado futuro.

Uma das ideias mais antigas para tentar encontrar algo que sirva para determinar um conjunto de equações é utilizar polinômios que sejam a aproximação das funções.

> [...]É bastante fácil entender por que razão isso acontece. Os polinômios são facilmente computáveis, suas derivadas e integrais são novamente polinômios, suas raízes podem ser encontradas com relativa facilidade, etc.
>
> A simplicidade dos polinômios permite que a aproximação polinomial seja obtida de vários modos, entre os quais podemos citar: Interpolação, Método dos Mínimos Quadrados, Osculação, Min-Max, etc, portanto é vantajoso substituir uma função complicada por um polinômio que a represente. Além disso temos o Teorema de Weirstrass que afirma que: toda função continua pode ser arbitrariamente aproximada por um polinômio. (FRANCO, 2006, p.280)

Logo, a utilização de funções polinomiais é uma alternativa para modelar o comportamento da série e fazer a previsão. TRAN e WARD (2016) também propõe a utilização de polinômios para recuperar as equações governantes. Contudo, trataremos essas funções com a utilização de métodos de regressão que consistem em ajustar as curvas que podem envolver tanto interpolação, onde é necessário um ajuste exato aos dados, quanto suavização, na qual é construída uma função de melhor ajuste que se aproxime dos dados.

Algumas referências apresentam regressão também como uma interpretação de redes neurais, como podemos ver em BUENO *et al.* (2016). De acordo com ELLACOTT *et al.* (2012), "A rede é treinada para resolver a soma dos quadrados das diferença entre dois valores, então a implementação desta rede neural é equivalente a usar o método dos mínimos quadrados para ajustar o modelo de regressão".

Seguindo esse raciocínio, neste trabalho iremos apresentar métodos que podem ser implementados em quaisquer linguagem de programação com o objetivo geral de:

- propor algoritmos numéricos capazes de recuperar a dinâmica do sistema e apresentar soluções aproximadas a partir dos dados observados,
- realizar previsão dos sistemas a quaisquer condições iniciais.

Contudo, em principio, iremos objetivar especificamente:

- implementar um método matemático capaz de resolver numericamente, sobre condições iniciais o sistema de Lorenz,
- obter algoritmos capazes de encontrar parâmetros para funções polinomiais que se aproximem dos dados da resolução numérica e recupere a dinâmica do sistema de Lorenz,
- analisar o comportamento dos sistemas na presença de ruídos observacionais,
- analisar o comportamento dos sistemas com base na reconstrução do sistema,
- realizar previsão do sistemas de Lorenz.

Dessa maneira, para efeito de organização, além da introdução, esta dissertação está distribuída em 4 capítulos. No capítulo 2, fazemos uma revisão bibliográfica sobre alguns dos conceitos que compõe o estudo de sistemas dinâmicos, enfatizando o uso de sistemas não lineares para previsão. No capítulo 3, apresentamos os métodos que propomos aplicar para a recuperação da dinâmica e previsão de sistemas dinâmicos. No capítulo 4 apresentaremos e discutimos os testes e resultados da metodologia aplicada ao sistema de Lorenz. Tais resultados foram obtidos com a

implementação dos métodos em linguagem de programação Python. Por fim, no capítulo 5, apresentamos uma conclusão sobre todos os resultados obtidos nesta pesquisa.

Capítulo 2

Conceitos Gerais

Os conteúdos desse capitulo apresentam definições e teorias interessantes para nosso trabalho de previsão.

2.1 Sistemas Dinâmicos

Sistema dinâmico é um conceito matemático representado por equações diferenciais de primeira ordem cujas soluções evoluem temporalmente. Formalmente, temos:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), t). \tag{2.1}$$

$$\mathbf{x} \in \mathbb{R}^m \quad , \quad t \in \mathbb{R} \quad e \quad \mathbf{f} : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^m.$$

É importante salientar que um sistema dinâmico (matemático) geralmente representa um modelo de um sistema real. Essas equações diferencias representam os fluxos contínuos no tempo e no espaço, onde:

- t é variação temporal,
- $\mathbf{x} = (x_1, x_2, ..., x_m)^T$ são as variáveis de posição, também definidas de variáveis de estado,
- $\dot{\mathbf{x}} = (\dot{x}_1, \dot{x}_2, ..., \dot{x}_m)^T$ constituem as evoluções temporais dos estados, definidas como velocidades,
- $\mathbf{f} = (f_1, f_2, ..., f_m)^T$ é o vetor que representa o campo de direções, também definidas de variação espacial.

Note que para simplificar a notação de derivada temporal estamos utilizando $\dot{\mathbf{x}}$ no lugar de $\dfrac{d\mathbf{x}}{dt}$.

Determinar a evolução temporal das variáveis dinâmicas do sistema pode ser útil para realizar simulações com o propósito de evitar prejuízos ou tragédias naturais, por exemplo, no estudo de um projeto de satélite artificial e o sistema de pilotagem automática de aviões.

Exemplo 2.1. (STROGATZ, 1994, p.155, tradução nossa) Seguindo o clássico modelo de competição de duas especies de Lotka-Volterra, ilustraremos o caso entre coelhos e ovelhas. Suponhamos que ambas as especies estão competindo pelo mesmo tipo de alimento (grama) e a quantidade disponível é limitada. Além disso, vamos ignorar todas as outras complicações, como predadores, efeitos sazonais e outras fontes de alimento. Então há dois efeitos mais que devemos considerar:

1. Cada espécie cresceria até sua capacidade de carga na ausência do outro. Isto pode ser modelado assumindo crescimento logístico para cada espécie. Os coelhos têm uma habilidade fantástica de se reproduzir, então talvez devêssemos atribuir-lhes uma taxa de crescimento intrínseca mais alta.

2. Quando coelhos e ovelhas se encontram, o problema começa. Às vezes, a ovelha toma posse do local de comida e impede o coelho de comer. Vamos supor que esses conflitos ocorram a uma taxa proporcional ao tamanho de cada população. (Se houvesse o dobro de ovelhas, as chances de um coelho encontrar uma ovelha seriam duas vezes maiores). Além disso, assumimos que os conflitos reduzem a taxa de crescimento de cada espécie, mas o efeito é mais severo para os coelhos.

Um exemplo de um modelo de sistema dinâmico específico que incorpora essas suposições é:

$$\begin{aligned} \dot{x}_1 &= x_1(3 - x_1 - 2x_2), \\ \dot{x}_2 &= x_2(2 - x_1 - x_2), \end{aligned} \tag{2.2}$$

onde $x_1(t)$ = população de coelhos, $x_2(t)$ = população de ovelhas.

Relação entre parâmetros e variáveis

Os sistemas possuem características de dependência de variáveis e de parâmetros. Mostraremos essas características através do modelo de Verhulst[1]. Dado o sistema:

$$\dot{x} = rx\left(1 - \frac{x}{K}\right), \tag{2.3}$$

[1] [...]first suggested to describe the growth of human populations by Verhulst in 1838. (STROGATZ, 1994, p.22)

tal que $x = x(t)$.

Determinar $x(t)$ significa encontrar uma função x que satisfaça à equação diferencial acima e também a uma condição inicial especificada $x(0) = x_0$, onde x é a variável dependente que representa o tamanho populacional. Além disso, t é uma variável que evolui livremente, isto é, independente. Já, r e K são a taxa de crescimento e capacidade de carga, respectivamente, as quais representam parâmetros que influenciam a equação. Neste modelo, não há dependência temporal em r e K e seus valores permanecem constantes.

Espaço de Fase

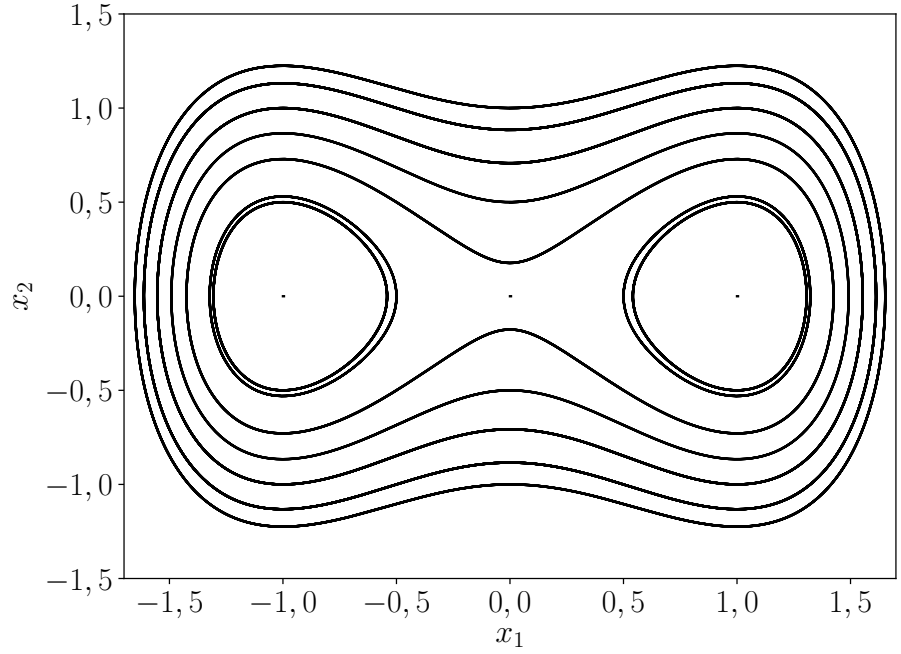
O espaço de fase de um sistema dinâmico é definido como o domínio das possíveis trajetórias e é uma ferramenta útil na compreensão do comportamento dos sistemas. Uma trajetória no espaço de fase representa a evolução temporal do sistema. Os sistemas dinâmicos mediante a condições iniciais, geram um conjunto de trajetórias possíveis chamado de variedade ou fluxo do sistema, formando o retrato de fase (Ver Figura 2.1(a)). Além disso, a imagem das derivadas (velocidades) forma o campo vetorial e também pode ser chamado de fluxo (Ver Figura 2.1(b)). Fluxo pode ter 3 sentidos diferentes: as equações diferenciais, a solução das equações diferenciais, e a velocidade (derivadas) da solução. Quando o sistema é não linear e, portanto, difícil de obter uma forma algébrica da solução, nesse caso podemos simular sua evolução temporal a partir de aproximações numéricas e representar essa evolução no espaço de fase.

Exemplo 2.2. Dado o sistema:

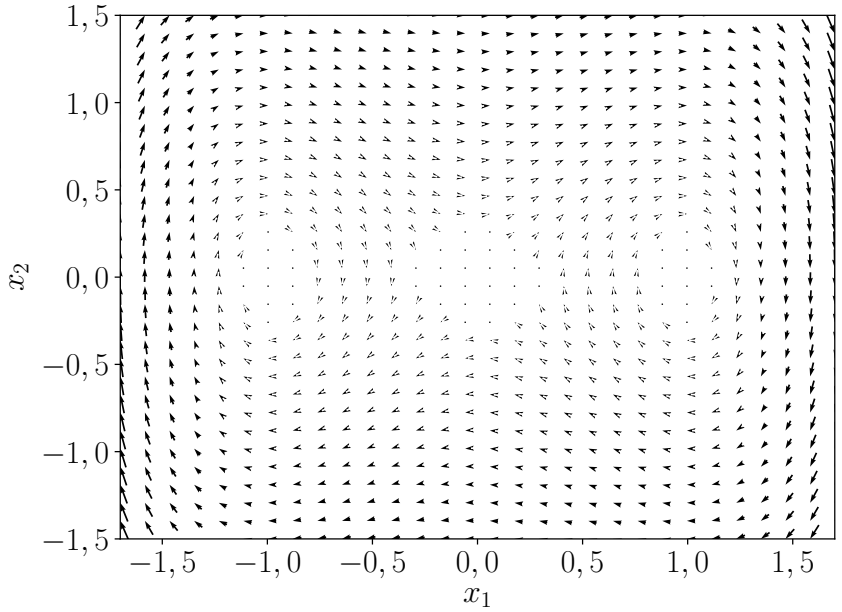
$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= x_1 - x_1^3. \end{aligned} \tag{2.4}$$

A Figura 2.1(a) mostra o retrato de fase do Sistema 2.4. Já a Figura 2.1(b) mostra o campo vetorial do mesmo.

Figura 2.1: Representação evolutiva no espaço de fase



(a) Retrato de Fase



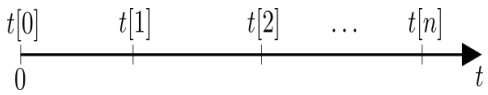
(b) Campo Vetorial

Fonte: Elaborada pelo autor.

Série Temporal

Considerando a utilização de ferramentas computacionais para a resolução de sistemas dinâmicos, é fundamental entender o conceito de tempo t discreto. Diferentemente do tempo contínuo, em que é descrito em todo o seu domínio pertencente a $T \subseteq \mathbb{R}^+$, o tempo discreto são valores definidos apenas em um conjunto sequencial de instantes de t, ou seja, o tempo é escrito através de uma função de índices $t[n], n \in \mathbb{Z}^+$, onde o tempo entre um termo e seu sucessor é considerado constante (Ver Figura 2.2). No entanto, o tempo discreto pode ter sido obtido através de um processo de amostragem de tempo contínuo.

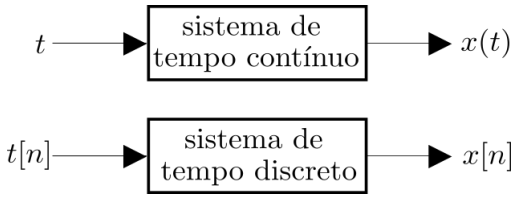
Figura 2.2: Tempo Discreto



Fonte: Elaborada pelo autor.

Tempo discreto e tempo contínuo são duas circunstancias alternativas para modelar variáveis que evoluam ao longo do tempo. Aplicar o tempo contínuo a um sistema resulta em saídas continuas e aplicar o tempo discreto a um sistema resulta em saídas discretas. Isto pode ser visto no esquema 2.3

Figura 2.3: Ilustração



Fonte: Elaborada pelo autor.

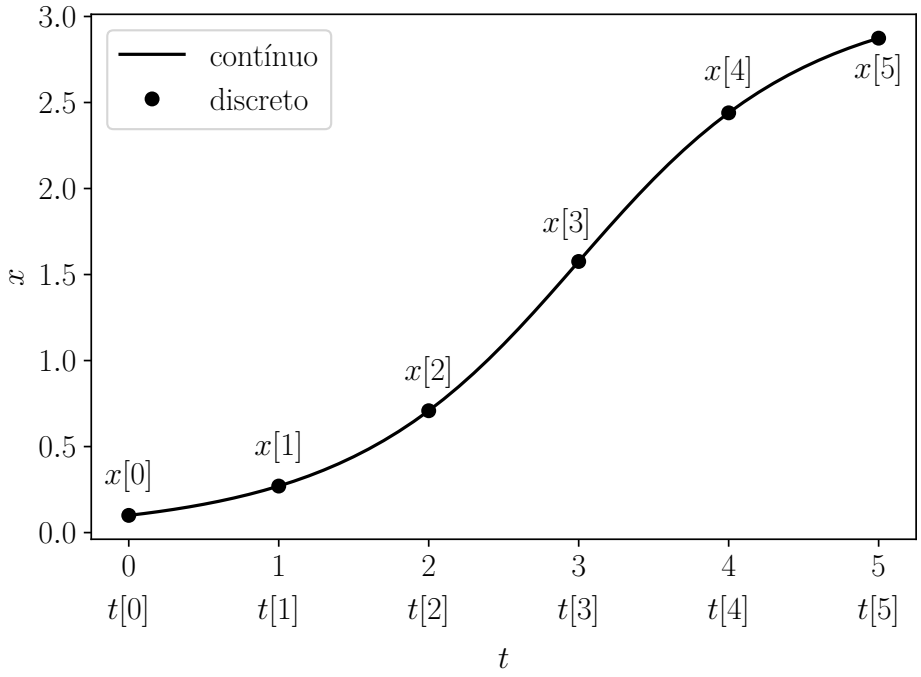
Uma série temporal é uma coleção de observações feitas sequencialmente ao longo do tempo, isto é, uma sequência de uma variável $x[n] = x(t[n])$.

> Dados de séries temporais surgem em vários campos do conhecimento como Economia (preços diários de ações, taxa mensal de desemprego, produção industrial), Medicina (eletrocardiograma, eletroencefalograma), Epidemiologia (número mensal de novos casos de meningite), Meteorologia (precipitação pluviométrica, temperatura diária, velocidade do vento), etc.(EHLERS, 2005, p.1)

Uma série temporal é dita ser contínua quando as observações são feitas para todo valor real no tempo ou discreta quando as observações são feitas em tempos

específicos que em geral são igualmente espaçados e associados a índices de números inteiros (Ver 2.4). Note que estes termos não se referem à variável observada, esta pode assumir valores discretos ou contínuos. Por outro lado, séries temporais discretas podem surgir de várias formas. Séries contínuas podem ser discretizadas, isto é, seus valores são registrados a certos intervalos de tempo. Séries de valores agregados ou acumulados em intervalos de tempo, por exemplo exportações medidas mensalmente ou quantidade de chuva medida diariamente. Finalmente, algumas séries são inerentemente discretas, por exemplo dividendos pagos por uma empresa aos seus acionistas em anos sucessivos.

Figura 2.4: Série Temporal



Fonte: Elaborada pelo autor.

2.2 Classificação dos Sistemas Dinâmicos

Na Teoria de Sistemas Dinâmicos tem-se a classificação para diferentes tipos de situações. Abordaremos alguns tipos que serão usados nesse trabalho.

2.2.1 Autonomia

Através da Equação 2.3, temos uma característica, f depende de $x(t)$, mas não depende de t explicitamente, isto é, não há parâmetros ou termos que se configuram para f ser do tipo $f(x,t)$. Sendo assim, define-se esse tipo de sistema como autô-

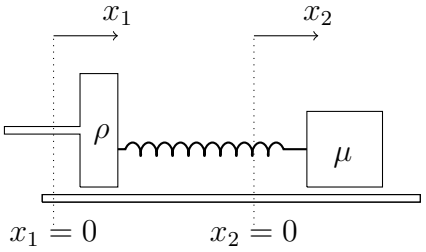
nomo. Uma vez que se apresentasse $f(x,t)$, então $\dfrac{\partial f}{\partial t} \neq 0$. Logo, o sistema seria não-autônomo.

Em geral, sistemas não-autônomos podem ser reescritos de forma autônoma aumentando sua dimensão. As vezes é preciso criar uma nova variável para cada termo não autônomo no sistema original, por exemplo, se temos uma equação $x_{m+1} = t$, então $\dot{x}_{m+1} = 1$. Logo, acrescenta ao sistema e torna-se:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t)). \tag{2.5}$$
$$\mathbf{x} \in \mathbb{R}^{m+1} \quad e \quad \mathbf{f} : \mathbb{R}^{m+1} \to \mathbb{R}^{m+1}.$$

Exemplo 2.3. (FITZPATRICK, 2013, tradução nossa) Considere uma versão modificada do sistema de massa-mola chamada de oscilação harmônica amortecida e forçada na qual uma extremidade da mola é presa à massa μ e a outra a um pistão móvel ρ (Veja 2.5).

Figura 2.5: Sistema oscilatório forçado



Fonte: Elaborada pelo autor.

Seja $x_1(t)$ o deslocamento horizontal da massa e $x_2(t)$ o deslocamento horizontal do pistão. A distensão da mola é $x_1(t) - x_2(t)$, assumindo que a mola não está esticada quando $x_1 = x_2 = 0$. Suponha que, à medida que a massa desliza sobre a superfície horizontal, esteja sujeita a uma força de amortecimento por atrito que se opõe ao seu movimento e que é diretamente proporcional à sua velocidade instantânea. Assim, a força horizontal que age sobre a massa pode ser escrita:

$$\begin{aligned} F &= -k(x_1 - x_2) - \mu v \dot{x}_1, \\ \mu \ddot{x}_1 &= -k(x_1 - x_2) - \mu v \dot{x}_1, \end{aligned}$$

onde $k > 0$ é a força constante da mola e $v > 0$ é a constante de amortecimento

Além disso, suponha que ρ se desloque em uma oscilação harmônica simples de frequência angular $\omega > 0$ e amplitude $\alpha > 0$, ou seja, $x_2 = \alpha \cos(\omega t)$. Assim, a equação se torna:

$$\ddot{x}_1 + v\dot{x}_1 = -\frac{k}{\mu}[x_1 + \alpha \cos(\omega t)].$$

Definindo $\omega_0^2 = \frac{k}{\mu}$, temos:

$$\ddot{x}_1 + v\dot{x}_1 + \omega_0^2 x_1 = \omega_0^2 \alpha \cos(\omega t). \tag{2.6}$$

Logo, 2.6 é uma equação diferencial ordinária de segunda ordem e pode ser reescrita como um sistema de equações diferenciais de primeira ordem. Definindo $x_2 = \dot{x_1}$:

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= \omega_0^2 \alpha \cos(\omega t) - (v x_2 + \omega_0^2 x_1). \end{aligned}$$

Assim, temos um sistema dinâmico não-autônomo. Definindo $x_3 = \omega t$, temos um novo sistema autônomo:

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= \omega_0^2 \alpha \cos(x_3) - (v x_2 + \omega_0^2 x_1), \\ \dot{x}_3 &= \omega. \end{aligned}$$

2.2.2 Linearidade

Um sistema também pode ser classificado como linear ou não-linear. Um sistema linear possui as seguintes propriedades:

$$\mathbf{f}(\mathbf{x} + \mathbf{y}) = \mathbf{f}(\mathbf{x}) + \mathbf{f}(\mathbf{y}),$$

$$\mathbf{f}(\alpha \mathbf{x}) = \alpha \mathbf{f}(\mathbf{x}),$$

em que $\alpha \in \mathbb{R}$.

Essas duas propriedades podem ser combinadas em uma só, chamada de principio da superposição:

$$\mathbf{f}(\alpha_1 \mathbf{x} + \alpha_2 \mathbf{y}) = \alpha_1 \mathbf{f}(\mathbf{x}) + \alpha_2 \mathbf{f}(\mathbf{y}).$$

Se o sistema é linear, então a função do segundo membro da equação 2.5 possui a seguinte configuração:

$$\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}, \tag{2.7}$$

onde $\mathbf{A} \in \mathbb{R}^{m+1} \times \mathbb{R}^{m+1}$ é a matriz de coeficientes constantes.

Se for linear, há soluções gerais que nos permitem determinar o comportamento futuro do sistema descrito de forma analítica, em função do estado atual do sistema.

Exemplo 2.4. O modelo de movimento harmônico simples de um sistema massa-mola é:

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -\frac{k}{\mu}x_1, \end{aligned} \tag{2.8}$$

onde k é força constante da mola e μ é massa de um objeto qualquer. Definindo $\omega^2 = \frac{k}{\mu}$, o sistema 2.8 tem solução geral analítica dada por:

$$\begin{aligned} x_1(t) &= e^{\omega t} + e^{-\omega t}, \\ x_2(t) &= \omega e^{\omega t} - \omega e^{-\omega t}, \end{aligned}$$

e é linear. De fato, seja $\boldsymbol{\alpha} = [\alpha_1, \alpha_2]^T$ um vetor constante e $\mathbf{y} = [y_1, y_2]^T$ um vetor qualquer de variáveis de estado, então:

$$\begin{aligned} \mathbf{f}(\alpha_1 x_1 + \alpha_2 y_1, \alpha_1 x_2 + \alpha_2 y_2) &= [\alpha_1 x_2 + \alpha_2 y_2, -\omega^2(\alpha_1 x_1 + \alpha_2 y_1)]^T \\ &= \alpha_1[x_2, -\omega^2 x_1]^T + \alpha_2[y_2, -\omega^2 y_1)]^T \\ &= \alpha_1 \mathbf{f}(\mathbf{x}) + \alpha_2 \mathbf{f}(\mathbf{y}). \end{aligned}$$

De outra maneira,

$$\begin{aligned} \frac{d}{dt}(\alpha_1 x_1 + \alpha_2 y_1) &= \quad \alpha_1 x_2 + \alpha_2 y_2 \quad &= \alpha_1 \frac{d}{dt}(x_1) + \alpha_2 \frac{d}{dt}(y_1), \\ \frac{d}{dt}(\alpha_1 x_2 + \alpha_2 y_2) &= -\tfrac{k}{\mu}(\alpha_1 x_1 + \alpha_2 y_1) &= \alpha_1 \frac{d}{dt}(x_2) + \alpha_2 \frac{d}{dt}(y_2) \end{aligned}$$

Já se o sistema for não-linear, essas soluções exatas, em geral, não existem analiticamente, mas o comportamento futuro a pequenos tempos, normalmente, pode ser obtido por solução numérica (computacional) das equações de evolução. Alguns sistemas bastante complicados podem aparecer quando estudamos a evolução temporal de sistemas descritos por equações não-lineares: órbitas periódicas e quase-periódicas, ciclos-limite, transientes, e caos.

Exemplo 2.5. O modelo de Vershulst 2.3 é um exemplo de sistema não-linear, De fato, seja $\mathbf{P} = [P_1, P_2]^T$ um vetor qualquer de variáveis de estado, então:

$$\begin{aligned}\frac{d}{dt}(P_1+P_2) &= r(P_1+P_2)\left(1-\frac{P_1+P_2}{K}\right)\\ &= rP_1\left(1-\frac{P_1+P_2}{K}\right)+rP_2\left(1-\frac{P_1+P_2}{K}\right)\\ &\neq \frac{d}{dt}(P_1)+\frac{d}{dt}(P_2),\end{aligned}$$

isto é, falha na propriedade de aditividade.

2.3 Noções de Estabilidade

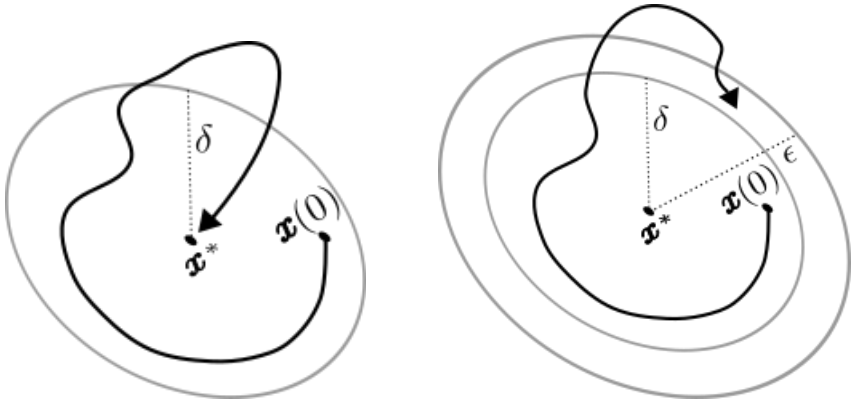
Primeiramente, define-se um ponto de equilíbrio ou *ponto fixo* de um sistema dinâmico como sendo o ponto em que o sistema permanece estacionário na medida em que o tempo evolui. Dado o sistema 2.1, um ponto $\mathbf{x}^* \in \mathbb{R}^m$ é dito ponto fixo do sistema se $\mathbf{f}(\mathbf{x}^*, t) = \mathbf{0}$.

Os pontos fixos possuem uma característica chamada *estabilidade*. O conceito de estabilidade é fundamental no estudo de sistemas dinâmicos e está associada a resposta de uma determinada perturbação. Se essa perturbação não afetar significativamente uma dada solução, então ela é *estável*. Do contrário, ela é *instável*.

Quando a pertubação leva a um estado estável, já se tem a previsão com facilidade. Em contrapartida, quando a pertubação leva a um estado instável, então torna-se difícil a previsão. Modelos que apresentam essa característica de dificuldade, torna-se interessante a ser trabalhado.

> Considere um ponto fixo $\mathbf{x}^*$ do Sistema 2.1. Dizemos que $\mathbf{x}^*$ é atratitvo se existe um $\delta > 0$ tal que $\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}^*$ sempre que $\|\mathbf{x}(0) - \mathbf{x}^*\| < \delta$. Em outras palavras, qualquer trajetória que inicia dentro de um raio de distancia δ de $\mathbf{x}^*$ se aproxima de $\mathbf{x}^*$ em algum tempo eventual e se mantém próximo para todo o tempo positivo. Como mostra o esquema na Figura 2.6, trajetórias fogem de $\mathbf{x}^*$ em um período de tempo, mas se aproximam cada vez mais em um tempo eventual. O conjunto de todas as condições iniciais que convergem para um mesmo ponto atrativo formam a bacia de atração.
>
> Em outro contraste, dizemos que $\mathbf{x}^*$ é Liapunov estável ou apenas neutralmente estável se dado $\epsilon > 0$, existe $\delta > 0$ tal que para $\|\mathbf{x}(0) - \mathbf{x}^*\| < \delta$, então $\|\mathbf{x}(t) - \mathbf{x}^*\| < \epsilon$, para todo $t \geq 0$. Logo, trajetórias que iniciam dentro de um raio de distancia δ de $\mathbf{x}^*$ nunca se afastam mais do que um raio de distancia ϵ de $\mathbf{x}^*$ para todo tempo positivo, ou seja, as trajetórias permanecem confinadas.
>
> Dizemos que $\mathbf{x}^*$ é estável se é atrativo e Liapunov estável.
>
> Finalmente, $\mathbf{x}^*$ é instável se nem é atrativo e nem é Liapunov estável. (STROGATZ, 1994, p.128, tradução nossa).

Figura 2.6: Estabilidade

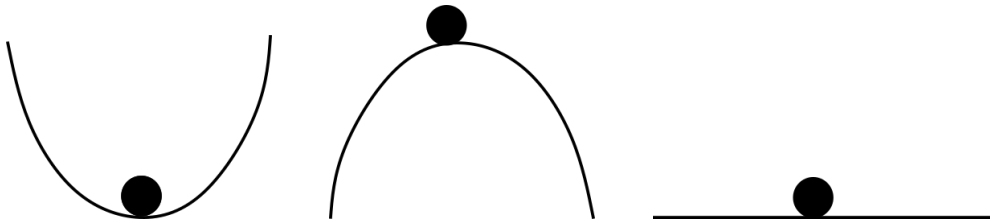


(a) Atrator (b) Liapunov Estável

Fonte: Elaborada pelo autor.

De maneira física, podemos ver na Figura 2.7 o conceito de estabilidade por um exemplo do movimento de um objeto em diferentes situações.

Figura 2.7: Objeto rolando em diferentes superfícies



(a) estável (b) instável (c) neutralmente estável

Fonte: Strogatz, 2014.

(a) Após uma perturbação, o objeto retorna à posição inicial

(b) Após uma perturbação, o objeto abandona à posição inicial, indo para uma posição as vezes desconhecida

(c) Após uma perturbação, o objeto tende a permanecer em uma nova posição

Exemplo 2.6. (STROGATZ, 1994, p.126, tradução nossa) Dado o sistema,

$$\begin{aligned} \dot{x}_1 &= ax_1, \\ \dot{x}_2 &= -x_2. \end{aligned}$$

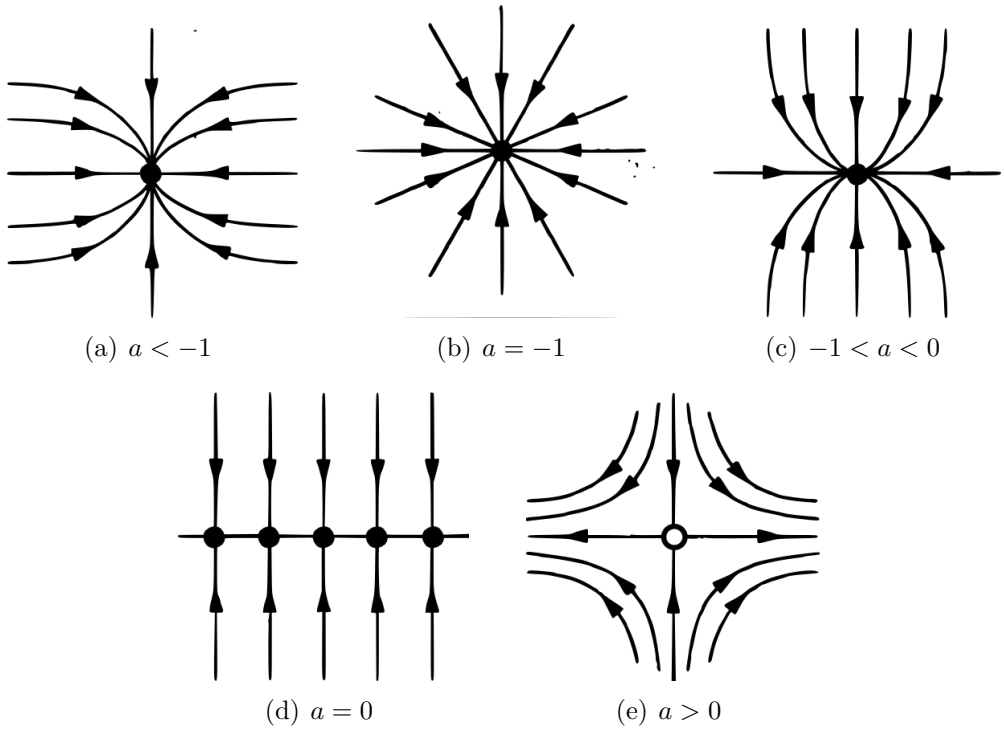
Trace o retrato de fase quando a varia entre $-\infty$ para $+\infty$, mostrando qualitativamente diferentes casos.

Solução: Nesse simples sistema, a solução é

$$\begin{aligned} x_1(t) &= x_1(0)e^{at}, \\ x_2(t) &= x_2(0)e^{-t}. \end{aligned}$$

O retrato de fase para diferentes valores de a são mostrados na Figura 2.8. Em cada caso, $x_2(t)$ decai exponencialmente. Quando $a < 0$, $x_1(t)$ também decai exponencialmente e assim todas as trajetórias tendem a origem quando $t \to \infty$. Contudo, as direções abordadas dependem do tamanho de a comparado com -1

Figura 2.8: Retrato de fase com diferentes valores de a



(a) $a < -1$ (b) $a = -1$ (c) $-1 < a < 0$

(d) $a = 0$ (e) $a > 0$

Fonte: Strogatz, 2014.

Na figura 2.8(a-c), a origem é estável. No entanto, 2.8(d) não é atraente, mas é Liapunov estável. Já a figura 2.8(e) é instável, pois nem é atraente e nem é Liapunov estável.

Esses conceitos levam a algumas das principais características de qualquer retrato de fase em sua análise qualitativa.

Exemplo 2.7. (STROGATZ, 1994, p.146, tradução nossa)

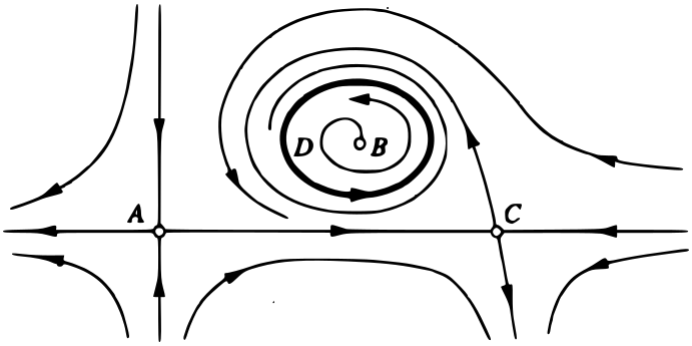
1. Os pontos fixos A, B e C na figura satisfazem $\mathbf{f}(\mathbf{x}^*) = \mathbf{0}$, e correspondem a estados estacionários ou equilíbrio do sistema.

2. As orbitas fechadas tal como D na figura 2.9 corresponde a soluções periódicas, isto é, soluções para o qual $\mathbf{x}(t+T) = \mathbf{x}(t)$ para todo t e para algum $T > 0$,

3. A organização das trajetórias ou padrão de fluxo perto de A e C é similar, mas diferente quando perto de B

4. Os pontos fixos A, B e C são instáveis, pois as trajetórias tendem a se afastarem quando estão próximas desses pontos, enquanto que a orbita fechada D é estável, isto é, as trajetórias tendem a se aproximar.

Figura 2.9: Comportamento qualitativo das soluções



Fonte: Strogatz, 2014.

2.4 Dinâmica Caótica

2.4.1 Caos

O *caos* tem sido definido como o comportamento aparentemente *estocástico* de sistemas determinísticos. Sistemas dinâmicos não-lineares são passíveis de apresentar um comportamento caótico desde que algumas condições sejam satisfeitas. O caos é um comportamento muito rico, onde o sistema experimenta uma dinâmica com diversas possibilidades de respostas.

> Caos é um comportamento aperiódico de longo prazo em um sistema determinístico que exibe dependência sensível a condições iniciais.
>
> 1. 'Aperiódico de longo prazo' significa que existem trajetórias que não se fixam em pontos fixos, órbitas periódicas ou orbitas quase-periódicas quando $t \to \infty$. Por razões praticas, podemos ter certeza que essas trajetórias não são raras. Por instância, podemos dizer que há um conjunto aberto de condições inicias que levam a trajetórias aperiódicas, ou talvez, dada uma condição inicial aleatória, obtenha trajetórias com probabilidade diferente de zero.

2. 'Determinístico' significa que os sistemas não são aleatórios ou tenham entradas ruidosas. O comportamento irregular surge pela não linearidade do sistema, mas não pelo ruido.
3. 'Dependência sensível a condições iniciais' significa que trajetórias bem próximas se afastam rapidamente a uma taxa exponencial, isto é, o sistema possui um expoente de Liapunov positivo.

(STROGATZ, 1994, p.323, tradução nossa)

Sendo assim, as características essenciais do comportamento caótico são a imprevisibilidade a partir de um longo prazo e a sensibilidade às condições iniciais. A previsibilidade até um tempo eventual do comportamento de sistemas caóticos é de grande importância, porque muitos fenômenos do mundo real apresentam algum tipo de comportamento complexo, e alguma lei determinística para evolução de seu estado, fazendo com que seja similar ao comportamento caótico. Várias áreas do conhecimento tem interesse no estudo da predição em séries temporais, dentre elas, em engenharia elétrica CAMPOS (2008), clima COSTA *et al.* (2015), finanças CORDEIRO JR. (2007), entre outros. Muitos trabalhos tem sido desenvolvidos com o objetivo de conseguir realizar ou aperfeiçoar métodos de previsão em sistemas caóticos, dentre eles,YANG *et al.* (2012), LU *et al.* (2018), FONTES (2013), PATHAK *et al.* (2018), PECORA *et al.* (2007), WANG *et al.* (2016), DE S. CAVALCANTE *et al.* (2013).

2.4.2 Atrator

De acordo com STROGATZ (1994, p.324, tradução nossa),

> Definimos um atrator como um conjunto fechado A com as seguintes propriedades:
>
> 1. A é um conjunto invariante: qualquer trajetória $x(t)$ que começa em A permanece em A por todo o tempo.
> 2. A atrai um conjunto aberto de condições iniciais: se um conjunto aberto U contem A e se $x(0) \in U$, então a distância entre $x(t)$ e A tende a zero quando $t \to \infty$. Logo, A atrai todas as trajetórias que começam suficientemente perto dele. O maior conjunto U é chamado de bacia de atração de A.
> 3. A é mínimo: não há subconjunto adequado de A que satisfaz as condições 1 e 2.

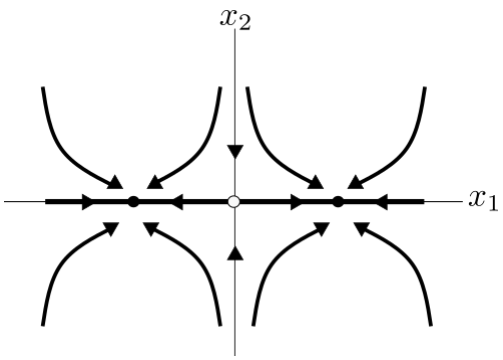
Exemplo 2.8. (STROGATZ, 1994, p.325, tradução nossa) Considere o sistema

$$\begin{aligned} \dot{x}_1 &= x_1 - x_1^3, \\ \dot{x}_2 &= -x_2. \end{aligned}$$

Seja $-1 \leq x_1 \leq 1$ e $x_2 = 0$ tal que formam um conjunto I.

O retrato de fase 2.10 mostra que existem pontos fixos estáveis nos pontos finais $(\pm 1, 0)$ de I, um ponto de sela na origem e que o conjunto I é invariante, pois qualquer trajetória que comece em I permanece em I o tempo todo. Na verdade, todo o eixo x_1 é um conjunto invariante, pois se $x_2(0) = 0$, então $x_2(t) = 0$ para todo tempo t. Logo, a condição (1) é satisfeita.

Figura 2.10: Ilustração qualitativa de um conjunto que não é atrator.



Fonte: Elaborada pelo autor.

Além disso, I atrai um conjunto aberto de condições inicias, visto que atrai todas as trajetórias no plano x_1x_2. Então a condição 2 também é satisfeita.

Entretanto I não é mínimo, pois o subconjunto formado pelos pontos fixo $(\pm 1, 0)$ satisfaz as condições 1 e 2. Logo, I não é *atrator*.

Esse exemplo induz a mostrar um conjunto que não é atrator, mas claro, se considerarmos o conjunto formado apenas pelo ponto $(1, 0)$ ou apenas pelo ponto $(-1, 0)$, então temos dois atratores, pois obedecem todas as condições da definição.

"Atratores são chamados de estranhos se tiverem uma estrutura fractal" BOEING (2016). Geralmente, este é o caso quando a dinâmica nele é caótica, mas também existem atratores estranhos nos quais não há caos. Um exemplo de sistema caótico que possui um atrator estranho é o sistema de Lorenz. Em 1963, o meteorologista E. N. Lorenz introduziu um modelo meteorológico descrito por três equações diferenciais:

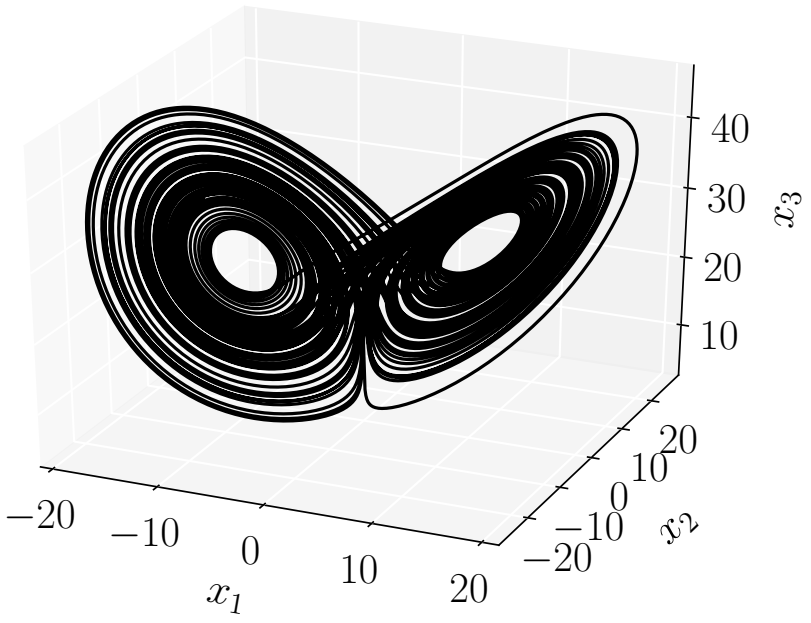
$$\begin{aligned} \dot{x}_1 &= \sigma(x_2 - x_1), \\ \dot{x}_2 &= rx_1 - x_2 - x_1x_3, \\ \dot{x}_3 &= x_1x_2 - bx_3, \end{aligned} \tag{2.9}$$

onde x_1 representa a amplitude das correntes de convecção, x_2 é a diferença de temperaturas entre as correntes ascendente e descendente e x_3 representa o desvio da temperatura normal no plano. Os três parâmetros σ, r e b são positivos e dependem das propriedades físicas do fluxo de ar.

> Lorenz descobriu que esse sistema determinístico, de aparência simples, poderia ter dinâmicas extremamente erráticas: soluções oscilam irregularmente, nunca exatamente repetindo, mas permanecendo sempre em uma região delimitada do espaço de fase. Quando ele traçou as trajetórias em três dimensões, descobriu que eles se instalaram em conjunto estranho. Diferentemente dos pontos fixos e dos ciclos limites, o atrator estranho não é um ponto, uma curva ou mesmo uma superfície - é um fractal, com uma dimensão fracionária entre 2 e 3.(STROGATZ, 1994, p.301, tradução nossa)[2]

No sistema 2.9, utilizando os parâmetros $\sigma = 10$, $r = 28$ e $b = 2.667$, suas soluções numéricas levam a um atrator. Neste trabalho, sempre que nos referirmos ao Sistema de Lorenz, estamos considerando estes valores de parâmetros.

Figura 2.11: Atrator de Lorenz



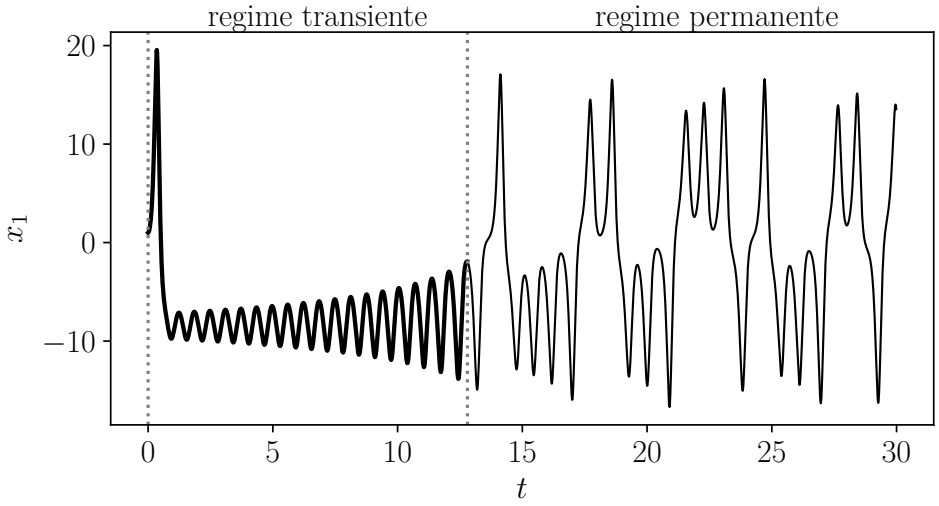
Fonte: Elaborada pelo autor.

Um detalhe interessante é a fase da evolução temporal que antecede a chegada até o atrator chamada de transiente. Quando se tratar de previsão, dada qualquer condição inicial pertencente ao transiente, então até um certo tempo, a solução convergirá para o atrator. Além disso, sua inclusão pode levar a um viés desnecessário

[2]Lorenz discovered that this simple-looking deterministic system could have ex tremely erratic dynamics: over a wide range of parameters, the solutions oscillate irregularly, never exactly repeating but always remaining in a bounded region of phase space. When he plotted the trajectories in three dimensions, he discovered that they settled onto a complicated set, now called a strange attractor. Unlike stable fixed points and limit cycles, the strange attractor is not a point or a curve or even a surface—it's a fractal, with a fractional dimension between 2 and 3.

nas simulações de estimativas de recuperação do sistema. “Após um transiente inicial, a solução se instala em uma oscilação irregular que persiste quando $t \to \infty$, mas nunca se repete de forma exata. O movimento é aperiódico”.(STROGATZ, 1994, p.318, tradução nossa)[3]. A Figura 2.12, mostra um exemplo da evolução temporal de uma variável do sistema de Lorenz passando de um regime transiente para um regime permanente (oscilações aperiódicas). Podemos ver que ao entrar em regime permanente, a evolução temporal se comporta de maneira irregular em relação ao tempo. A partir de um transiente inicial, a solução começa a oscilar alternando em torno de um valor positivo e um negativo.

Figura 2.12: Regime transiente e permanente



Fonte: Elaborada pelo autor.

2.4.3 Expoente de Liapunov

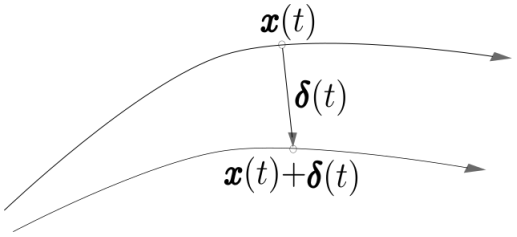
O movimento no atrator exibe uma dependência sensível das condições iniciais. Isso significa que duas trajetórias que começam muito próximas tendem a divergir rapidamente entre si e, a partir daí, terão futuros totalmente diferentes. Suponhamos que $\mathbf{x}(t)$ é um ponto de atração e considere um ponto próximo, digamos $\bar{\boldsymbol{x}} = \mathbf{x}(t) + \boldsymbol{\delta}(t)$, onde $\boldsymbol{\delta}(t)$ é uma variação da distancia entre os pontos, isto é, a diferença entre os pontos. (Ver Figura 2.13).

Agora, vejamos como $\boldsymbol{\delta}(t)$ cresce. Assume-se que, o raio $\boldsymbol{\delta}(t)$ tenha variado exponencialmente ao longo do tempo, de maneira que a relação entre $\boldsymbol{\delta}(0)$ e o valor correspondente no instante t, dado por $\boldsymbol{\delta}(t)$, seja:

$$\boldsymbol{\delta}(t) \sim \boldsymbol{\delta}(0)e^{\lambda t}, \tag{2.10}$$

[3]After an initial transient, the solution settles into an irregular oscillation that persists as $t \to \infty$, but never repeats exactly. The motion is aperiodic.

Figura 2.13: Divergência de trajetórias



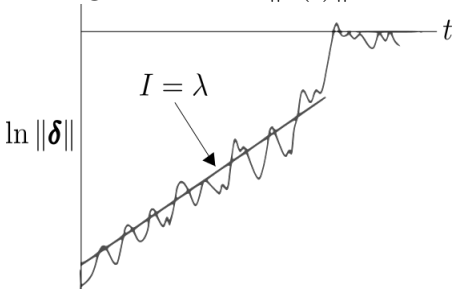
Fonte: Elaborada pelo autor.

onde λ é chamado de *expoente de Liapunov*.

> O número λ é freqüentemente chamado de expoente de Liapunov, embora esse seja um uso grosseiro do termo, por duas razões.
>
> Primeiro, na verdade, existem n expoentes de Liapunov diferentes para um sistema n-dimensional, definidos a seguir. Considere a evolução de uma esfera infinitesimal de condições iniciais perturbadas. Durante sua evolução, a esfera será distorcida em um elipsoide infinitesimal. Seja $\delta_k(t)$, $k = 1, \ldots, n$, tal qual denota o comprimento do k-ésimo eixo principal do elipsoide. Então, $\delta_k(t) \sim \delta_k(0)e^{\lambda_k t}$ ", onde os λ_k são os expoentes de Liapunov. Para t grande, o diâmetro do elipsoide é controlado pelo maior λ_k positivo. Assim, nosso λ é na verdade, o maior expoente de Liapunov.
>
> Segundo, λ depende (levemente) de qual trajetória estudamos. Devemos calcular a média de muitos pontos diferentes na mesma trajetória para obter o verdadeiro valor de λ_k.(STROGATZ, 1994, p.322, tradução nossa).

Os expoentes de Liapunov medem o quão previsível ou imprevisível é o sistema dinâmico. "Em estudos numéricos do atrator de Lorenz, descobre-se que $\lambda \approx 0.9$. Assim, as trajetórias vizinhas separam-se exponencialmente rápido. Equivalentemente, se traçarmos $\ln \|\delta(t)\|$ por t, encontramos uma curva que que se aproxima de uma reta com uma inclinação positiva de λ".(STROGATZ, 1994, p.321, tradução nossa). (Ver 2.14)

Figura 2.14: $\ln \|\delta(t)\| \times t$

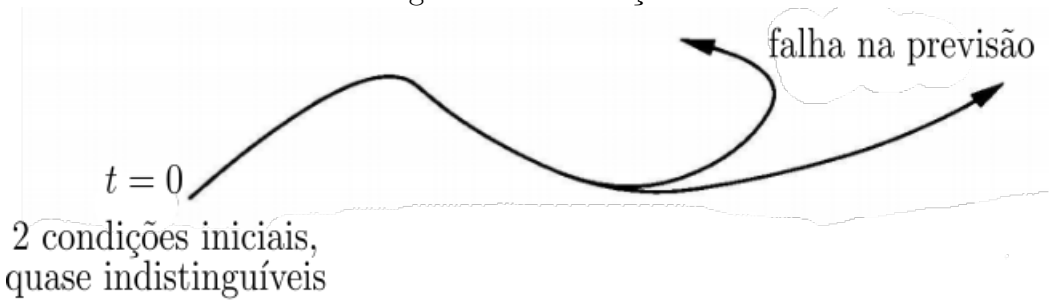


Fonte: Strogatz, 1994.

Em outras palavras, os expoentes de Liapunov expressam a perda de informação no tempo, ou seja, existe um horizonte de tempo além do qual a previsão se divide,

como mostrado esquematicamente na figura 2.15. Temos que a previsão é possível em um tempo curto, mas imprevisível em um certo tempo longo.

Figura 2.15: Predição



Fonte: Elaborada pelo autor.

Portanto, a análise do crescimento exponencial será indispensável para indicar o tempo de previsão de um experimento de recuperação de um sistema dinâmico caótico. Sendo assim, seja $\overline{x}$ o valor aproximado e x o valor exato. O erro de x, define-se como $\Delta x = x - \overline{x}$. Define-se ainda erro absoluto de x, como o valor absoluto $E_{abs} = |x - \overline{x}|$. A importância de um erro pode, em geral, ser melhor apreciada se o compararmos à quantidade a ser aproximada, ou seja, utilizando o erro relativo definido como $E_r = \frac{E_{abs}}{x}$. Por sua vez, o erro relativo de x está relacionado com o erro percentual definido como $E_{\%} = 100|E_r|$. Então, faremos nossa análise do tempo de previsão a partir do estudo da evolução do $E_{\%}$ pelo tempo t.

Capítulo 3

Metodologia

Desenvolveremos técnicas que construam modelos de sistemas dinâmicos a partir de informações de estado atual obtidas por uma série temporal. Essas técnicas objetivam determinar, através do modelo construído, uma estimativa dos valores de estado futuro, sem que se conheçam as reais equações diferenciais que governam a dinâmica do sistema.

Para se chegar a esse objetivo, trabalharemos com sistemas conhecidos, pois servirão para analisar a eficácia do método, dessa forma, os dados da série prevista podem ser comparados com os da série original no passado e no futuro. Permitindo a avaliação da precisão da técnica de previsão.

Escolheremos o sistema de Lorenz como sendo o nosso sistema conhecido. Resolveremos numericamente utilizando o método de Runge-Kutta de quarta ordem. Essa solução numérica nos dará as séries temporais do sistema que servirão para a estimativa de recuperação do sistema através da aproximação por funções polinomiais.

> Modelos lineares globais podem ser considerados como a primeira aproximação da função, isto é, em uma expansão de série de Taylor em torno das observações atuais. A generalização óbvia é usar um polinômio.[...]Polinômios têm a vantagem que nós estamos muito familiarizados com eles, o que nos dá alguma esperança de que podemos entender o resultado.(KANTZ e SCHREIBER, 2004, p., tradução nossa).

O uso de funções polinomiais servirão como aproximações para as variáveis de posição, de velocidade e das variáveis espaciais do sistema. Para isso, em alguns casos buscaremos encontrar coeficientes dos polinômios que façam com que a aproximação se adéque ao comportamento da série e/ou recupere a dinâmica do sistema. Algumas vezes precisaremos usar um método para ajustar as aproximações, esse método é chamado de método dos Mínimos Quadrados.

No caso de aproximações do comportamento da série, poderemos usar métodos que não buscam necessariamente os coeficientes dos polinômios, mas que se ajustam

aos dados e possuem a propriedade de continuidade e de diferenciação, por exemplo, Spline.

Contudo, o que mais nos interessa para esse trabalho são os coeficientes dos polinômios que se empregam nas funções polinomiais espaciais, pois essas funções serão as estimativas da dinâmica do sistema. Tendo o sistema recuperado, então poderemos fazer o uso da solução numérica para estimar a previsão.

Portanto, nesse capítulo apresentaremos os métodos que serão usados para o nosso objetivo de previsão. Primeiro apresentaremos definições e métodos de aproximação polinomial. Depois, definiremos ruído observacional que será incluso na análise de eficácia e obtenção de resultados do método de previsão. Em seguida, faremos uma abordagem do método de coordenadas atrasadas com intenção de solucionar a situação de possível insuficiência das informações disponíveis. Finalmente, apresentaremos o passo a passo do nosso método de previsão em sistemas dinâmicos.

3.1 Método de Aproximação Polinomial

A aproximação de funções por polinômios é uma das ideias mais antigas da analise numérica, e ainda uma das mais usadas. Os polinômios são facilmente computáveis, suas derivadas e integrais são novamente polinômios, suas raízes podem ser encontradas com relativa facilidade, etc.

Logo, utilizaremos funções polinomiais para modelar o comportamento da série temporal e fazer a recuperação do sistema.

Definição 3.1. (Polinômio Interpolador unidimensional)

Dado um conjunto de $n+1$ pontos, ou pares ordenados, reais, distintos, seja $(u[0], v[0]), (u[1], v[1]), (u[2], v[2]), \ldots, (u[n], v[n])$, determina-se uma função polinomial $P(u)$ de grau menor ou igual a G, de modo que

$$P(u[0]) = v[0] \ ; \ P(u[1]) = v[1] \ ; \ \ldots \ ; P(u[n]) = v[n], \tag{3.1}$$

onde segundo RUGGIERO e DA ROCHA LOPES, $\max G = n$

Definição 3.2. (Polinômio Interpolador multidimensional)

Dados $\mathbf{u} \in \mathbb{R}^m$ e para cada tem-se um conjunto de entrada de $n+1$ pontos reais e distintos, e um conjunto de saída de $n+1$ pontos reais e distintos. Sejam eles $(\mathbf{u}[0], v[0]), (\mathbf{u}[1], v[1]), (\mathbf{u}[2], v[2]), \ldots, (\mathbf{u}[n], v[n])$, determina-se uma função polinomial de varias variáveis[1] $F(\mathbf{u})$ de grau total menor ou igual a G de modo que

$$F(\mathbf{u}[0]) = v[0] \ ; \ F(\mathbf{u}[1]) = v[1] \ ; \ \ldots \ ; F(\mathbf{u}[n]) = v[n]. \tag{3.2}$$

[1] A definição de polinômio multidimensional também pode ser vista em WADE (2013).

Essas definições nos levam a buscar vários métodos com o propósito de fazer interpolação polinomial. Lembrando uma função polinomial de grau G unidimensional e/ou multidimensional, respectivamente, é dada por

$$P(u) = C[0] + C[1]u + C[2]u^2 + \cdots + C[n]u^G,$$
$$F(\mathbf{u}) = \sum_{j_1}^{N_1} \sum_{j_2}^{N_2} \cdots \sum_{j_m}^{N_m} C[j_1, j_2, \ldots, j_m]\left(u_1^{j_1} u_2^{j_2} \cdots u_m^{j_m}\right),$$

tal que $\mathbf{j} \in \mathbb{Z}_+^m$ e $\mathbf{N} \in \mathbb{N}^m$. Solucionando o sistema de equações encontramos os coeficientes

$$C[0], C[1], C[2], \ldots, C[n], \text{ para } P(u);$$
$$C[j_1, j_2, \ldots, j_m], \text{ para } F(u),$$

de modo que, respectivamente, a Equação 3.1 e/ou a Equação 3.2 aconteçam[2]. Outros métodos nem sempre se expressam dessa forma, mas conseguem resolver a Equação 3.1. Isso pode ser visto em RUGGIERO e DA ROCHA LOPES (1996). Já para se resolver a Equação 3.2, usamos o método simples recém mostrado ou o método de mínimos quadrados que veremos adiante.

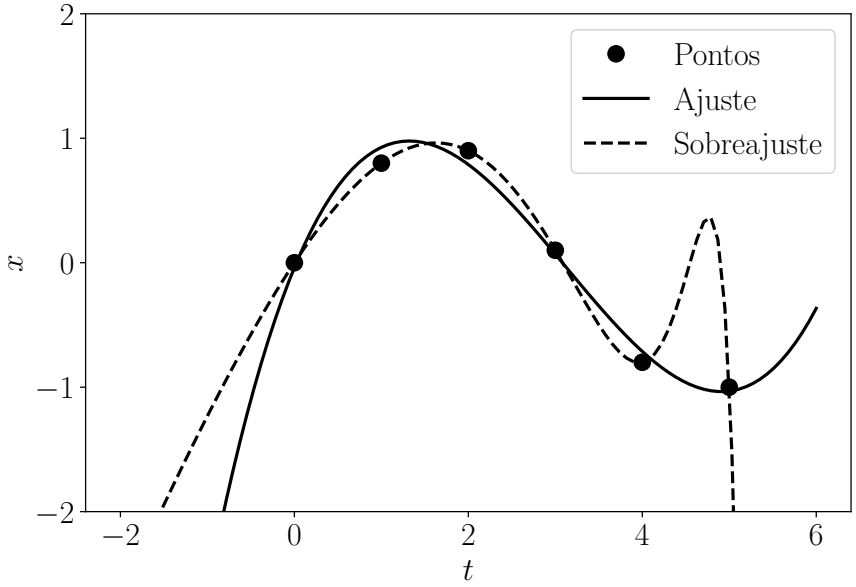
A interpolação polinomial é uma forma de encontrar uma função que se adéque aos dados. Entretanto, ao se trabalhar com um número elevado de pontos, pode haver a necessidade de usar um grau polinomial elevado, ou as vezes a interpolação pode causar grandes oscilações inesperadas ao longo do trajeto, os sobreajustes, tornando complicado alcançar o resultado desejado. Sendo assim, as vezes é preciso obter uma função polinomial ajustada, por mais que não passe por alguns pontos, mas não tenha oscilações indesejáveis, mesmo que o resíduo não seja nulo. Isso é mostrado na figura Figura 3.1.

> Geralmente os coeficientes do polinômio de interpolação determinam menos precisão para uma aproximação desejada. Portanto, não é uma boa ideia determinar os coeficientes somente para uso no cálculo de valores de interpolação. Além disso, você não deve confundir o polinômio de interpolação (e seus coeficientes) como o polinômio de melhor ajuste sobre um conjunto de dados. O ajuste é um processo de suavização, já que o número de coeficientes ajustados é tipicamente muito menor do que o número de pontos de dados. Os valores assim calculados não passarão exatamente sobre os pontos tabulados. Portanto, os coeficientes ajustados podem ser determinados de forma precisa e estável mesmo na presença de erros estatísticos nos valores tabulados. Interpolação, onde o número de coeficientes e o número de pontos tabulados são iguais, leva os valores tabulados como perfeitos. Se os dados contiverem erros estatísticos, então os erros podem ser ampliados em oscilações do polinômio

[2] A existência e unicidade pode ser vista em (RUGGIERO e DA ROCHA LOPES, 1996, p.214)

de interpolação entre os pontos tabulados.(PRESS *et al.*, 2007, p.129, tradução nossa)

Figura 3.1: Diferentes ajustes de curvas



Fonte: Elaborada pelo autor.

Em outras palavras, na interpolação, os pontos conhecidos devem se encaixar perfeitamente na curva estimada na relação de entrada e saída. Neste caso, não há preocupação com a variação da curva ajustada e a quantidade de coeficientes, desde que a curva forneça a saída perfeita (para um dado conjunto de dados). Todavia, em situações reais, em geral temos a necessidade de um modelo mais suave e regularizado. Para isso, o método de *Mínimos Quadrados* é mais adequado, pois a curva é ajustada e a quantidade de coeficientes deve ser reduzida ao máximo.

O Método dos Mínimos Quadrados, do inglês *Least Squares*, é uma técnica de otimização matemática que procura encontrar o melhor ajuste para um conjunto de dados tentando minimizar a soma dos quadrados das diferenças entre o valor estimado e os dados observados, tais diferenças são chamadas resíduos. Com isso, podemos obter uma aproximação com certa margem de segurança.

Para o caso 3.1, devemos determinar

$$C[0], C[1], C[2], \ldots, C[n];$$

reais de modo que

$$\min \sum_{i=0}^{n} w[n] \| P(u[i]) - v[i] \|^2,$$

onde $w[n]$ são fatores de ponderamento. Ou seja, queremos minimizar a soma das distâncias entre a função polinomial aplicada aos pontos de entrada e os pontos de saída. Para o caso 3.2, devemos encontrar os coeficientes

$$C[j_1, j_2, \ldots, j_m],$$

reais, de modo que

$$\min \sum_{i=0}^{n} w[n] \|F(\mathbf{u}[i]) - v[i]\|^2. \tag{3.3}$$

Um detalhe a ser definido é a escolha da norma, usualmente denotada por $\|.\|$. Basicamente ela irá estabelecer o tipo de métrica para a distância entre dois vetores, isto é, $d(F(\mathbf{u}), v) = \|F(\mathbf{u}) - v\|$. Através das normas pode-se realizar testes para determinar o melhor o ajuste com o uso do método de mínimos quadrados. Os tipos de normas usualmente utilizadas no espaço $\mathbb{R}^m$ ou chamadas apenas de norma l_p:

- $\|F(\mathbf{u}) - v\|_p^2 = \sum_{i=0}^{n} |F(\mathbf{u}[i]) - v[i]|^p$ (norma geral l_p),
- $\|F(\mathbf{u}) - v\|_1^2 = \sum_{i=0}^{n} |F(\mathbf{u}[i]) - v[i]|$ (norma l_1 ou retangular),
- $\|F(\mathbf{u}) - v\|_2^2 = \sum_{i=0}^{n} |F(\mathbf{u}[i]) - v[i]|^2$ (norma Euclidiana),
- $\|F(\mathbf{u}) - v\|_\infty^2 = (\max_{i=0}^{n}(|F(\mathbf{u}[i]) - v[i]|))^2$ (norma infinita ou do máximo).

Nestas normas l_p, o valor do parâmetro $p \in \mathbb{R}$ causa um efeito sobre o peso de cada uma das coordenadas, ordenadas por tamanho, no valor da norma. Quanto maior o valor de p, maior a importância de múltiplas coordenadas no valor total da norma, ao passo que p pequeno prioriza as maiores coordenadas em detrimento das menores.

> Existem diferentes penalidades que podem ser aplicadas, com diferentes propriedades. A penalidade mais comumente utilizada é a soma dos quadrados das ponderações, às vezes chamada de "norma-l_2"de w. O motivo é técnico, mas funciona, basicamente, para ajustar melhor os dados que estão autorizados a ter grandes ponderações positivas e negativas. A soma desses quadrados das ponderações gera uma grande penalidade quando ela tem grandes valores absolutos.
>
> Se incorporarmos a penalidade de norma-l_2 nas regressões lineares padrão de quadrados mínimos, obteremos o procedimento estatístico chamado regressão ridge ou regressão em crista. Se, em vez disso, utilizarmos a soma dos valores absolutos (em vez dos quadrados), conhecido como norma-l_1, temos um procedimento conhecido como lasso (Hastie et al., 2009). De modo mais geral, isso é chamado de regularização-l_1. Por motivos bastante técnicos, a regularização-l_1 acaba zerando muitos coeficientes. Como esses coeficientes são as ponderações multiplicadoras

> nas características, a regularização-l_1 realiza efetivamente uma forma automática de seleção de característica.(FAWCETT e PROVOST, 2018, p.137)

No caso em que os dados podem ser corrompidos por ruídos a qual veremos mais adiante, é interessante que o método de mínimos quadrados consiga dar mais importância a certas coordenadas com objetivo de manter a eficácia do algoritmo para a aproximação das equações governantes. Em TRAN e WARD (2016), por exemplo, os autores apresentam uma teoria chamada "Theory from Compressive Sensing" que se baseia em resultados e técnicas de otimização para encontrar os melhores coeficientes quando há a presença de ruido.

Outro exemplo são normas definidas[3] como:

$$\|F(\mathbf{u}) - v\|^2 = \sum_{i=0}^{n} \rho[(F(\mathbf{u}[i]) - v[i])^2],$$

onde ρ é a função de perda. Essa função ρ pode ter várias opções de definição. Algumas opções são:

$$\rho(z) = z, \text{ (linear ou norma-}l_2\text{)}, \tag{3.4}$$

$$\rho(z) = 2(\sqrt{1+z} - 1), \text{ (soft } l_1\text{)}, \tag{3.5}$$

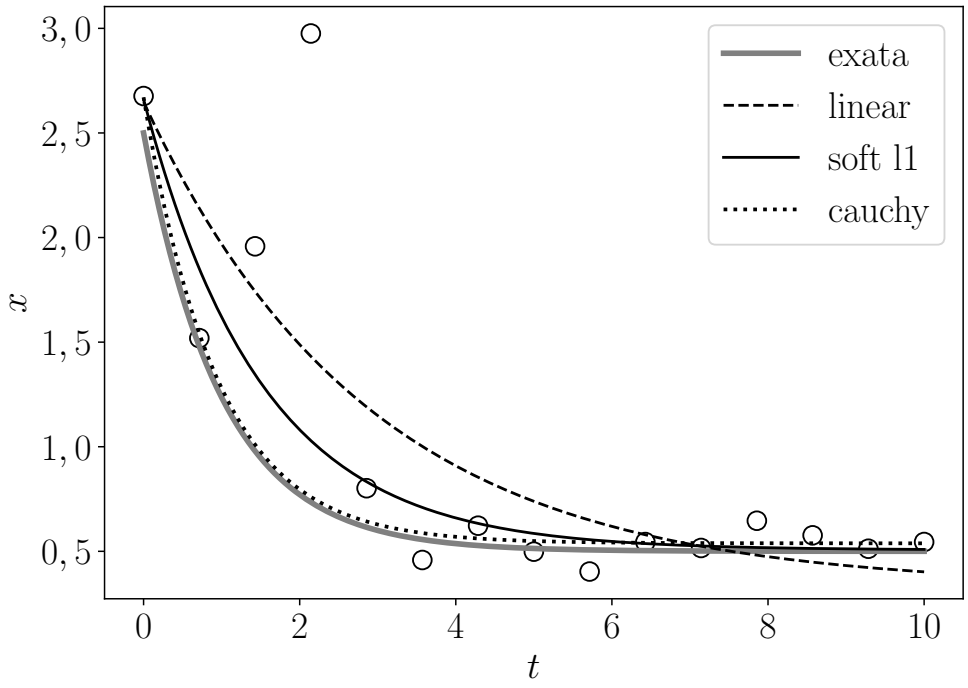
$$\rho(z) = \ln(1+z), \text{ (cauchy)}, \tag{3.6}$$

em que $z = (F(\mathbf{u}[i]) - v[i])^2$.

A Figura 3.2 mostra uma ilustração de ajustes com essas diferentes normas. Nela apresenta dados com ruídos(isso será definido mais adiante) e diferentes tipos de normas aplicadas ao método de mínimos quadrados. As normas atribuem um nível de penalidade, e com isso, são uteis para a seleção dos dados necessários. Funciona como uma medida corretiva com relação ao ruído. Podemos ver que a norma "cauchy" foi a melhor escolha para corrigir o ajuste de modo a se aproximar da solução exata.

[3] A definição dessas normas podem ser vistas em SCIPY (2007).

Figura 3.2: Diferença entre alguns ajustes de curvas



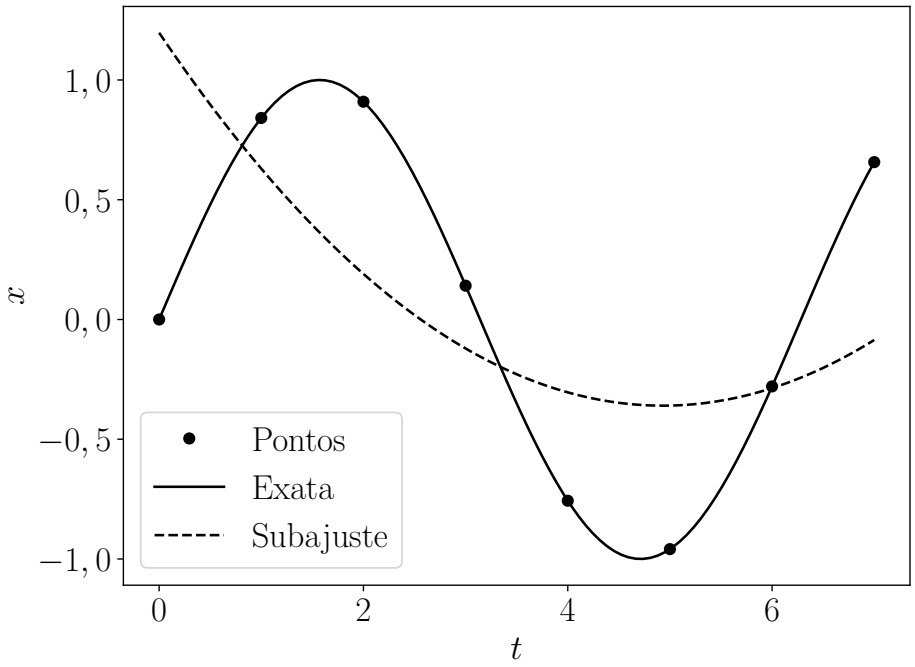
Fonte: Scipy, 2019.

Através desses exemplos, podemos ver que a escolha da norma influência na aplicação e nos resultados para cada tipo de sistema. Nesse contexto, torna-se interessante o teste com vários tipos de normas no método de Mínimos Quadrados a fim de buscar a aproximação mais eficiente.

Como vimos, a interpolação pode gerar sobreajuste e o método de mínimos quadrado pode ser a alternativa para melhorar o ajuste. Contudo, existe mais um contratempo, o método de mínimos quadrados apresenta a pequena inconveniência de que, ao trabalhar com um grande número de pontos, podemos ter um subajuste(underfitting). Ao contrario de sobreajuste, o modelo se torna subajustado quando possui um ajuste insatisfatório aos dados observados.
A Figura 3.3 mostra que o ajuste teve um grau de liberdade muito pequeno e não conseguiu se ajustar aos pontos, tornando muito diferente do comportamento esperado.

> Um perigo comum em aprendizado de máquina é o sobreajuste - produzir um modelo de bom desempenho com os dados que você treina, mas que não lide muito bem com novos dados.
>
> [...]O outro lado é o subajuste, produzindo um modelo que não desempenha bem nem com os dados usados no treino, apesar de que, quando acontece isso, você decide que seu modelo não é bom o suficiente e continua a procurar por melhores. (GRUS, 2018, p.207)

Figura 3.3: Ilustração de um subajuste



Fonte: Elaborada pelo autor.

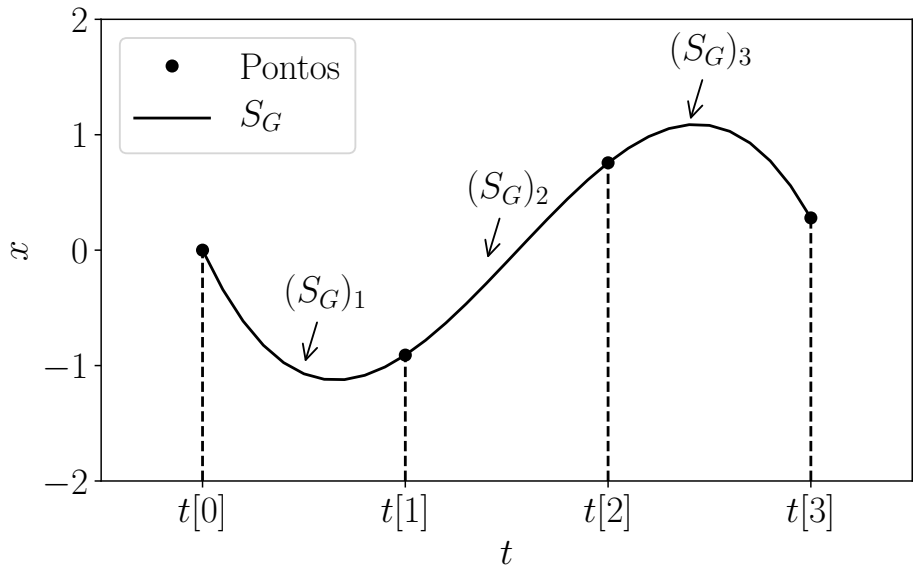
Para isso, podemos buscar a alternativa de fazer ajuste por partes utilizando uma função polinomial de grau menor, e impor algumas condições para que a função seja contínua e diferenciável. Esse tipo de particionamento é chamada de método Spline.

Definição 3.3. Dado um conjunto de $n+1$ pontos ou pares ordenados $\{(u[i], v[i])\}_{i=0}^{n}$ tais que $u[i] < u[i+1]$. Uma Spline de grau G que interpola esses pontos é uma função S com as seguintes propriedades:

- em cada subintervalo $[u[i], u[i+1]]$, $S_G(u)$ é um polinômio de grau menor ou igual a G.
- $S_G(u)$ é contínua e tem derivada contínua até ordem $G-1$ em todo o intervalo
- $S_G(u[i]) = v[i]$

A Figura 3.4 mostra que em vez de encontrar um polinômio de algum grau que se ajuste por todos os dados, podemos ajustar em vários subintervalos e garantir condições para que a nossa aproximação seja continua e tenha derivadas contínuas até a ordem que quisermos.

Figura 3.4: Ilustração de Spline



Fonte: Elaborada pelo autor.

Existem várias formas de fazer um ajuste Spline, um tipo que poderemos usar nesse trabalho é a chamada “Basis Spline” ou B-Spline, onde

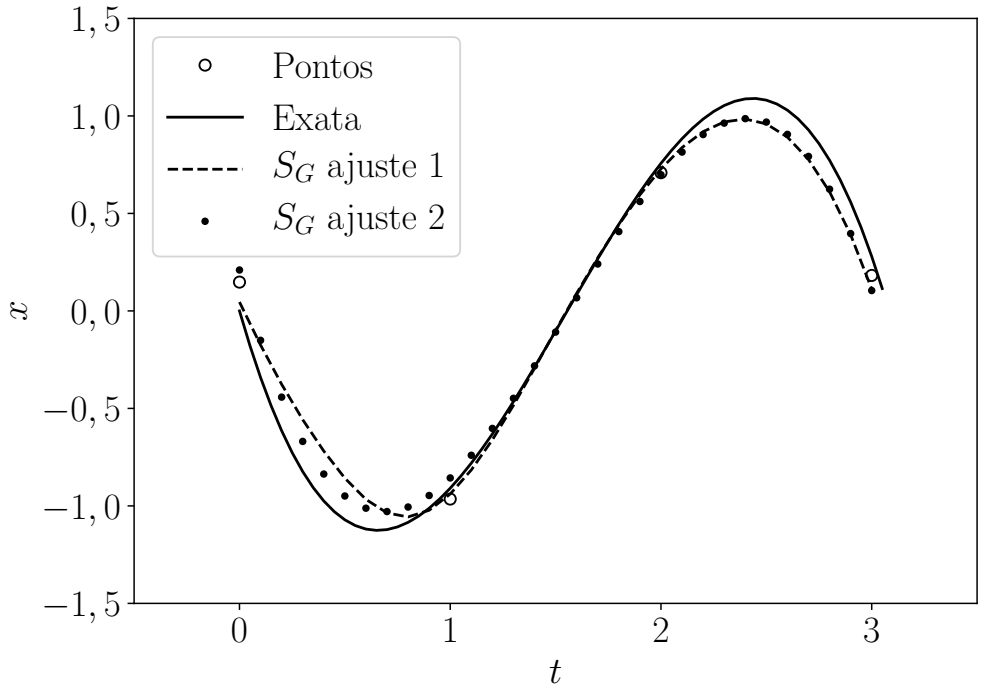
$$S(t) = \sum_{i=0}^{n} u[n]B[j, G](t), \tag{3.7}$$

e as bases $B[j, G](t)$ são obtidas através da forma recursiva de Carl De Boor.

Essa versão de Spline é definida de forma paramétrica e possui propriedades avançadas de ajuste de dados e suavização.

A Figura 3.5 apresenta o ajuste de uma B-Spline. A curva preenchida é exata, os pontos(vazios) possuem algum valor de ruído, enquanto que a tracejada e pontilhada são ajustadas. Isso mostra que é possível alternar o ajuste de modo a encontrar a melhor aproximação a exata.

Figura 3.5: Ilustração de Ajuste usando B-Spline



Fonte: Elaborada pelo autor.

As definições e variações de Spline podem ser vistos em RUGGIERO e DA ROCHA LOPES (1996), PEREIRA (2014), ARAÚJO (2016) e POLLOCK *et al.* (1999).

3.2 Ruído Observacional

A presença de ruídos em dados obtidos em experimentos pode ser inevitável e isso pode esconder informações importantes para identificar o sistema dinâmico associado a evolução temporal, tornando o seu estudo uma tarefa complexa. De acordo com SCHOPF (2007 apud MACHADO, 2003, p.32), "Ruído pode ser definido como o conjunto das influências não-sistemáticas sobre o comportamento de um sistema, não estando compreendido no modelo determinístico (previsível) desse sistema".

A contaminação por ruído pode acarretar interpretações incorretas dos resultados, principalmente quando o objetivo é realizar previsão. É importante ressaltar que existem duas classes de ruído capazes de afetar nosso método. Ruído dinâmico é aquele que pode afetar a evolução do sistema, isto é, quando o ruído entra na equação diferencial. Ruído observacional é aquele que afeta as variáveis já produzidas pelo sistema, aqui chamadas de variáveis de posição. Assim, por exemplo, em sistemas calculados numericamente, a precisão numérica leva a ruídos entrando nas equações diferenciais, o chamamo de ruído dinâmico, mas que é muito pequeno e

será ignorado aqui. Frequentemente encontramos também situações nas quais ocorrem modificações nas amostras do estado de um experimento, ou seja, o ruído entra na variável de posição e é do tipo que chamamos de ruído observacional.

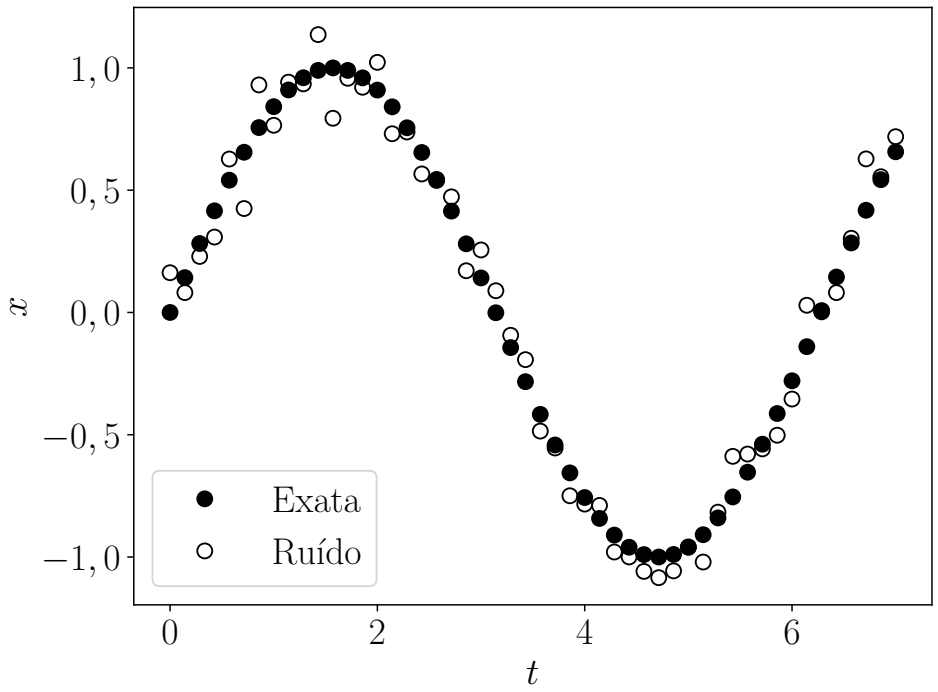
Tanto o ruído dinâmico quanto o ruído observacional podem ser aditivos ou multiplicativos. Neste trabalho apenas exploraremos os efeitos de ruído observacional aditivo. Ou seja, os dados são gerados por equações diferenciais supostamente não afetadas por ruídos dinâmicos, mas os valores dos dados sofrem alteração após o registro na série temporal. Isso pode ser visto na Figura 3.6. Tais perturbações podem ser causadas, por exemplo, por imprecisão ou limitações nos equipamentos de medida.

Para efeito de análise, considera-se que os termos das séries temporais possuem contaminação por ruídos aditivos. Esta contaminação é definida como

$$\mathbf{x_r} = \mathbf{x} + \mathbf{r}, \tag{3.8}$$

onde $\mathbf{r} = (r_1, r_2, \ldots, r_m)$ são os componentes aleatórios.

Figura 3.6: Ilustração de ruídos



Fonte: Elaborada pelo autor.

Visto isso, após produzir as séries temporais do sistema dinâmico com o método Método de Runge-Kutta de quarta ordem (Seção A.1), faremos o acréscimo do ruído em cada termo de cada série temporal. O valor aleatório não pode ser muito grande,

pois pode destruir completamente a informação.

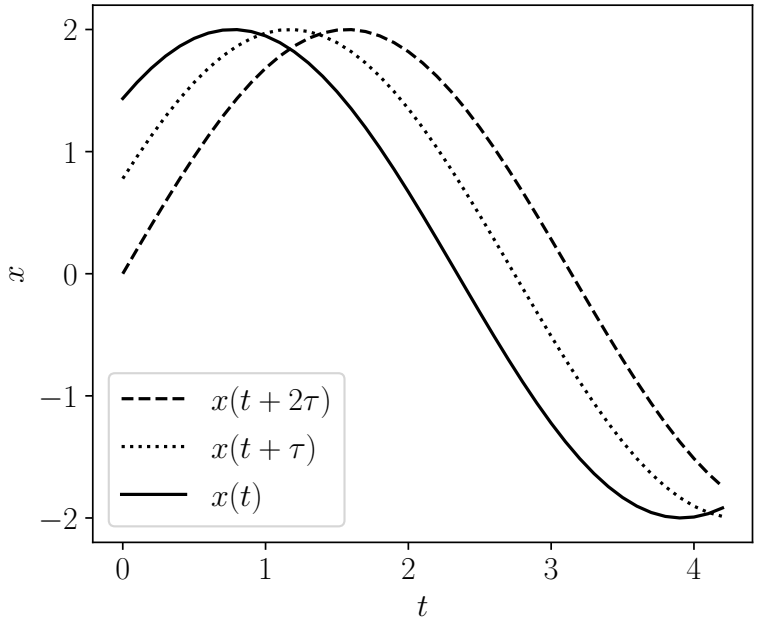
3.3 Método de Coordenadas Atrasadas

Em problemas reais, normalmente tem-se uma série temporal de dados experimentais, mas não se sabe se apenas com essa série temporal é possível recuperar uma dinâmica de modo que possa descrever e prever a evolução do temporal. Além disso, detectar a presença de um atrator estranho. Em (STROGATZ, 1994, p.438, tradução nossa),

> [...]como você poderia demonstrar a presença de um atrator estranho, dado que você só mede uma única série temporal $x(t)$? [...]Roux et al.(1983) exploraram uma surpreendente técnica de análise de dados, agora conhecida como reconstrução de atratores (Packard et al. 1980, Takens 1981). A alegação é que, para alguns sistemas governados por um atrator, a dinâmica no espaço de fase completo pode ser reconstruída a partir de medições de apenas uma única série temporal! De alguma forma, essa única variável carrega informações suficientes sobre todas as outras. O método é baseado em atrasos de tempo. Por exemplo, defina um vetor bidimensional $\mathbf{B}(t) = (x(t), x(t+\tau))$, onde τ é o valor de atraso. Então a série temporal $x(t)$ gera uma trajetória $\mathbf{B}(t)$ em um espaço de fase bidimensional.

Sendo assim, no espaço de fases formados pela evolução temporal das variáveis de estado $x(t), x(t+\tau), x(t+2\tau), \ldots, x(t+(\mu-1)\tau)$, onde μ é chamado de *dimensão de imersão* e τ é o passo de reconstrução múltiplo do espaçamento entre os termos da série temporal. O atrator reconstruído é topologicamente equivalente ao atrator verdadeiro, sobre o qual conhece-se apenas a evolução temporal da variável de estado $x(t)$. . A Figura 3.7 ilustra o método de coordenadas atrasadas.

Figura 3.7: Ilustração do método de coordenadas atrasadas.



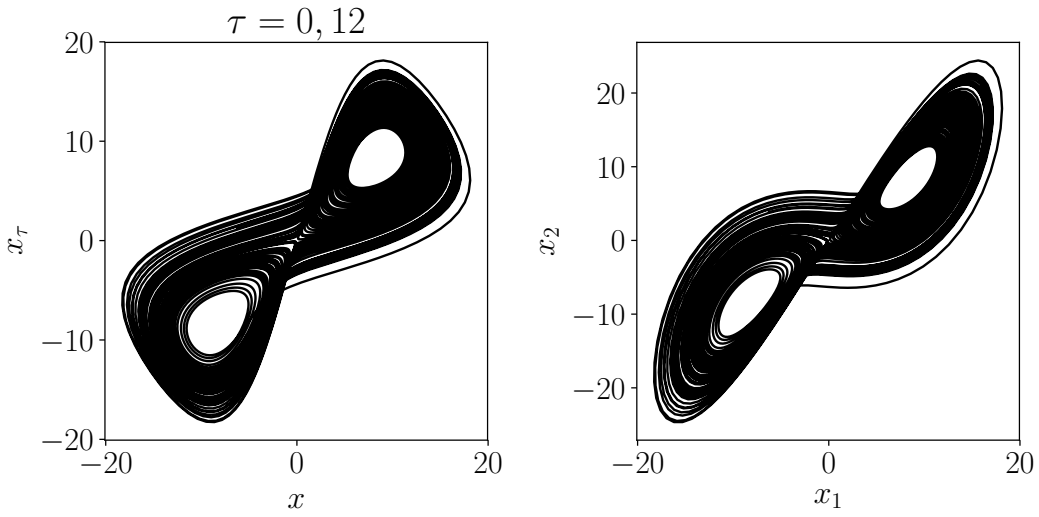
Fonte: Elaborada pelo autor.

Contudo, é inevitável pensar como saber o valor de μ e de τ, como (STROGATZ, 1994, p.440, tradução nossa) destaca,

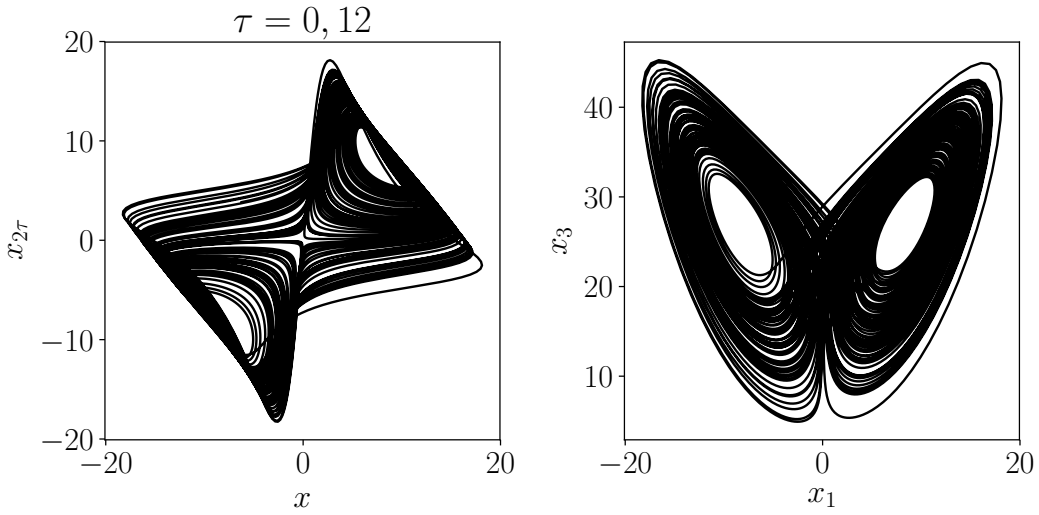
> Primeiro, como escolher a dimensão de imersão, ou seja, o número de atrasos? A série temporal deve ser convertida em um vetor com dois componentes, três ou mais? Grosso modo, é necessário um atraso suficiente para que o atrator subjacente possa se desenredar no espaço de fase. A abordagem usual é aumentar a dimensão de incorporação e depois calcular as dimensões de correlação dos atratores resultantes. Os valores calculados continuarão aumentando até que a dimensão de incorporação seja grande o suficiente; então há espaço suficiente para o atrator e a dimensão de correlação estimada se estabilizará no valor 'verdadeiro'.

A escolha de μ é feita com o método de falsos vizinhos, podendo ser vista em OLIVEIRA JR. (2012) e τ é feita a partir do método de informação mútua que pode ser visto em CAMPANHARO *et al.* (2014). Ao menos, exemplificaremos na Figura 3.8, o resultado da reconstrução do atrator, para o caso em que usamos a variável x_1 solucionada numericamente do sistema de Lorenz e definimos $x[n]$, $x_\tau[n]$ e $x_{2\tau}[n]$ como a séries em coordenadas atrasadas usando $\mu = 3$ e $\tau = 0,12$.

Figura 3.8: Reconstrução do atrator de Lorenz



(a) $x \times x_\tau,\ x_1 \times x_2$



(b) $x \times x_{2\tau},\ x_1 \times x_3$

Fonte: Elaborada pelo autor.

3.4 Previsão de Sistemas

A partir das séries temporais, podemos fazer a recuperação de sistemas dinâmicos. A recuperação é feita através de aproximação por funções polinomiais. Vamos usar o Método de Aproximação Polinomial (Seção 3.1).

Na primeira etapa procuramos as funções polinomiais unidimensionais de grau G, aplicada aos dados temporais, para cada uma das m variáveis do sistema,

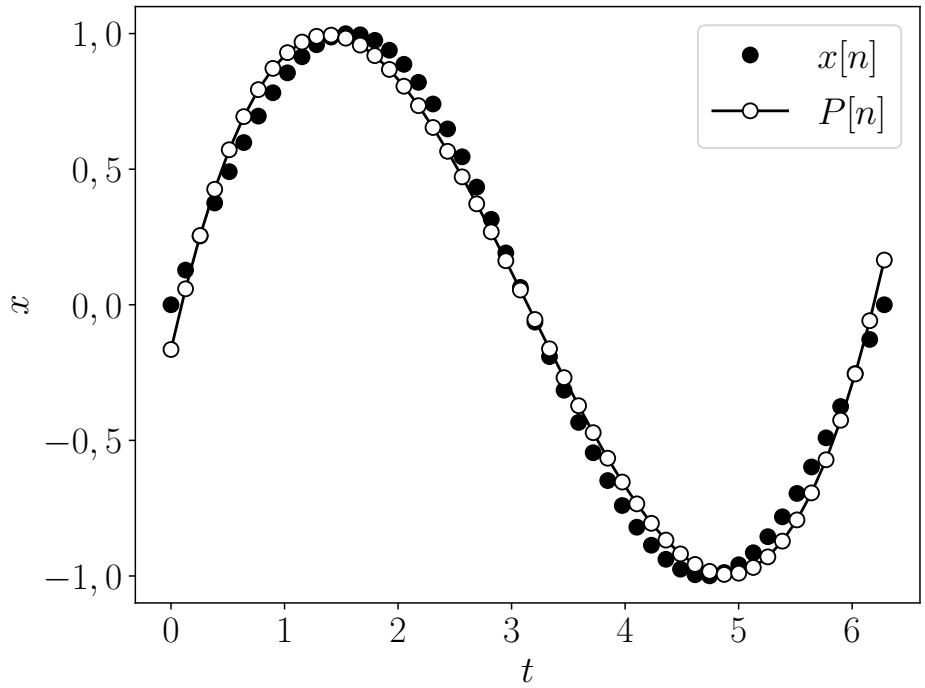
$$P_1(t[n]), P_2(t[n]), P_3(t[n]), \ldots, P_m(t[n]), \tag{3.9}$$

que se ajustem, respectivamente, as séries temporais

$$x_1[n], x_2[n], x_3[n], \ldots, x_m[n]. \tag{3.10}$$

Esse ajuste pode ser uma simples interpolação até um ajuste usando B-Spline, bem como visto na Seção 3.1. A Figura 3.9 ilustra esse ajuste.

Figura 3.9: Ilustração da aproximação de P(t[n]).



Fonte: Elaborada pelo autor.

Daí, prosseguimos para a segunda etapa fazendo o cálculo da diferenciação com respeito ao tempo dessas funções $\mathbf{P}(t[n])$, isto é, calculamos as derivadas $\mathbf{P}'(t[n])$ que também são funções polinomiais e representam a velocidade com respeito ao

tempo. Definimos por

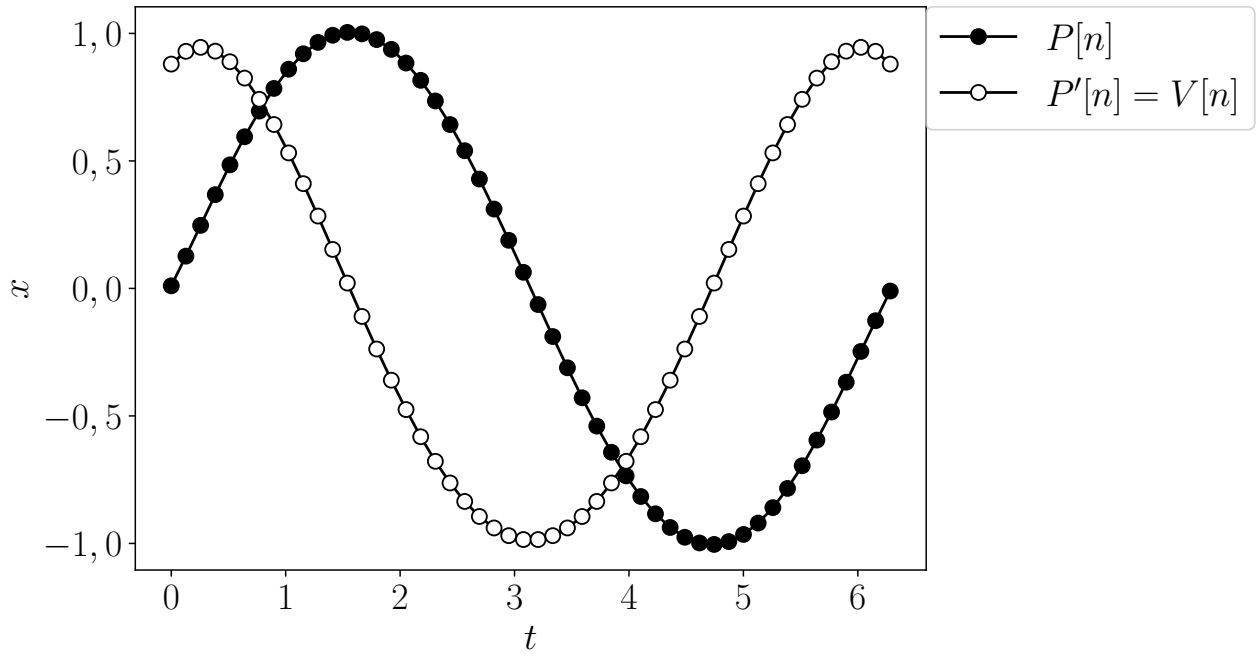
$$V_1(t[n]), V_2(t[n]), V_3(t[n]), \ldots, V_m(t[n]), \tag{3.11}$$

Este cálculo de diferenciação são aproximações de

$$\dot{x}_1[n], \dot{x}_2[n], \dot{x}_3[n], \ldots, \dot{x}_m[n]. \tag{3.12}$$

Por sua vez, essas funções $\mathbf{V}(t[n])$ serão fundamentais para calcular a aproximação da dinâmica do sistema. A Figura 3.10 ilustra esse cálculo.

Figura 3.10: Ilustração de V(t[n]).



Fonte: Elaborada pelo autor.

Daí passaremos para a próxima etapa buscando funções polinomiais multidimensionais de grau G, isto é,

$$\begin{gathered} F_1(P_1[n], P_2[n], P_3[n], \ldots, P_m[n]), \\ F_2(P_1[n], P_2[n], P_3[n], \ldots, P_m[n]), \\ F_3(P_1[n], P_2[n], P_3[n], \ldots, P_m[n]), \\ \vdots \\ F_m(P_1[n], P_2[n], P_3[n], \ldots, P_m[n]), \end{gathered} \tag{3.13}$$

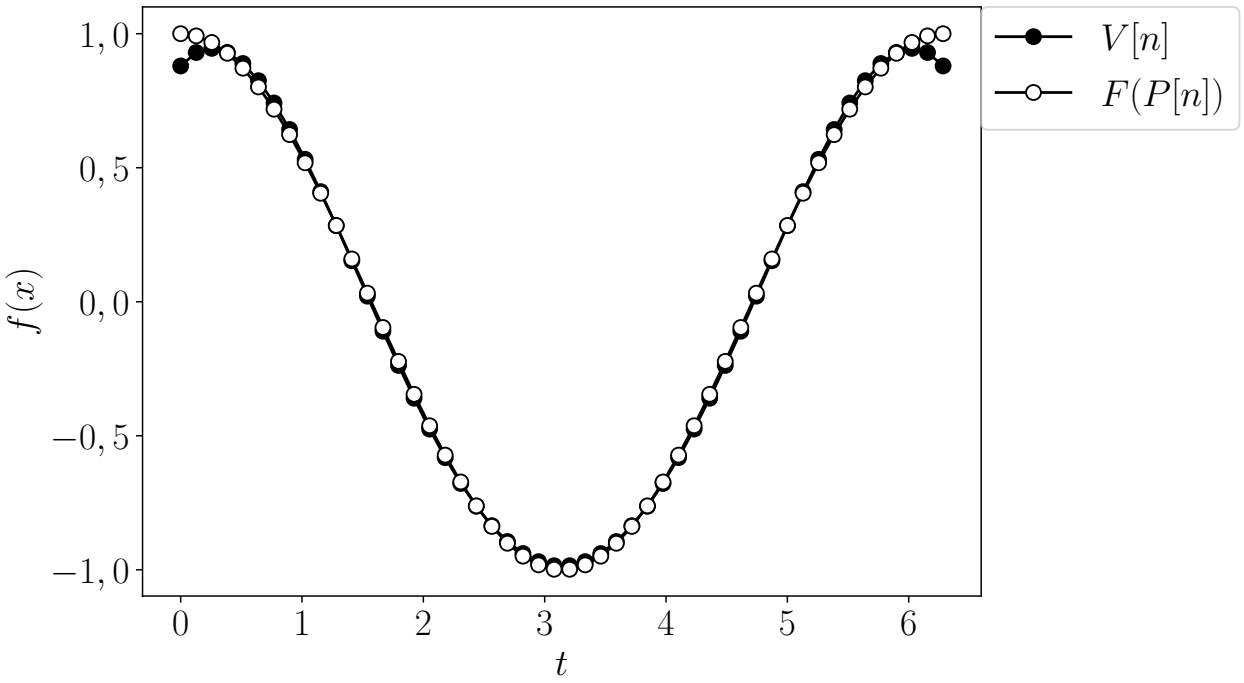
que se ajustem, respectivamente aos Dados 3.11, através do Método de Aproximação Polinomial (Equação 3.3).

Os Dados 3.13 representam as aproximações das variáveis espaciais, isto é,

$$\mathbf{f}(x_1[n], x_2[n], x_3[n], \ldots, x_m[n]), \tag{3.14}$$

e serão as funções de nosso interesse, pois é a partir delas que faremos a previsão. A Figura 3.11 ilustra esse ajuste.

Figura 3.11: Ilustração de F(P[n]).



Fonte: Elaborada pelo autor.

Solucionaremos numericamente o Sistema 3.13 usando o Método de Runge-Kutta de quarta ordem (Apêndice A.1) com condições iniciais. As soluções numéricas são as séries temporais que definimos por X_m. Como queremos previsão do sistema, então usaremos como condições iniciais algum dos Dados 3.10, isto é,

$$X_1[0] = x_1[n], X_2[0] = x_2[n], X_3[0] = x_3[n], \ldots, X_m[0] = x_m[n].$$

Evidentemente que quando falamos de previsão, queremos saber de dados desconhecidos, então devemos usar os dados finais de $x[n]$. Entretanto, é importante ressaltar que essa metodologia não obriga a usar os dados finais de $x[n]$, podendo servir a qualquer dado.

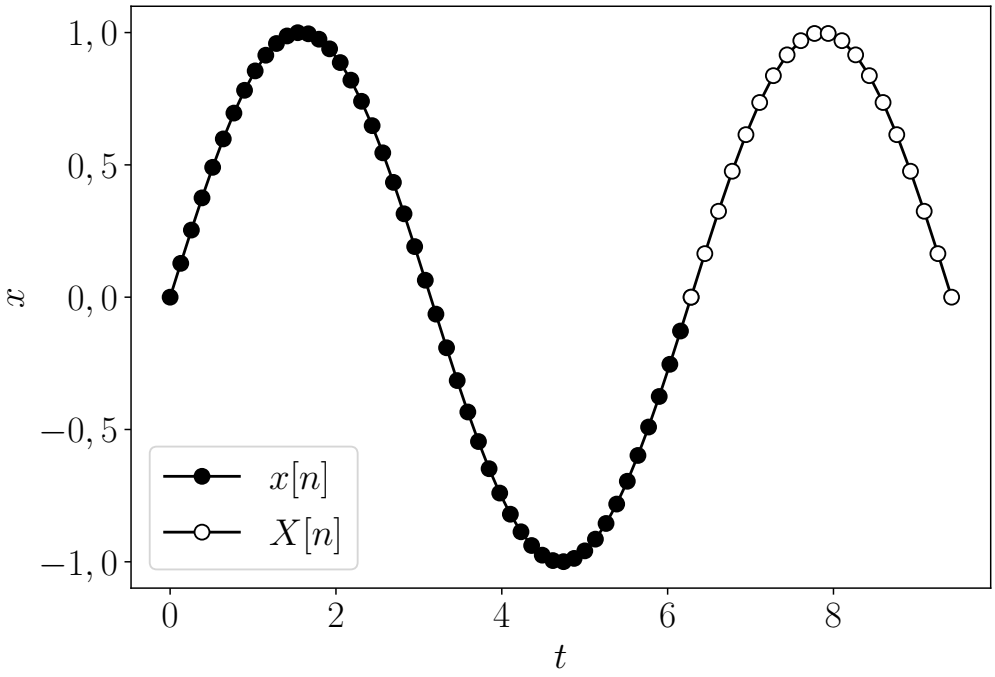
Essa etapa nos dará as séries temporais que constituem os valores de estado

futuro, ou seja,

$$X_1[k] \cong x_1[n+k], X_2[k] \cong x_2[n+k], X_3[k] \cong x_3[n+k], \ldots, X_m[k] \cong x_m[n+k],$$

onde $k \in \mathbb{N}$. A Figura 3.12 ilustra esses valores.

Figura 3.12: Ilustração da previsão.



Fonte: Elaborada pelo autor.

Capítulo 4

Resultados e Discussões

4.1 Sistema de Lorenz

4.1.1 Geração das séries temporais

Dado o sistema de Lorenz,

$$\begin{aligned} \dot{x}_1 &= \sigma(x_2 - x_1), \\ \dot{x}_2 &= r x_1 - x_2 - x_1 x_3, \\ \dot{x}_3 &= x_1 x_2 - b x_3, \end{aligned}$$

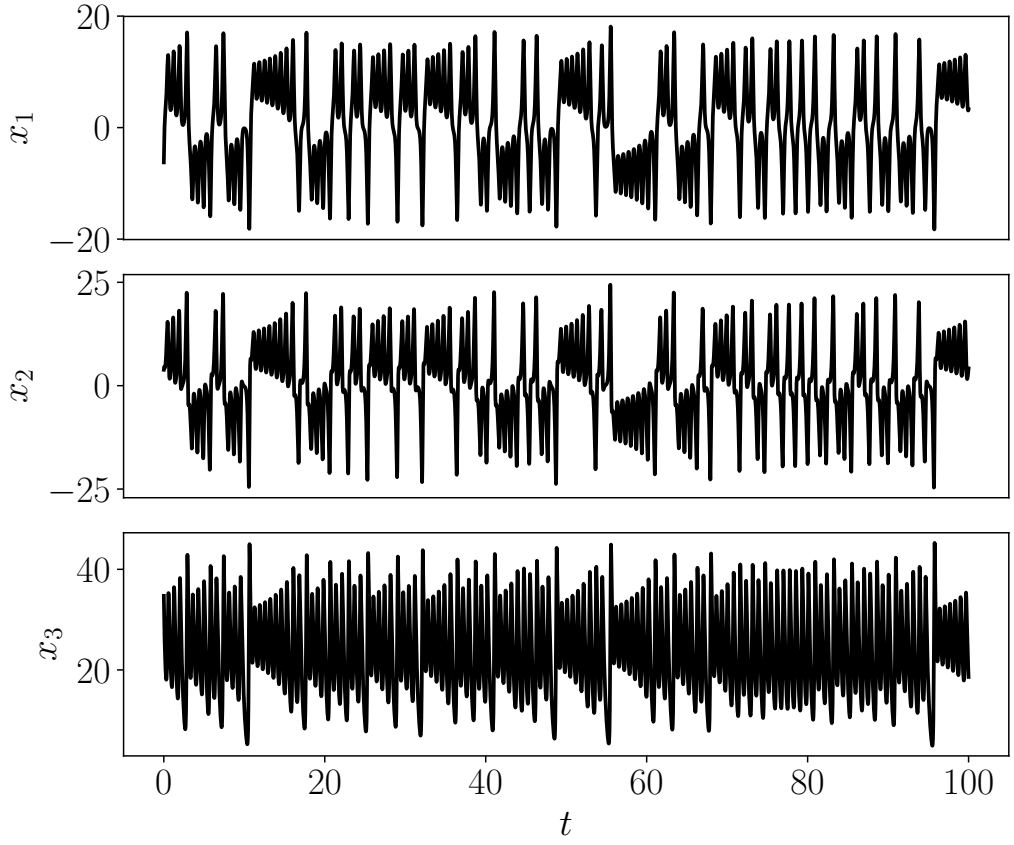
com as condições iniciais

$$\begin{aligned} x_1[0] &= 0,1\ , \\ x_2[0] &= 0,1\ , \\ x_3[0] &= 0,1\ , \end{aligned} \tag{4.1}$$

em um tempo t discretizado, sobre um passo $\Delta t = 10^{-2}$. Inicialmente com $t[0] = 0.0$ e $t[N] = 200$, onde N é o índice final da série, vamos usar o Método de Runge-Kutta de quarta ordem (Apêndice A.1) uma vez para assegurar de eliminar o transiente inicial, assim, geram-se 20001 pontos em cada série.

Dessas séries, escolhemos as novas condições iniciais usando os últimos dados. Daí, realizamos pela segunda vez o Método de Runge-Kutta de quarta ordem (Apêndice A.1), com $t[0] = 0$ e $t[n] = 100$. Assim, obtemos a figura similar a Figura 2.11 gerando $N = 10001$ pontos das séries temporais. Através da Figura 4.1 analisamos qualitativamente se há vestígios do transiente inicial e percebemos que, aparentemente, as séries estão em regime permanente.

Figura 4.1: Séries Temporais do Sistema de Lorenz



Fonte: Elaborada pelo autor.

4.1.2 Previsão do Sistema de Lorenz

Antes de trabalhar com a metodologia apresentada no Capitulo 3, se o sistema de Lorenz é caótico, então já podemos ter ideia do tempo de previsão através de uma analise de pertubações na condição inicial, isto é, usar condições iniciais bem próximas que definiremos como $\mathbf{x}_{\boldsymbol{\delta}} = \mathbf{x} + \delta$. Daí, iremos solucionar o sistema de Lorenz com cada condição inicial e vamos procurar uma média do tempo de previsão analisando a evolução da diferença entre as soluções. Com isso, poderemos ter mais uma demonstração para a eficiência do nosso método. Essa diferença é definida como,

$$E(\mathbf{x} - \mathbf{x}_{\boldsymbol{\delta}}) \quad = \quad \frac{|x_1 - (x_\delta)_1| + |x_2 - (x_\delta)_2| + |x_3 - (x_\delta)_3|}{|x_1| + |x_2| + |x_3|}. \tag{4.2}$$

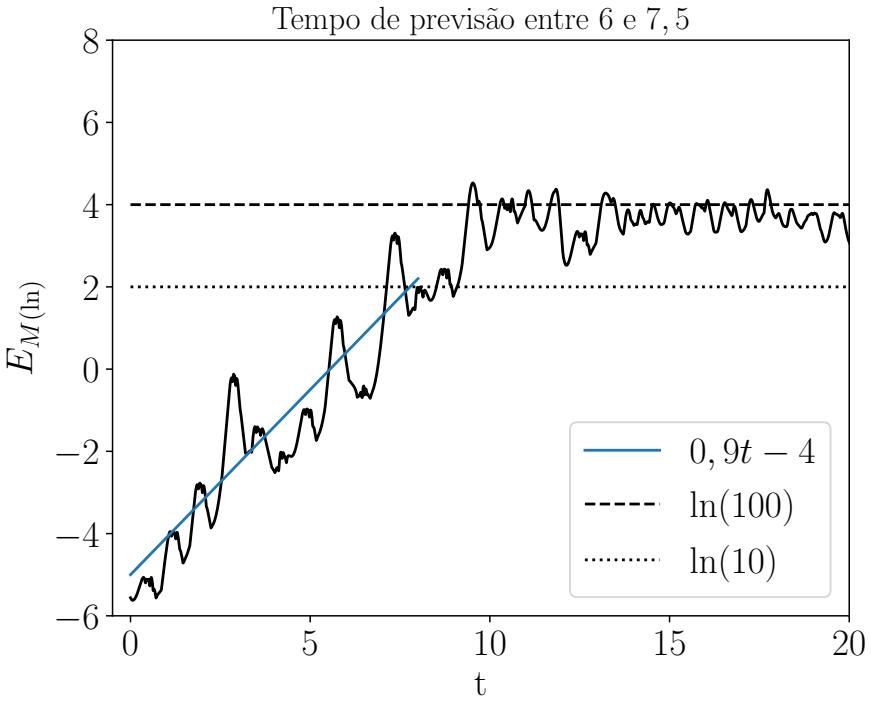
Usamos as condição iniciais das nossas séries e acrescentamos $\delta = 1/(i + 100)$, onde $i = 1, \ldots, 10$, ou seja, usaremos 10 diferentes condições iniciais próximas. Esse valor não é padrão, mas é um valor razoável para realizar uma média. Em seguida, fazemos a média do logaritmo do erro percentual $E_{\%} = 100E(\mathbf{x} - \mathbf{x}_{\boldsymbol{\delta}})$ que definimos

por,

$$E_{M(\ln)} = \sum_{i=1}^{10} [\ln E_{\%}(\mathbf{x} - \mathbf{x}_{\boldsymbol{\delta}_{[i]}})]/10. \tag{4.3}$$

O tempo de previsão é dado pela investigação da inclinação da evolução exponencial do erro e uma reta com a mesma inclinação, vejamos na Figura 4.2. O tempo de previsão estará no trecho em que o crescimento exponencial sofre uma grande elevação e seus valores passam a saturar.

Figura 4.2: Crescimento da média exponencial do erro



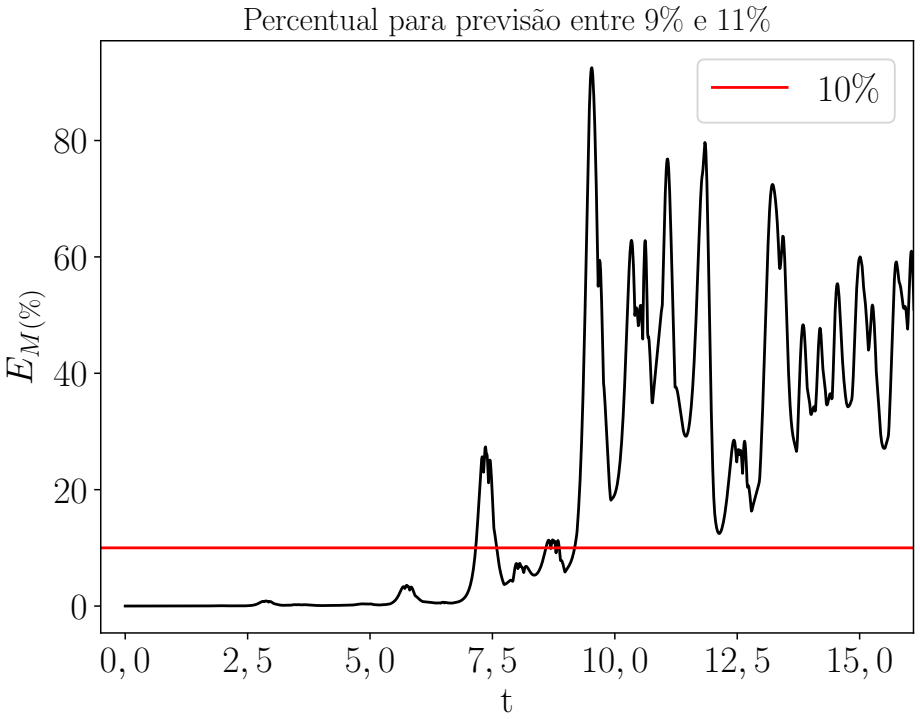
Fonte: Elaborada pelo autor.

Logo, o valor de inclinação coincide com o valor apresentando em Expoente de Liapunov (Equação 2.4.3) e nosso tempo de previsão deve estar entre 6 e 7,5 unidades de tempo. Indubitavelmente que analisar a inclinação da curva não é um bom parâmetro para nossos próximos resultados, então a Figura 4.3 apresenta o percentual do erro até esse tempo de previsão. Esse percentual será nosso parâmetro.

Portanto, para a nossa analise de previsão, estabeleceremos que o tempo de previsão será escolhido pelo critério de atingimento de 10% do erro percentual.

Agora, seguiremos gradualmente a metodologia apresentada na Previsão de Sistemas (Seção 3.4). De início, nesta etapa, é feito o uso da metologia com todos os 10001 pontos apresentados na Séries Temporais do Sistema de Lorenz (Figura 4.1). Na sequência, precisamos usar uma função polinomial unidimensional de posição. Aplicaremos o método de B-Spline. Nesse trabalho, usamos a implementação das

Figura 4.3: Crescimento médio percentual do erro



Fonte: Elaborada pelo autor.

funções B-spline contida no manual SCIPY (2007). A implementação possui limitação no grau da função, tal que podemos usar graus de 1 a 5. Entretanto, não usaremos grau 1 nem 2, pois estes não produzem derivadas contínuas e suaves, necessárias à nossa recuperação da dinâmica.

Dessa forma, tendo a oportunidade, precisamos analisar qual o melhor grau para a B-Spline, entre 3 e 5, de modo que queremos verificar o quão difere no resultado do tempo de previsão. Em outras palavras, nosso critério será verificar qual a melhor previsão com relação a escolha do grau dessa função, nessa primeira etapa.

Antes de analisar os resultados, é preciso lembrar que para se verificar o tempo de previsão é preciso realizar todas as etapas da metodologia da Previsão de Sistemas (Seção 3.4). Como bem sabemos, na terceira etapa, também precisa-se verificar qual a melhor norma para as funções multidimensionais. As normas que iremos trabalhar serão as que abordamos na Equação 3.4, Equação 3.5 e Equação 3.6 . Nessa etapa desse trabalho, essas funções possuem grau menor ou igual a 2, por comparação ao sistema de Lorenz. Além disso, devemos verificar em diferentes condições iniciais.

Logo, nossa análise será buscar qual o melhor grau na primeira etapa e qual a melhor norma na terceira etapa.

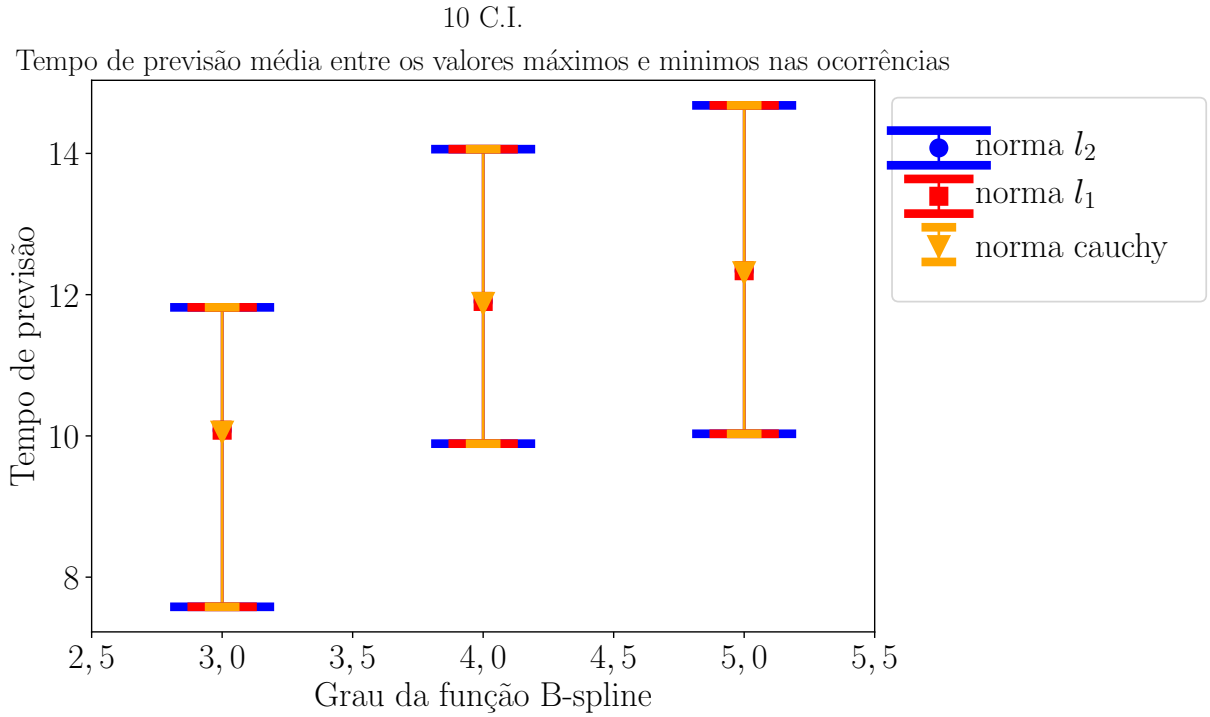
O tempo de previsão é aquele em que o erro percentual entre as séries originais

e estimadas atinge 10%. Esse erro é definido por,

$$E_{\%}(\mathbf{x}-\mathbf{x_{prev}}) \quad = \quad 100\frac{|x_1-(x_{prev})_1|+|x_2-(x_{prev})_2|+|x_3-(x_{prev})_3|}{|x_1|+|x_2|+|x_3|}, \quad (4.4)$$

A Figura 4.4 mostra o resultado de 10 realizações usando diferentes condições iniciais. Com isso, temos nosso tempo de previsão médio e sua margem de erro com relação ao grau e a norma utilizados.

Figura 4.4: Análise para escolha do grau e da norma para o melhor tempo de previsão.



Fonte: Elaborada pelo autor.

Portanto, temos que o melhor grau para a função B-spline é 5 e a norma pode ser qualquer uma das três.

A Figura 4.5 mostra o resultado de B-spline de grau 5 interpolando os dados da série.

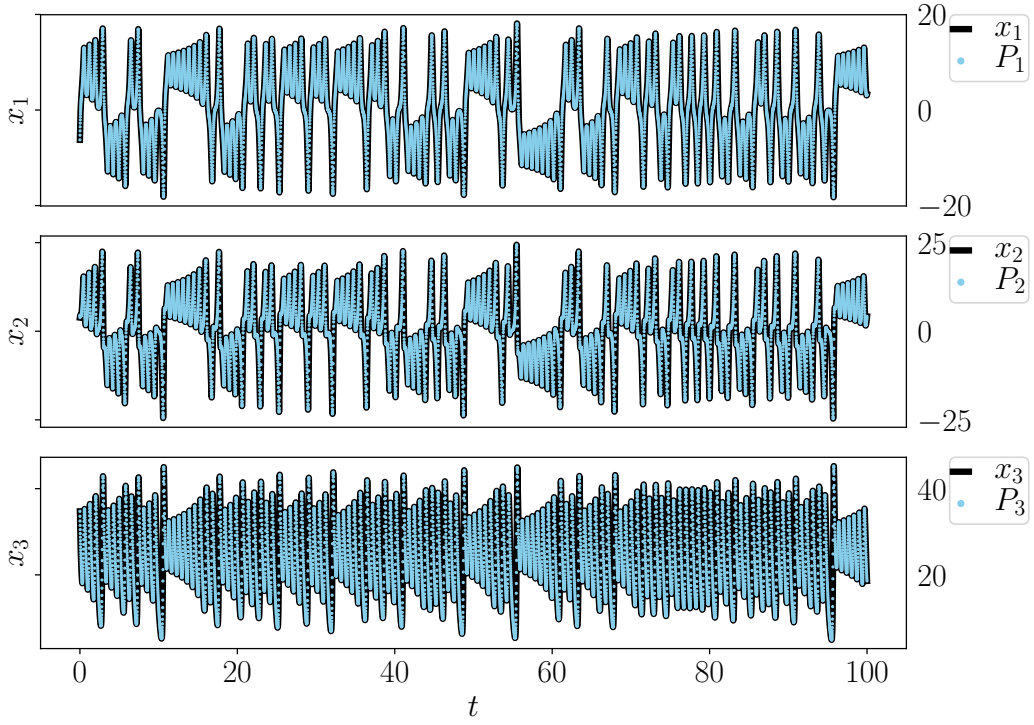
Daí, definindo o erro percentual das funções polinomiais de posição como

$$E_{\%}(\mathbf{x}-\mathbf{P}) \quad = \quad 100\frac{|x_1-P_1|+|x_2-P_2|+|x_3-P_3|}{|x_1|+|x_2|+|x_3|}, \quad (4.5)$$

temos como resultado $E_{\%}(\mathbf{x}-\mathbf{P}) < 10^{-13}$.

Como vimos, as funções polinomiais B-spline $\mathbf{P}(t)$ se ajustam as variáveis de posição $\mathbf{x}(t)$ sobre um intervalo de tempo de pontos distintos $t[0], t[1], t[2], \ldots, t[n]$. É interessante avaliar se essas funções polinomiais $\mathbf{P}(t)$ estão realmente bem ajustadas

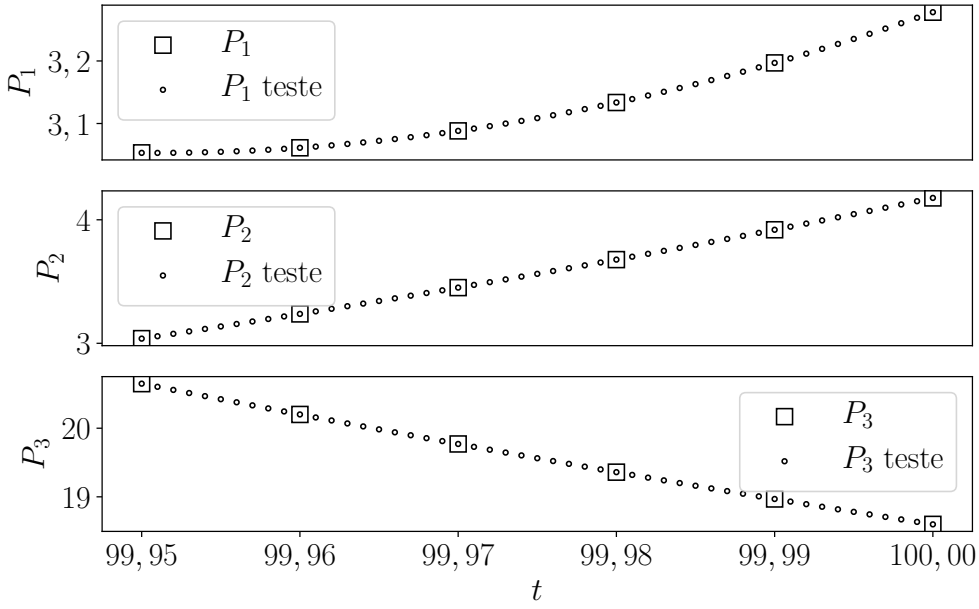
Figura 4.5: B-Spline interpolante para aproximação das Séries Temporais originais.



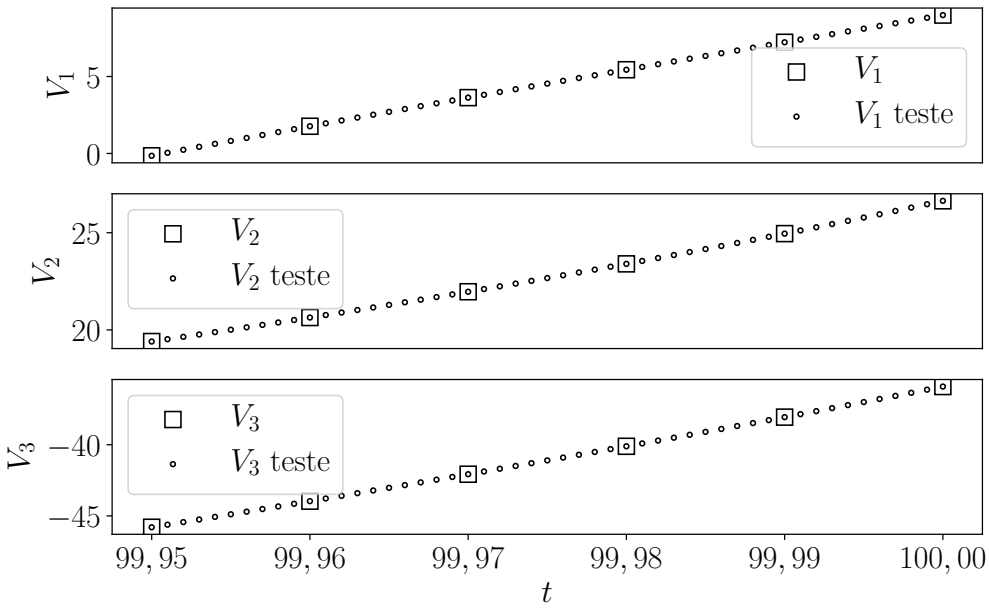
Fonte: Elaborada pelo autor.

no momento em que o intervalo de tempo é mais dividido. Temos que verificar se usando os mesmos parâmetros das funções polinomiais B-splines de posição e velocidade, e passando por um tempo mais discretizado, então não fogem do comportamento e passa pelos mesmos pontos da discretização do tempo usada anteriormente. Em suma queremos verificar se não há oscilações, descontinuidade e desencontros de pontos ao tomarmos, por exemplo, $\Delta t = 10^{-3}$ em que o resultado é mostrado na Figura 4.6.

Figura 4.6: Teste de suavidade



(a) **P** teste

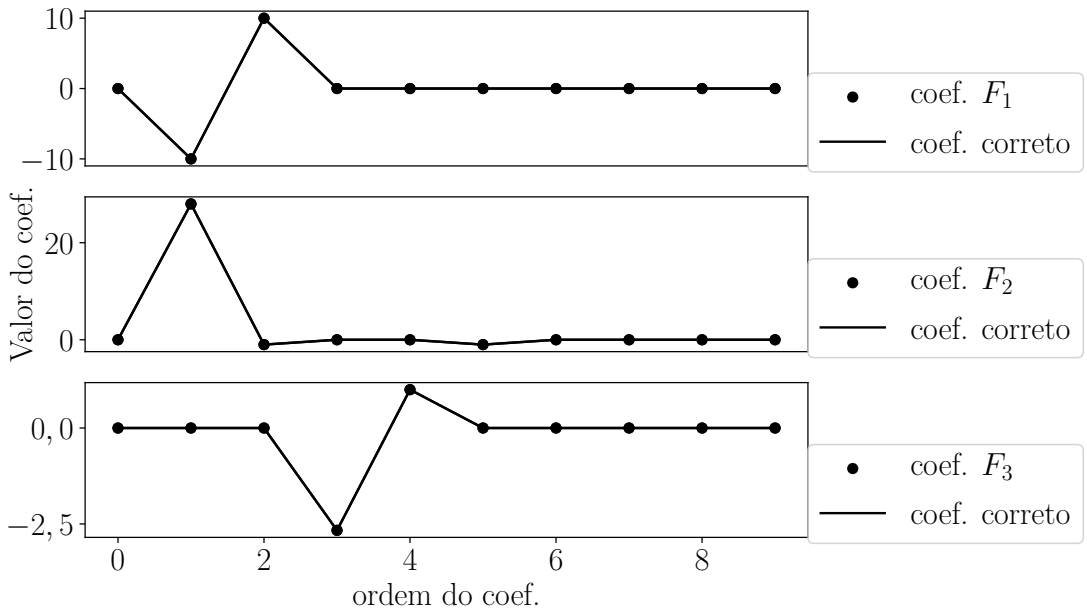


(b) **V** teste

Fonte: Elaborada pelo autor.

Vejamos na Figura 4.7 e na Tabela 4.1 os coeficientes obtidos para as funções polinomiais espaciais $\mathbf{F}(\mathbf{P})$. A implementação dessa etapa é baseada no manual SCIPY (2007). Esses coeficientes serão usados para as funções com o intuito de recuperar a dinâmica do sistema de Lorenz.

Figura 4.7: Comparação de coeficientes



Fonte: Elaborada pelo autor.

Apresentamos a Tabela 4.1 mostrando quais são esses valores.

Tabela 4.1: Valores resultantes para os coeficientes

equação / parcela	$\dot{x_1} = f_1$	$\dot{x_2} = f_2$	$\dot{x_3} = f_3$
1	0,0	0,0	0,0
x_1	-10,0	27,99	0,0
x_2	10,0	-0,99	0,0
x_3	0,0	0,0	-2,667
x_1x_2	0,0	0,0	0,99
x_1x_3	0,0	-0,99	0,0
x_2x_3	0,0	0,0	0,0
x_1^2	0,0	0,0	0,0
x_2^2	0,0	0,0	0,0
x_3^2	0,0	0,0	0,0

Fonte: Elaborada pelo autor.

A Tabela 4.2 apresenta a diferença entre os coeficientes originais e os aproximados.

Tabela 4.2: Diferença entre os coeficientes estimados e os coeficientes exatos

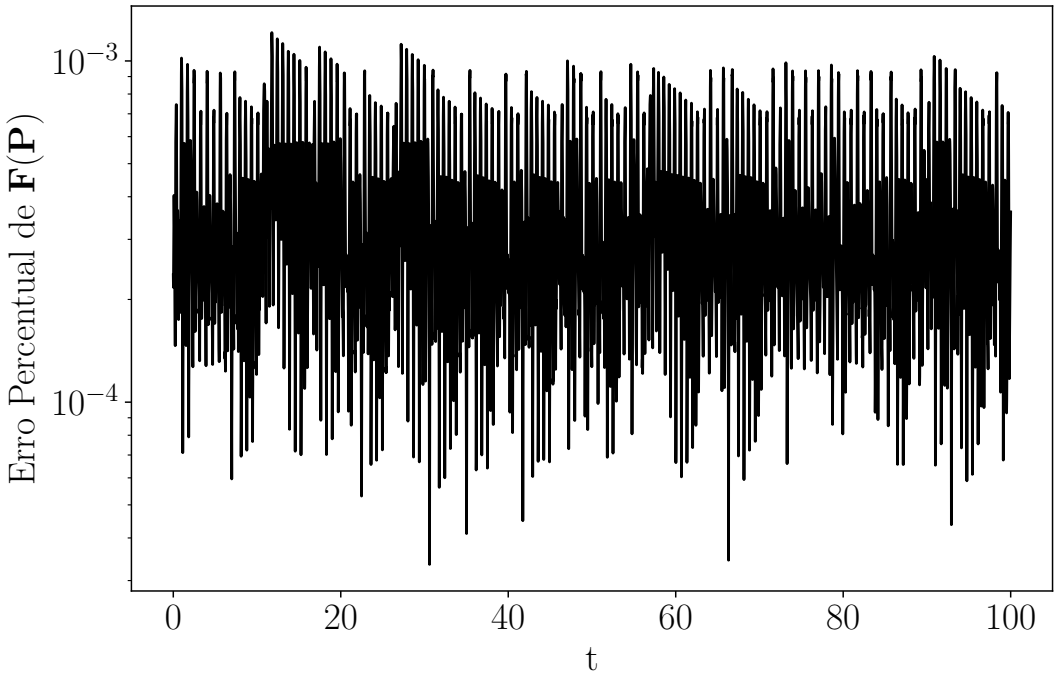
1	0,0	0,0	0,0
x_1	0,0	0,01	0,0
x_2	0,0	0,01	0,0
x_3	0,0	0,0	0,0
x_1x_2	0,0	0,0	0,01
x_1x_3	0,0	0,01	0,0
x_2x_3	0,0	0,0	0,0
x_1^2	0,0	0,0	0,0
x_2^2	0,0	0,0	0,0
x_3^2	0,0	0,0	0,0

Fonte: Elaborada pelo autor.

Daí, tendo as séries espaciais, vejamos a Figura 4.8 que mostra o erro percentual das funções polinomiais espaciais, isto é,

$$E_{\%}(\mathbf{f}-\mathbf{F}) = \frac{|f_1(\mathbf{x})-F1(\mathbf{P})|+|f_2(\mathbf{x})-F2(\mathbf{P})|+|f_3(\mathbf{x})-F3(\mathbf{P})|}{|f_1(\mathbf{x})|+|f_2(\mathbf{x})|+|f_3(\mathbf{x})|} \quad (4.6)$$

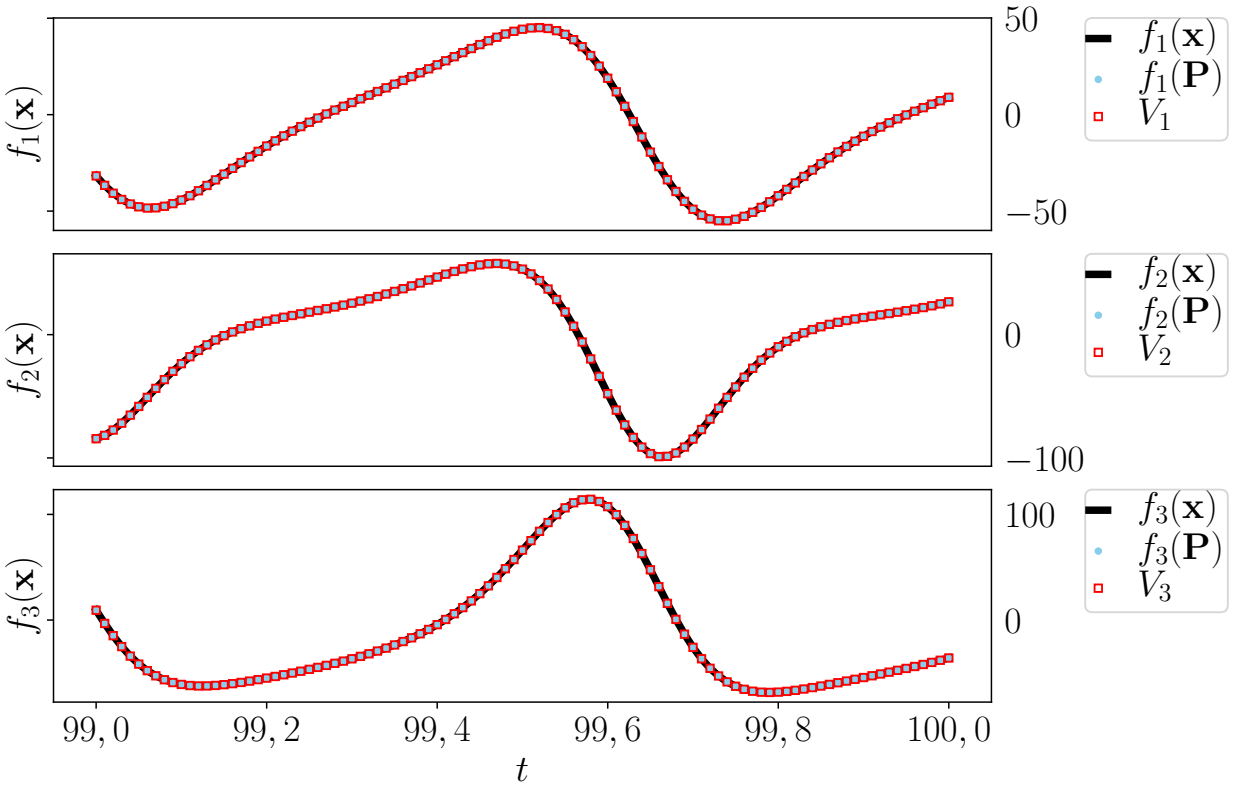
Figura 4.8: Erro percentual das funções polinomiais espaciais



Fonte: Elaborada pelo autor.

Para essa etapa é importante fazer a comparação entre trechos das funções polinomiais das velocidades $\mathbf{V}(t)$, trecho das trajetórias da funções polinomiais espaciais $\mathbf{F}(\mathbf{P})$ e trecho das variação espacial $\mathbf{f}(\mathbf{x})$. Queremos verificar se apresentam suavidade e semelhança de comportamento, isso com o pretexto de observar se em cada etapa do método, os parâmetros calculados são realmente ótimos. Isso pode ser visto na Figura 4.9.

Figura 4.9: Comparação entre as funções polinomiais de velocidade, espaciais e as varáveis espaciais originais

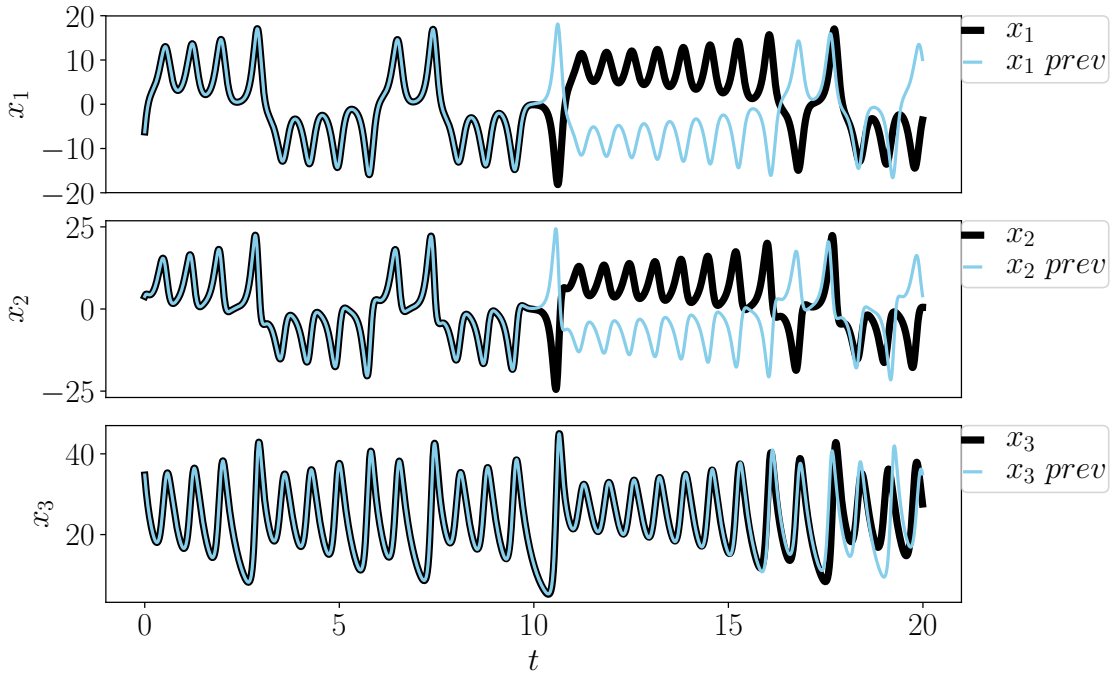


Fonte: Elaborada pelo autor.

Previsão

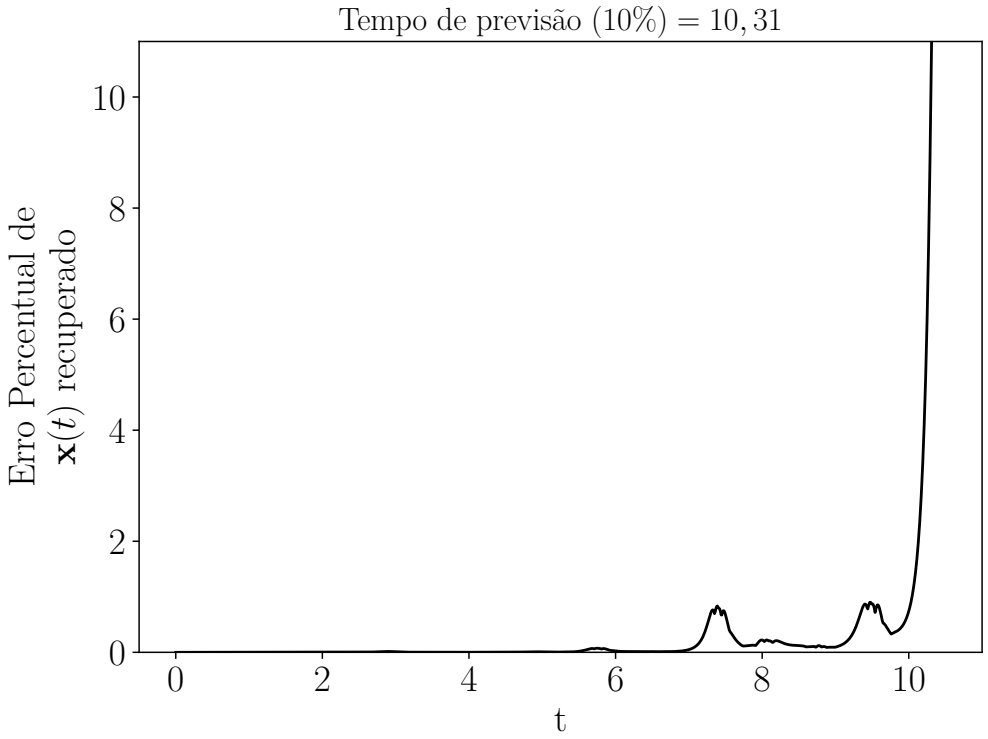
Nessa etapa, iremos usar os valores da Tabela 4.1 para formar o novo sistema de funções polinomiais e aplicar a Método de Runge-Kutta de quarta ordem (Apêndice A.1) à condição inicial igual a da série temporal dada na Figura 4.1. Assim iremos resultar as séries temporais aproximadas. Essas são as evoluções temporais do sistema e consideradas a nossa previsão. O resultado será comparado com as séries da Figura 4.1 e em seguida apresentamos o quão previsível é o nosso sistema aproximado usando a Equação 4.4. Nosso tempo de previsão é de 10,31 unidades de tempo, como podemos ver nas Figura 4.10 e Figura 4.11.

Figura 4.10: Comparação entre as séries temporais originais e recuperadas



Fonte: Elaborada pelo autor.

Figura 4.11: Erro percentual entre as séries temporais originais e recuperadas



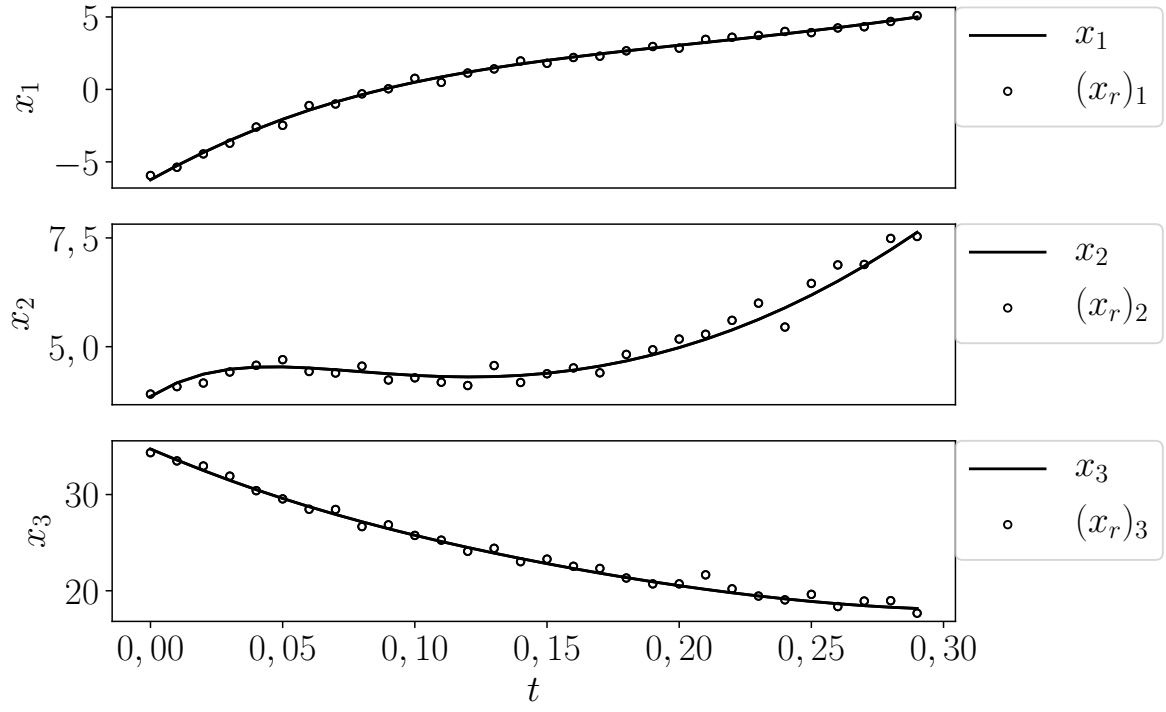
Fonte: Elaborada pelo autor.

4.1.3 Previsão do Sistema de Lorenz com Ruídos Observacionais

Na sessão anterior, obtemos os resultados de previsão usando os dados da Séries Temporais do Sistema de Lorenz (Figura 4.1), tal que foram considerados registros exatos. Dessa vez, iremos refazer o mesmo procedimento acrescentando algumas contaminações nesses dados, bem como mostrado na Seção 3.2. Sendo assim, iremos verificar se a metodologia ainda é eficaz nessa situação.

De inicio, acrescentaremos às séries temporais, valores aleatórios de uma distribuição normal $\mathbf{r} = (r_1, r_2, r_3, \ldots, r_m)$ com média $O = 0$ e desvio padrão $\mathbf{D} = (D_1, D_2, D3, \ldots, D_m)$, onde cada entrada é uma amplitude de ruído diferente tal que seus índices são respectivos aos índices das variáveis de posição, e assim, $\mathbf{D} = 0.01 \max(|\mathbf{x}(t)|)$. Logo, definiremos como $\mathbf{x_r} = \mathbf{x} + \mathbf{r}$. Vejamos na Figura 4.12.

Figura 4.12: Amostra de alguns dados com e sem ruídos



Fonte: Elaborada pelo autor.

Tendo os dados com ruídos, agora queremos além de encontrar coeficientes para funções polinomiais espaciais, o método também possa eliminar o ruído já no ajuste da função polinomial de posição, ou seja, funcionar como uma espécie de filtro. Naturalmente que, para sistemas desconhecidos não se tem valores corretos de comparação, pode ou não haver ruídos nos dados observados.

Para isso, as funções B-Splines, para a aproximação das funções polinomiais

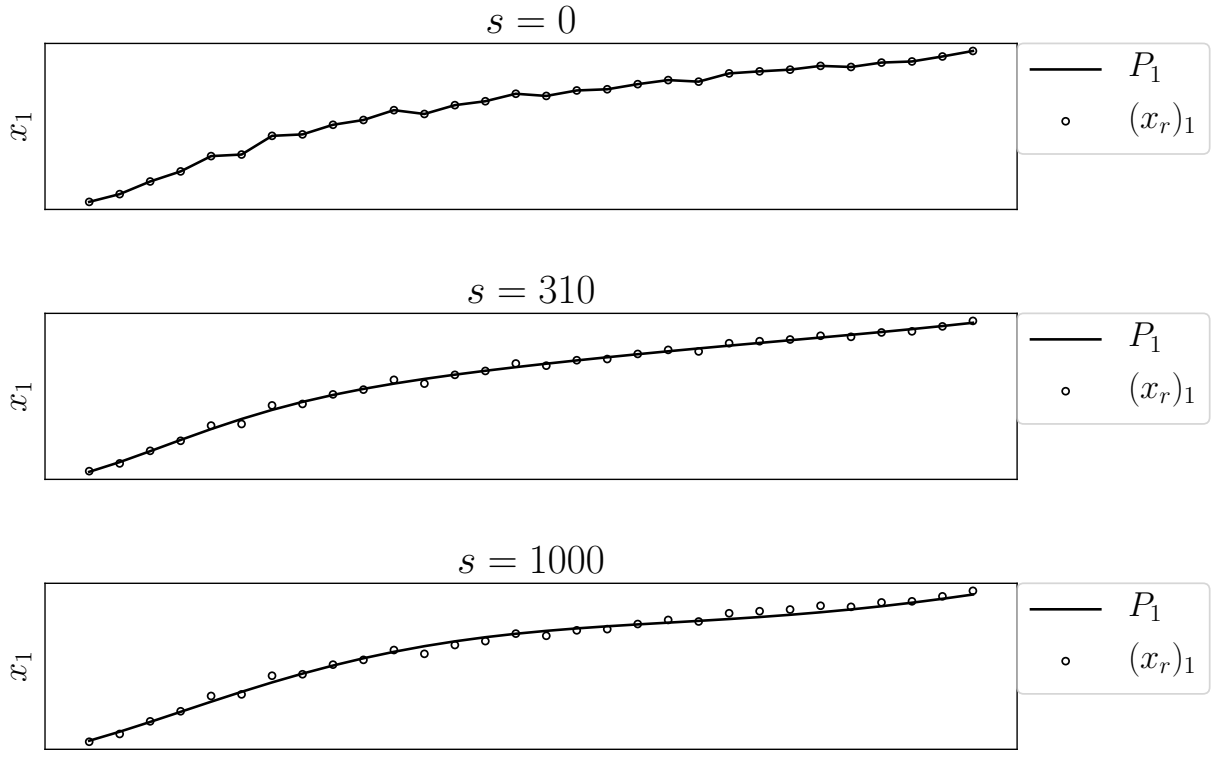
de posição, devem passar por um processo de ajuste por mínimos quadrados com restrição, isto é,

$$\sum(w_m[n](P_m[n] - x_m[n])) \leq s, \tag{4.7}$$

onde w_m são pesos, P_m é a função polinomial a ser ajustada, x_m são os dados que possuímos e s é um valor de restrição.

Como podemos ver, a Equação 4.7 é feita apenas com a norma l_1, mas s tem o mesmo papel de relevância aos dados. Se queremos interpolar, então $s = 0$, mas se não, então o ajuste deve ser $s > 0$. Quanto maior for s, maior é a suavidade, como podemos ver na Figura 4.13. Quanto aos pesos, nesse trabalho, iremos atribuir valores unitários, pois s já é um valor substancial.

Figura 4.13: Amostras de trechos da série temporal do sistema de Lorenz com o uso de alguns valores para a suavidade s



Fonte: Elaborada pelo autor.

Nessa circunstância, temos que analisar qual é o melhor valor de s. Nosso diagnóstico é baseado no comparativo com os dados exatos. Definimos $\mathbf{P_r}(t)$ como a função polinomial unidimensional aplicada aos dados com ruídos. Como vimos na seção anterior, se estamos interpolando, então temos que

$$\frac{1}{N}|\mathbf{P}(t) - \mathbf{x}(t)| = 0.$$

Mas para o caso em que os dados estão contaminados por ruídos, $\mathbf{P_r}(t)$ precisará passar por um processo de suavização de modo que, claramente, a melhor suavização é aquela que

$$\frac{1}{N}|\mathbf{P_r}(t) - \mathbf{x}(t)| \cong 0$$

e é menor que

$$\frac{1}{N}|\mathbf{P_r}(t) - \mathbf{x_r}(t)|.$$

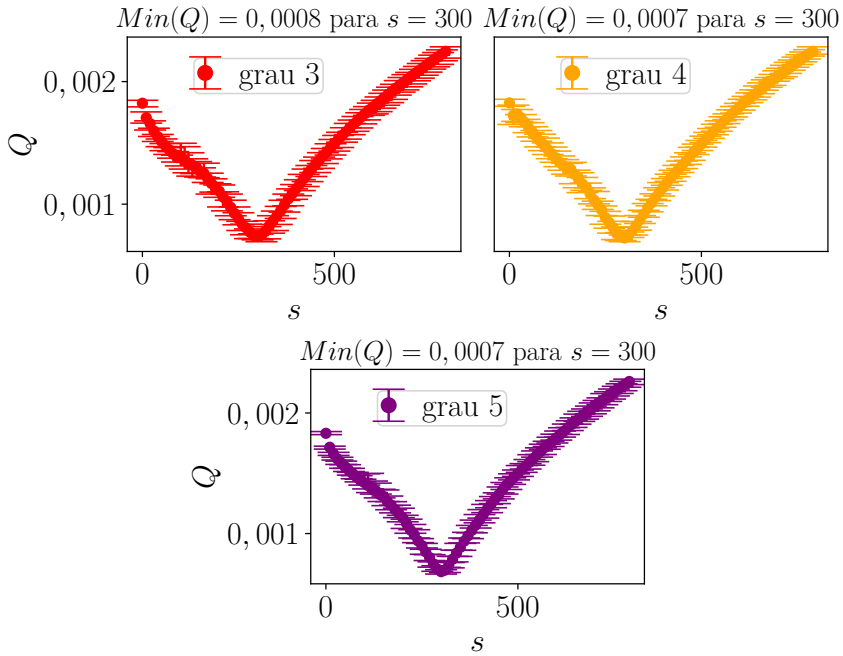
Sendo assim, realizamos vários testes variando o valor de s e definimos

$$\mathbf{Q} = \frac{1}{N}|\mathbf{P_r}(t) - \mathbf{x}(t)|,$$

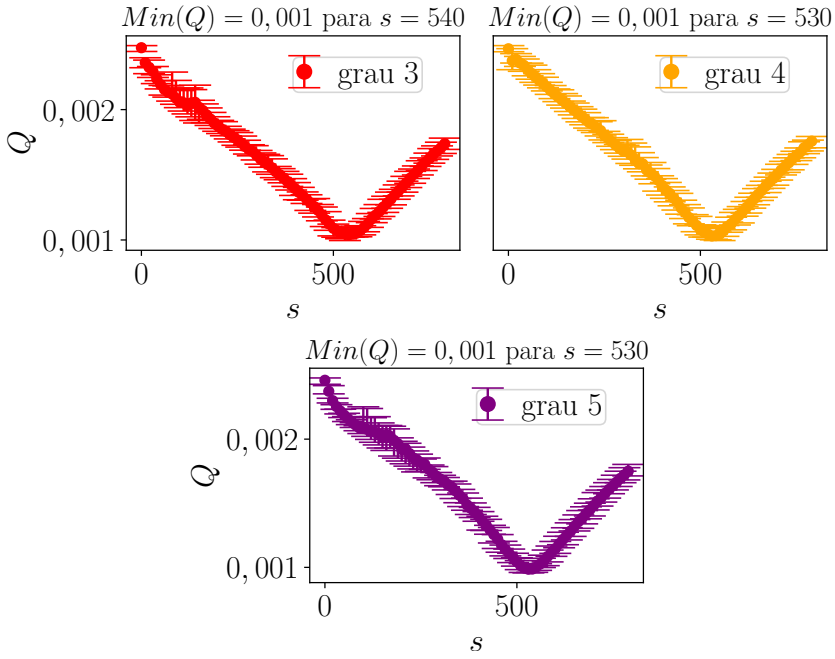
como podemos ver na Figura 4.14.

Figura 4.14: Analise da escolha do valor de s

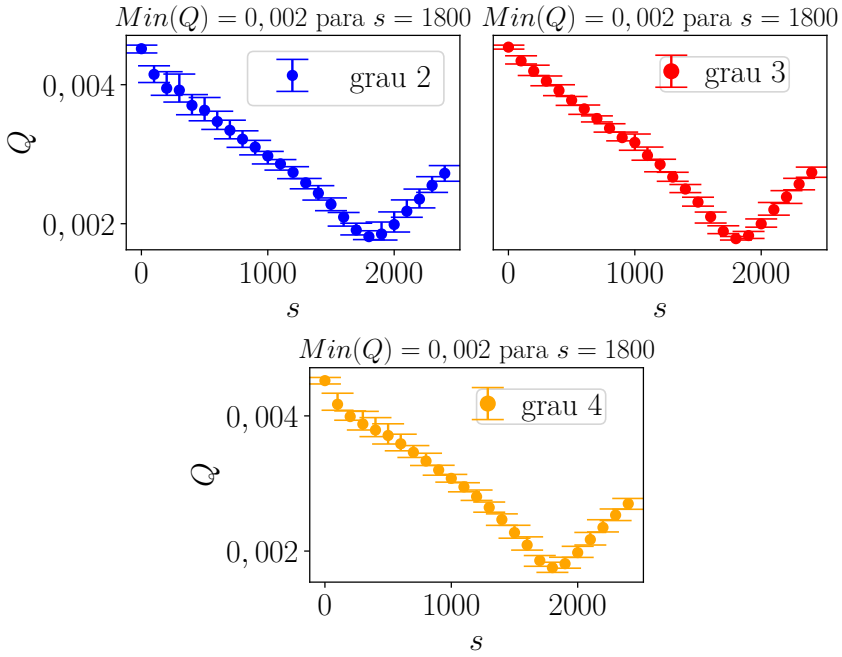
(a) $|(P_r)_1(t) - x_1(t)|$



(b) $|(P_r)_2(t) - x_2(t)|$



(c) $|(P_r)_3(t) - x_3(t)|$

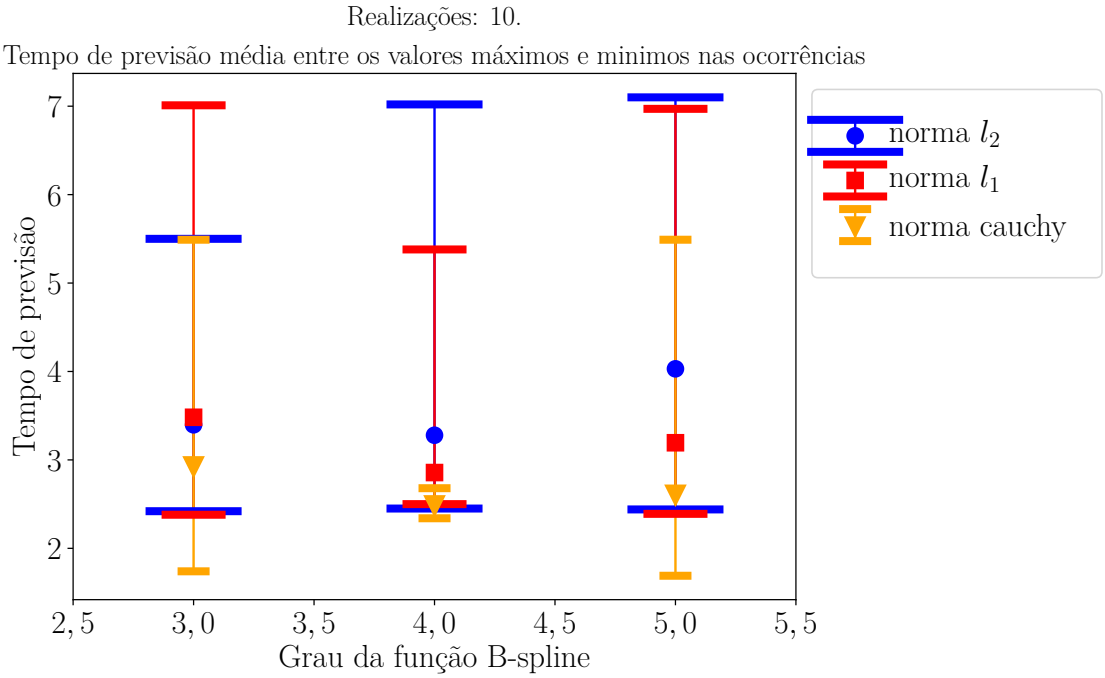


Fonte: Elaborada pelo autor.

As figuras mostram os melhores valores para suavização, dependendo do grau escolhido na função B-spline. Usaremos esses valores para os próximos resultados.

Da mesma forma que a seção anterior, é necessário realizar testes para verificar qual o melhor grau e norma para aplicar ao método. Vejamos na Figura 4.15, 10 realizações feitas com diferentes valores de ruídos.

Figura 4.15: Analise da escolha do grau e norma para o melhor tempo de previsão diante de séries com ruídos

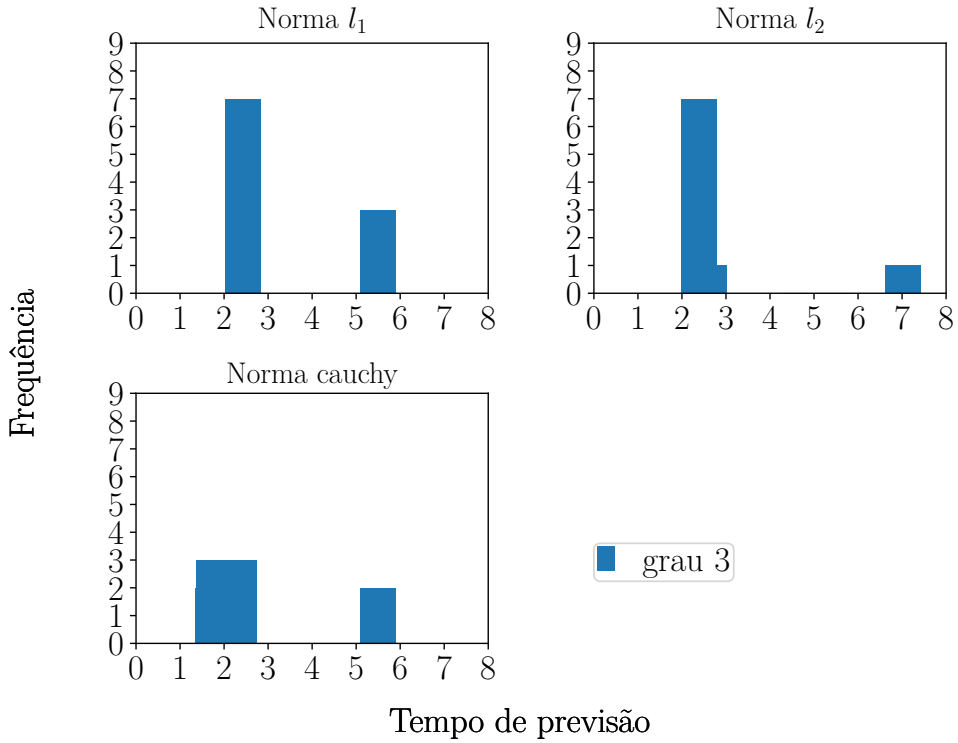


Fonte: Elaborada pelo autor.

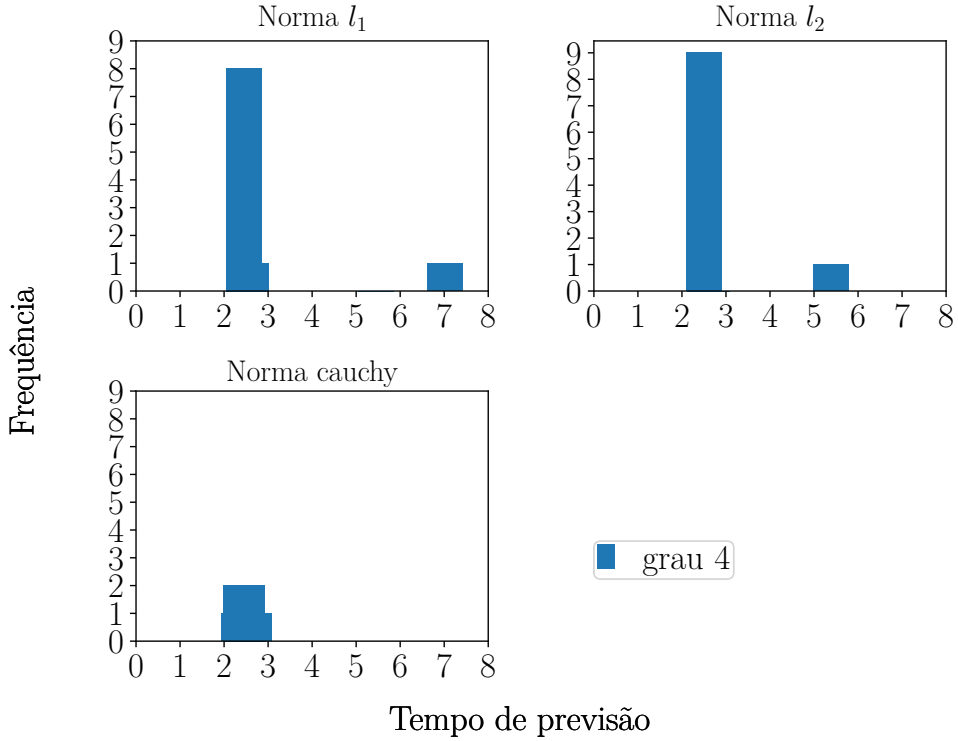
Durante essas realizações, podemos ver na Figura 4.16 a frequência em que os intervalos de tempo de previsão ocorrem.

Figura 4.16: Analise da escolha do valor de s

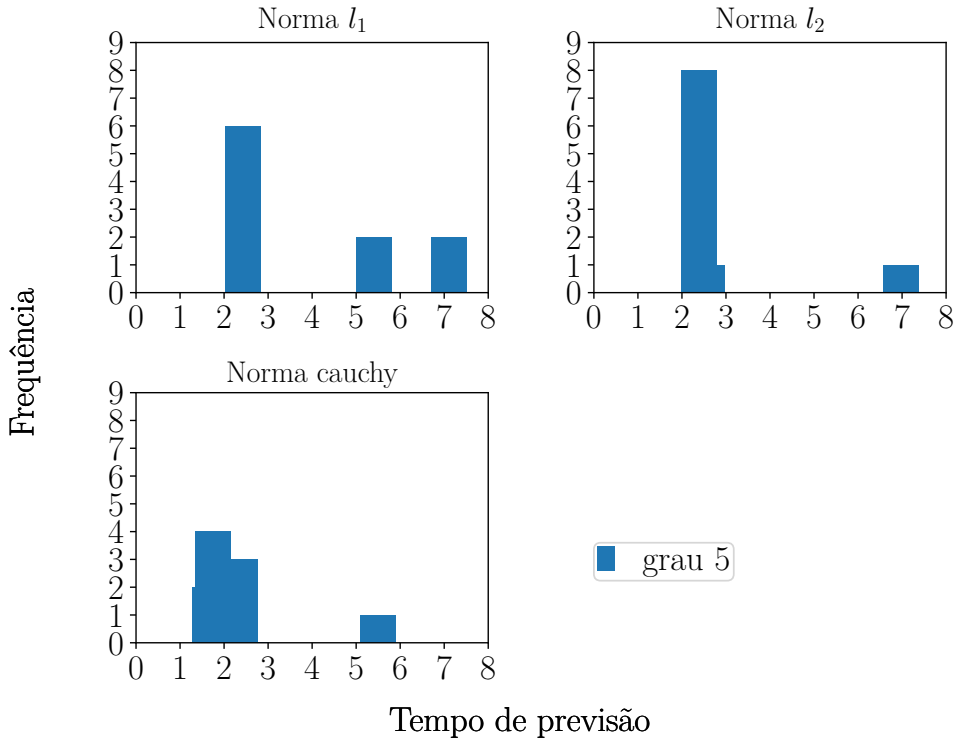
(a) Ocorrência com relação ao grau 3 e as normas



(b) Ocorrência com relação ao grau 4 e as normas



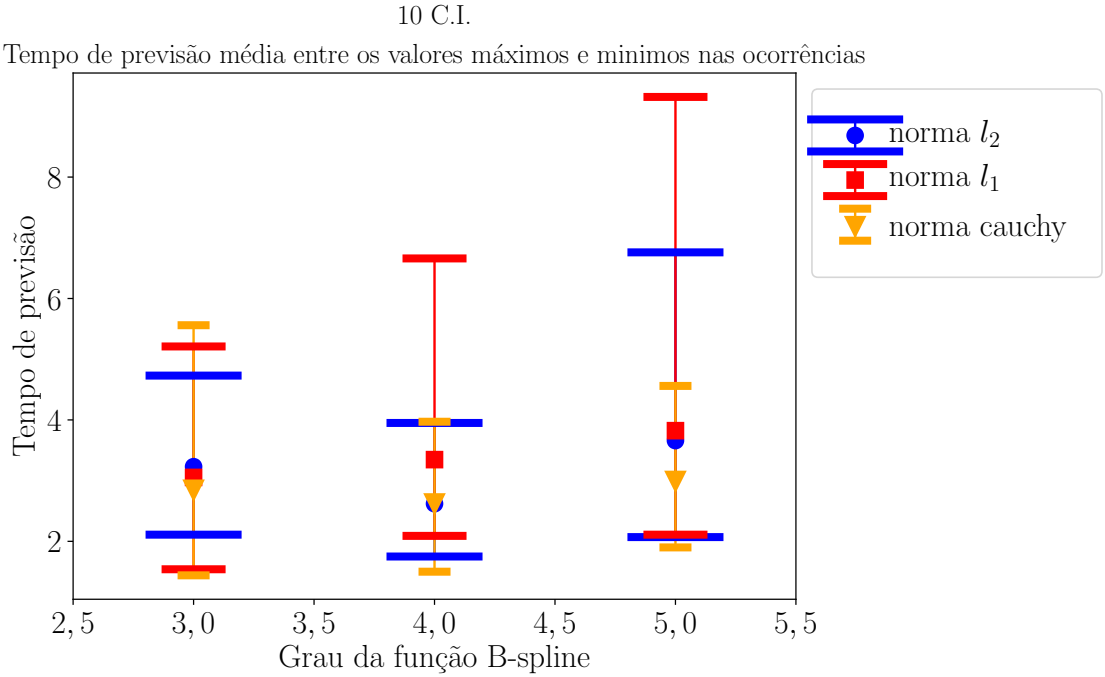
(c) Ocorrência com relação ao grau 5 e as normas



Fonte: Elaborada pelo autor.

Também, na Figura 4.17, apresentamos o resultado de 10 realizações para valores fixos de ruídos, mas com condições inicias diferentes, para o tempo de previsão médio e sua margem de erro com relação aos graus e as normas utilizados.

Figura 4.17: Análise da escolha do grau e norma para o melhor tempo de previsão diante de séries com ruídos com 10 diferentes condições iniciais.



Fonte: Elaborada pelo autor.

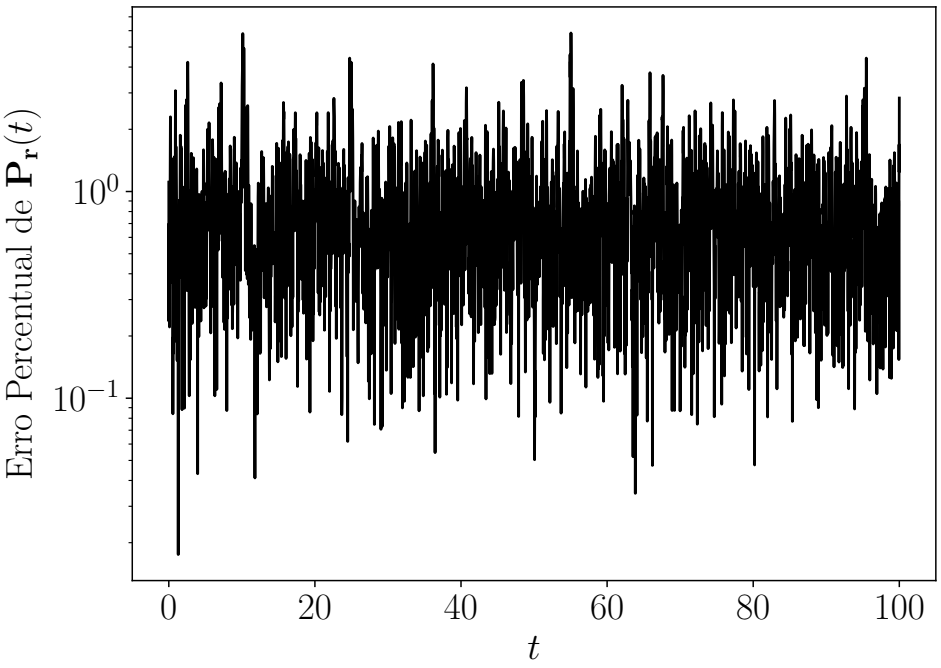
Esses resultados mostram que devemos usar B-spline de grau 5 para as funções polinomiais de posição e norma l_1 para a aproximação das funções polinomiais espaciais.

Evidentemente, que essa análise pode ser mudada dependendo do passo entre os dados temporais e o acréscimo do ruído, visto que nessa aplicação o nosso passo é $\Delta t = 0,01$ e o acréscimo é apenas 1% do $\max(\mathbf{x})$. Para outros dados, provavelmente os resultados de s, graus das funções e normas serão outros. Entretanto, a metodologia é a mesma, o que nos assegura sua eficácia.

Daí, usando grau 5, norma l_1, a Equação 4.5, mas sendo $\mathbf{P_r}(t)$ as funções polinomiais de posição suavizadas para os dados com ruídos, e fixando os valores de ruídos, apresentamos na Figura 4.18 o resultado para o erro percentual das funções polinomiais de posição com relação ao tempo.

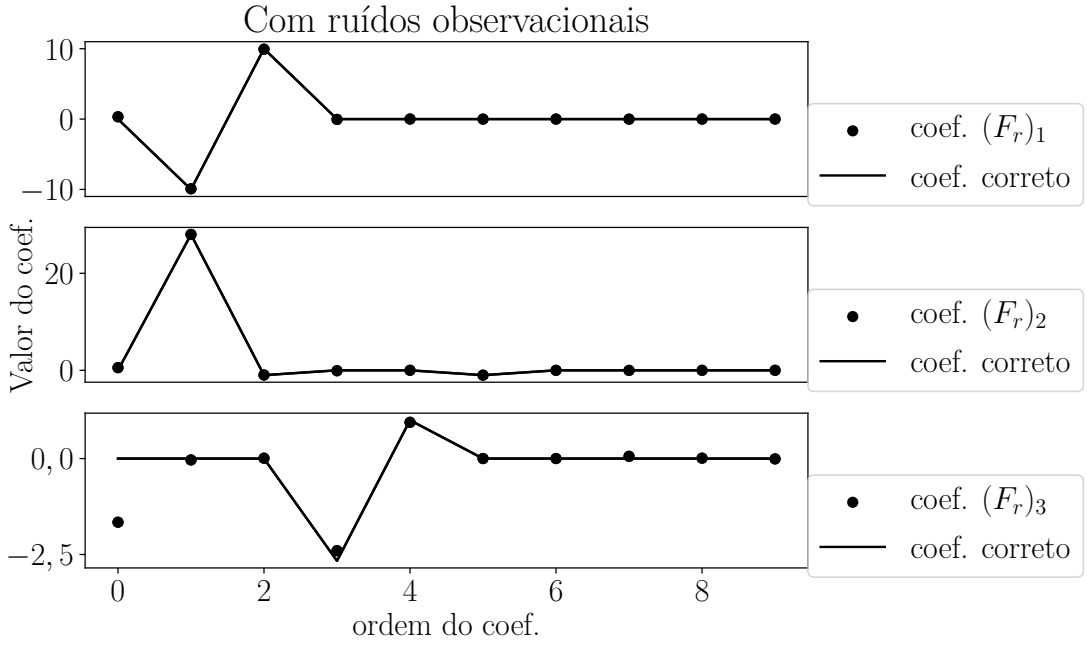
Logo em seguida, na Figura 4.19, na Figura 4.20, na Tabela 4.3 e na Tabela 4.4 apresentamos os resultados para os coeficientes das funções polinomiais espaciais.

Figura 4.18: Erro percentual das funções polinomiais de posição diante de séries com ruídos



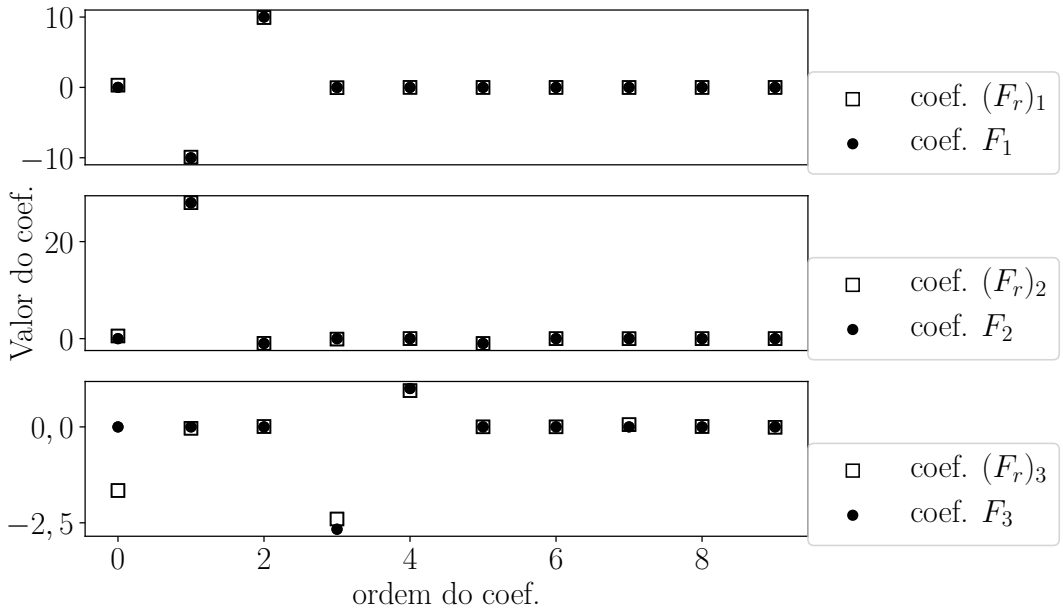
Fonte: Elaborada pelo autor.

Figura 4.19: Comparação de coeficientes diante de ruídos



Fonte: Elaborada pelo autor.

Figura 4.20: Efeito do ruído no cálculo dos coeficientes das funções polinomiais espaciais



Fonte: Elaborada pelo autor.

Tabela 4.3: Valores resultantes para os coeficientes diante de ruído

equação / parcela	$\dot{x_1} = f_1$	$\dot{x_2} = f_2$	$\dot{x_3} = f_3$
1	-1,0	-0,9	-1,7
x_1	-10,2	27,8	-0,5
x_2	10,0	-0,8	0,4
x_3	-0,2	0,0	-2,4
x_1x_2	0,0	0,0	1,0
x_1x_3	0,0	-0,9	0,0
x_2x_3	0,0	0,0	0,0
x_1^2	0,0	0,0	0,0
x_2^2	0,0	0,0	0,0
x_3^2	0,0	0,0	0,0

Fonte: Elaborada pelo autor.

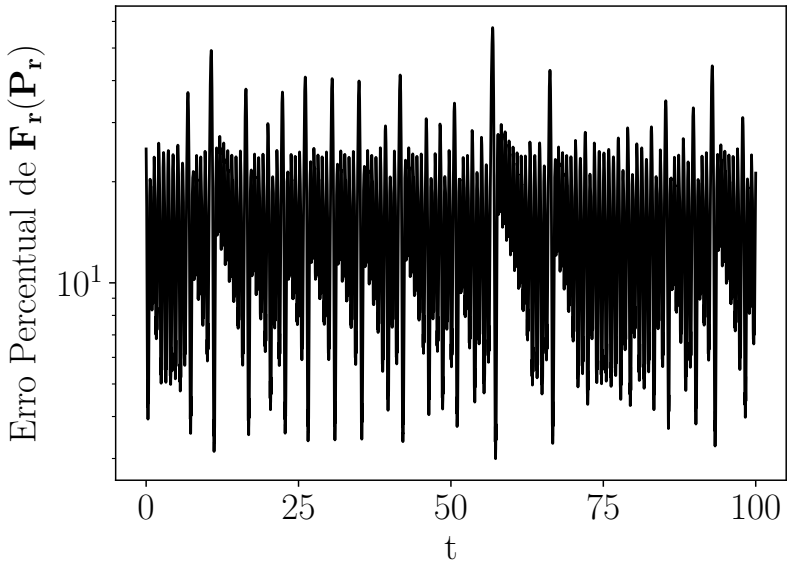
Tabela 4.4: Diferença entre os coeficientes estimados diante de ruído e os coeficientes exatos

1	1,0	0,8	0,4
x_1	0,2	0,2	0,5
x_2	0,0	0,2	0,4
x_3	0,2	0,0	0,006
x_1x_2	0,0	0,0	0,0
x_1x_3	0,0	0,0	0,0
x_2x_3	0,0	0,0	0,0
x_1^2	0,0	0,0	0,0
x_2^2	0,0	0,0	0,0
x_3^2	0,0	0,0	0,0

Fonte: Elaborada pelo autor.

A Figura 4.21 mostra o resultado para o erro percentual das funções polinomiais espaciais com relação ao tempo definido na Equação 4.6.

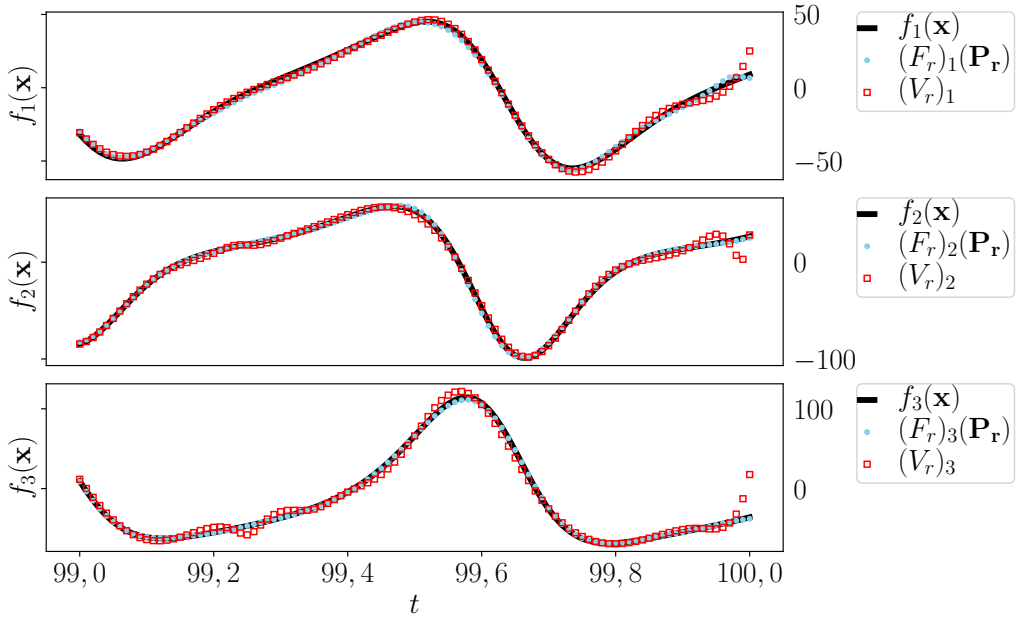
Figura 4.21: Erro percentual das funções polinomiais espaciais diante de ruídos



Fonte: Elaborada pelo autor.

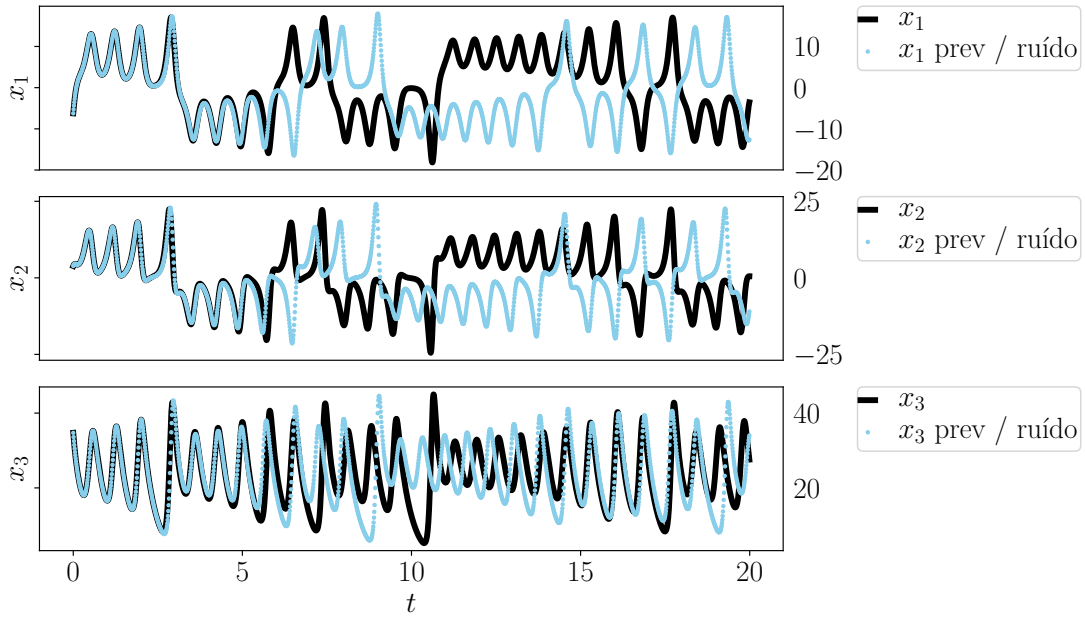
As Figura 4.22, Figura 4.23 e Figura 4.24 mostram o comparativo da funções polinomiais e o tempo de previsão.

Figura 4.22: Comparação entre as funções polinomiais de velocidade, espaciais diante de ruídos e as variáveis espaciais originais.



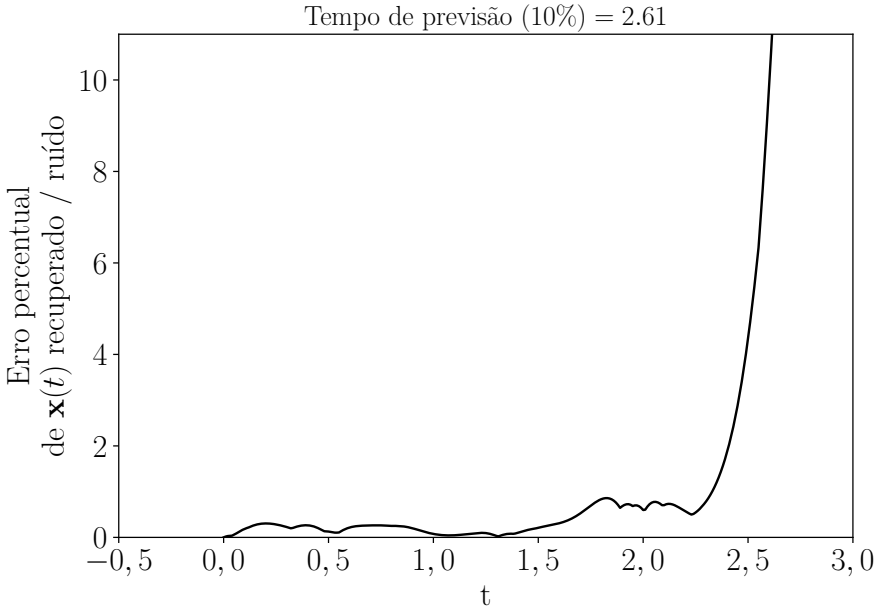
Fonte: Elaborada pelo autor.

Figura 4.23: Comparação entre as série temporais originais e recuperadas diante de ruídos



Fonte: Elaborada pelo autor.

Figura 4.24: Erro percentual entre as séries temporais originais e recuperadas diante de ruídos



Fonte: Elaborada pelo autor.

4.1.4 Previsão mediante reconstrução do atrator de Lorenz pelo Método das Coordenadas Atrasadas

Essa etapa consiste em reconstruir uma dinâmica possuindo apenas dados de uma série temporal, assim fazendo como no Método de Coordenadas Atrasadas (Seção 3.3). Em particular, trataremos do sistema de Lorenz a qual iremos usar apenas uma variável dinâmica para nossos testes, seja x_1 ou x_2 ou x_3, isto é, iremos tentar responder a possibilidade de construir uma dinâmica e realizar a previsão de uma série temporal. Escolhemos x_1 da Figura 4.1 sendo a série temporal conhecida e definimos

$$\begin{aligned} x(t) &= x_1(t), \\ x_\tau(t) &= x_1(t+\tau), \\ x_{2\tau}(t) &= x_1(t+2\tau), \end{aligned}$$

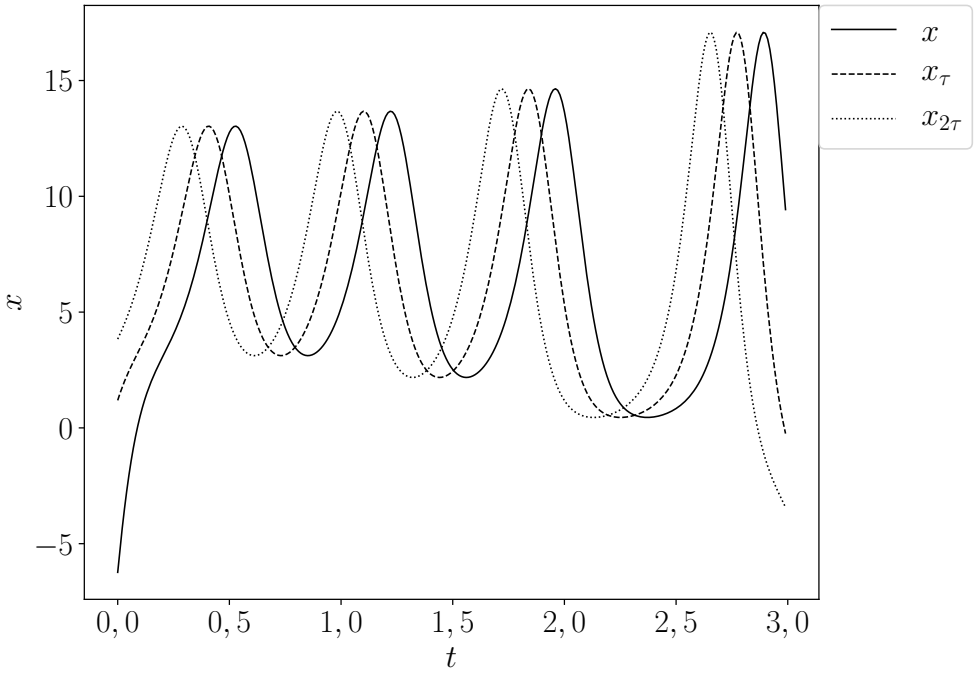
onde $\mu = 3$ é nossa dimensão de imersão e $\tau = 0,12$ é nosso atraso temporal como visto na Seção 3.3. Entretanto como estamos trabalhando com dados discretos, é preciso converter esse atraso temporal contínuo em um atraso de índices das séries

a qual definimos

$$\begin{aligned} x[n] &= (x_1[0], x_1[1], ..., x_1[10001 - 2\alpha]), \\ x_\tau[n] &= (x_1[\alpha], x_1[\alpha + 1], ..., x_1[10001 - \alpha]), \\ x_{2\tau}[n] &= (x_1[2\alpha], x_1[2\alpha + 1], ..., x_1[10001]), \end{aligned}$$

onde $\alpha = \text{int}\left(\frac{\tau}{\Delta t}\right)$. Além disso, para que as séries tenham a mesma cardinalidade, essa definição diminui os dados para $N = 9977$. Isso é mostrado na Figura 4.25.

Figura 4.25: Trecho das séries temporais por coordenadas atrasadas a partir de uma série temporal do sistema de Lorenz



Fonte: Elaborada pelo autor.

Com isso, iremos usar a metodologia da Seção 3.4 para construir um sistema, onde x_τ e $x_{2\tau}$ fazem o papel, respectivamente, de x_2 e x_3, e que dependa dessas três séries, mas nosso interesse está apenas na previsão da série conhecida.

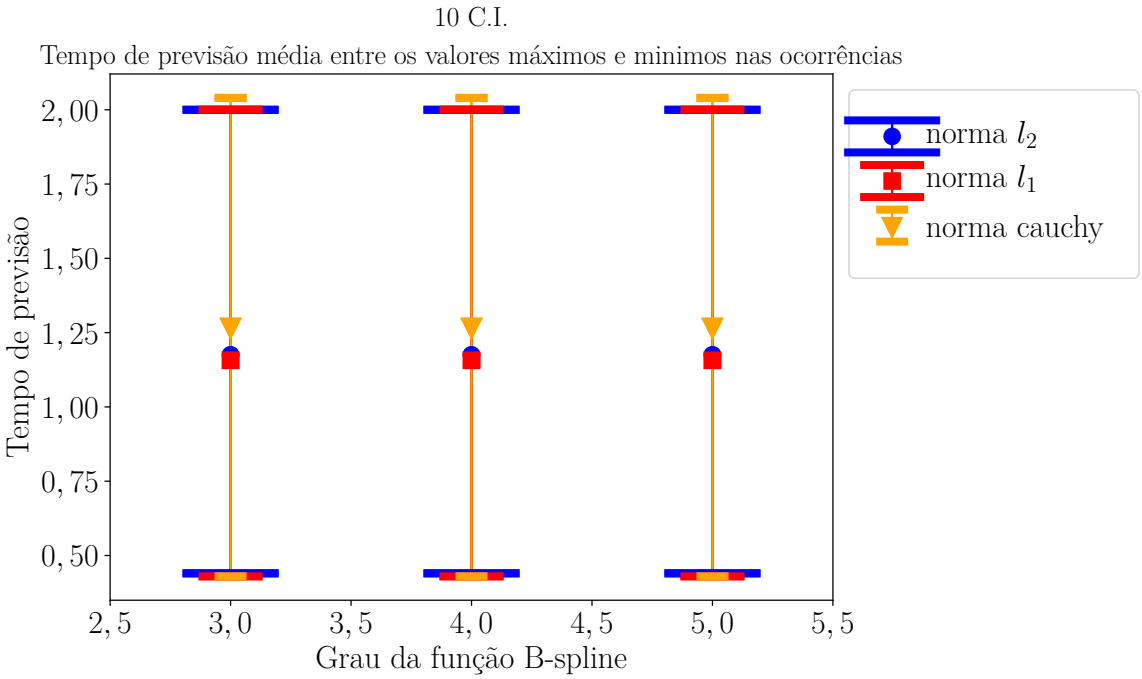
Da mesma forma, temos que verificar o grau e norma em diferentes condições iniciais, a fim de encontrar o melhor resultado de previsão, onde o erro percentual é definido por

$$E_{\%}(x - x_{prev}) = 100\frac{|x - x_{prev}|}{\max|x|}.$$

Além disso, essas séries irão produzir uma dinâmica desconhecida de modo que precisamos aumentar o grau da função espacial para alavancar a aproximação, pois

um teste rápido, percebemos que usando essas funções em segundo grau não temos uma boa aproximação. Sendo assim, nessa etapa usaremos funções polinomiais multidimensionais de terceiro grau. Veremos na Figura 4.26.

Figura 4.26: Analise da escolha do grau e norma para o melhor tempo de previsão adiante de séries por coordenadas atrasadas



Fonte: Elaborada pelo autor.

Como pode-se ver, a escolha pode ser qualquer grau da função B-Spline e norma l_1 para aproximação das funções espaciais. Vamos apresentar os resultados quando usamos o grau 5. A Tabela 4.5 mostra o resultado dos valores dos coeficientes.

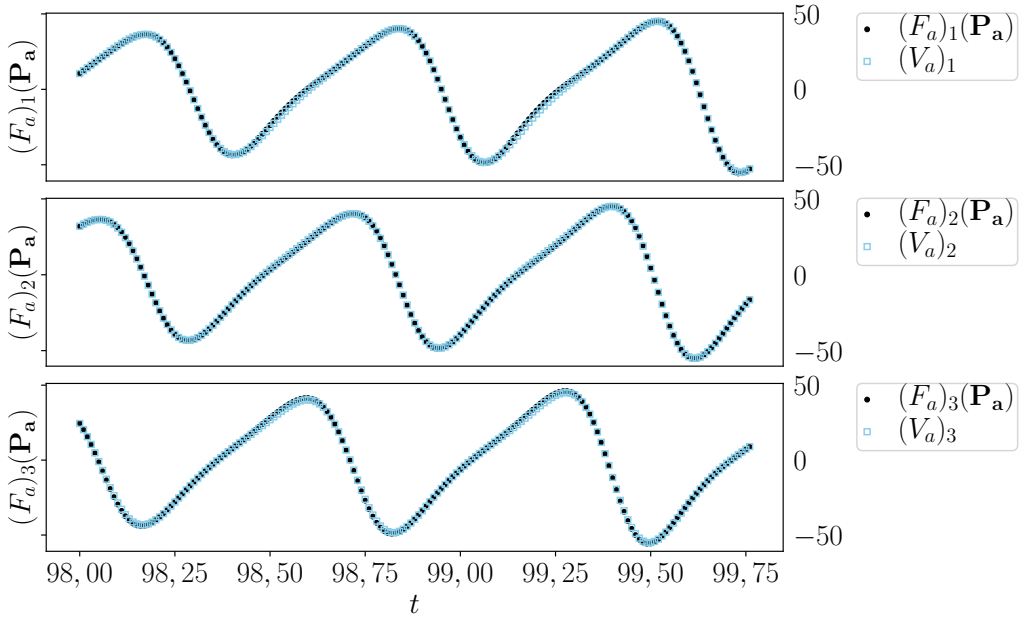
Na Figura 4.27 veremos o resultado da aproximação das funções polinomiais espaciais e da velocidade.

Tabela 4.5: Valores resultantes para os coeficientes do sistema construído

equação / parcela	$\dot{x_1} = f_1$	$\dot{x_2} = f_2$	$\dot{x_3} = f_3$
1	-0,1	0,0	0,0
x	-16,3	-2,5	1,8
x_τ	22,6	-2,3	-15,2
$x_{2\tau}$	-4,8	4,1	14,2
xx_τ	0,0	0,0	0,0
$xx_{2\tau}$	0,0	0,0	-0,1
$x_\tau x_{2\tau}$	0,0	0,0	0,1
$xx_\tau x_{2\tau}$	0,0	0,0	0,0
x^2	0,0	0,0	0,0
x_τ^2	0,0	0,0	0,0
$x_{2\tau}^2$	0,0	0,0	0,0
$x^2 x_\tau$	-0,1	0,0	0,0
$x^2 x_{2\tau}$	0,0	0,0	0,0
xx_τ^2	-0,1	-0,1	0,1
$xx_{2\tau}^2$	0,0	0,0	0,0
$x_\tau^2 x_{2\tau}$	-0,1	0,0	0,0
$x_{2\tau}^2 x_\tau$	0,0	0,0	-0,1
x^3	-0,1	0,0	0,0
x_τ^3	-0,1	0,0	0,0
$x_{2\tau}^3$	0,0	0,0	0,0

Fonte: Elaborada pelo autor.

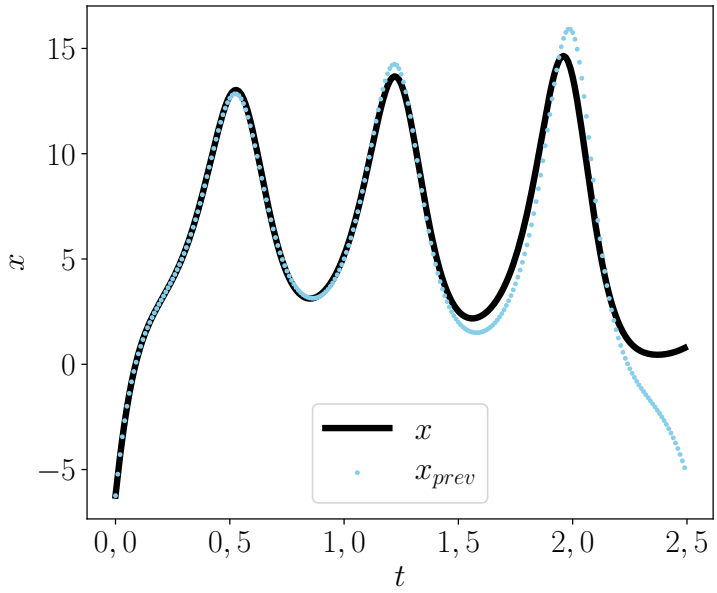
Figura 4.27: Comparação entre as funções polinomiais de velocidade, espaciais com base na reconstrução e as varáveis espaciais originais



Fonte: Elaborada pelo autor.

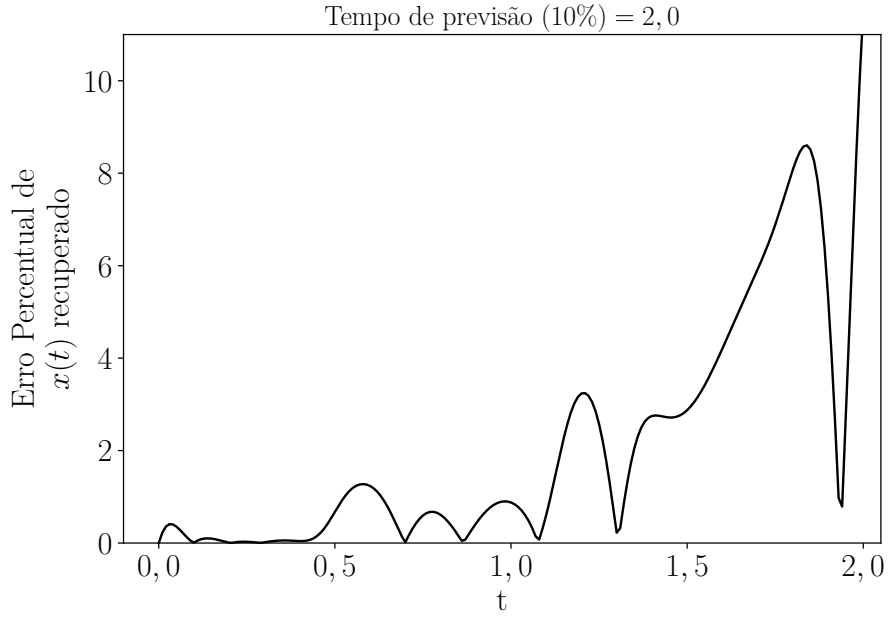
Assim, as Figura 4.28 e Figura 4.29 mostram a reconstrução da série e o tempo de previsão.

Figura 4.28: Comparação entre as série temporal original e a recuperada com base na reconstrução



Fonte: Elaborada pelo autor.

Figura 4.29: Erro percentual entre as variáveis de posição originais e recuperadas com base na reconstrução



(a) Total

Fonte: Elaborada pelo autor.

Logo, temos um tempo de previsão de 2,0 unidades de tempo. Não é um resultado tão bom em comparativo o tempo de previsão mostrado na Figura 4.2, mas mostra a eficiência do método.

Capítulo 5

Conclusões

Para tentar reproduzir a dinâmica de sistemas cujas equações não são conhecidas, começamos estudando a recuperação de um sistema conhecido. Usamos o sistema de Lorenz como exemplo para desenvolver o nosso método. Mostramos que o conhecimento de séries temporais de dados oriundos do sistema pode ser usado para recuperar a dinâmica que origina o comportamento observado. Em seguida, usando a dinâmica recuperada, abordamos o problema da previsão do estado futuro.

Desenvolvemos uma técnica de inteligência artificial utilizando funções polinomiais para calcular as velocidades de evolução das variáveis de estado do sistema e com isso, recuperamos sua dinâmica para, em seguida, realizar a previsão. O algoritmo proposto é capaz de se adaptar ao sistema, com parâmetros flexíveis, extraídos a partir dos dados coletados em observações. Sendo assim, os resultados são eficazes diante de um sistema caótico e o método ainda pode ser ajustado e aplicado a outros sistemas. Exploramos várias situações, no intuito de mostrar que sua aplicabilidade a problemas reais é viável.

Nossos resultados mostraram que as funções B-Spline de grau 5 são boas em todos os casos estudados nesse trabalho. As implementações dessas funções usando o manual do SCIPY (2007) foram de grande ajuda, pois potencializou o processo de ajuste por polinômios.

Em nosso primeiro caso, o método praticamente recuperou a dinâmica do sistema. Os resultados dos coeficientes ficaram muitos próximos, chegando a acertar em sua maioria, mas pelo fato do sistema ser caótico, já era de se esperar que isso acarretasse em um erro exponencial em que conseguimos obter um tempo de previsão em torno de 10 unidades de tempo.

No segundo caso, tivemos ruído aditivos, o que tornou a aplicação do método mais difícil. Para reduzir o efeito do ruído, fizemos testes com diferentes valores de um parâmetro de suavização e conseguimos encontrar valores que fazem uma filtragem nos dados com ruídos de modo a aproximá-los cada vez mais dos dados considerados exatos. Devido a isso, conseguimos obter um tempo de previsão em perto de 3

unidades de tempo. Evidentemente que em situações reais, onde não conhecemos o sistema, a tarefa de encontrar o valor de suavização torna-se complicada. Mas em compensação, o uso de B-Spline ajustada possibilita flexionar a curva de modo a atender nosso interesse. Sendo assim, podemos analisar qualitativamente e julgar valores ótimos para a suavização. Todavia, em nossos testes, também é possível recuperar a dinâmica se fizermos interpolação nos dados, mas em comparativo o resultado não é bom.

No terceiro caso, tivemos um problema mais complexo. Consideramos conhecida a série temporal de apenas uma das variáveis do sistema de Lorenz. Com o uso do método de coordenadas atrasadas, formamos outras séries com o objetivo de construir um sistema de múltiplas variáveis. Não objetivamos recuperar o sistema de Lorenz, mas fazer a construção de uma dinâmica que permita fazer a previsão da série temporal conhecida. Funções polinomiais multidimensionais com grau menores que 3 não deram boas aproximações, então tivemos que optar em usar funções de terceiro grau, consequentemente usamos muitos coeficientes, o que tornou mais trabalhoso. Tivemos o resultado de previsão em torno 2 unidades de tempo. Esses resultados mostram a eficiência do método. Do ponto de vista de eficácia, existem metodologias que podem apresentar resultados mais consistentes com o problema, é o caso de se trabalhar com a adaptação do método de solução numérica para equações diferencias com retardamento. Por sua vez, demandaria um pouco mais de aprofundamento, mas deixo como proposta para trabalhos futuros.

Em geral, conclui-se que as aproximações por funções polinomiais são eficazes para recuperar as equações governantes dos sistemas. Logo, conforme temos os métodos para a previsão de um sistemas caótico, então é possível sua aplicabilidade a dados reais de sistemas desconhecidos.

Referências Bibliográficas

ARAÚJO, D. S., 2016, *Curvas B-Spline e de Bézier Aplicadas a CADG*. Tese de Mestrado, Instituto Nacional de Matemática Pura e Aplicada, Rio de Janeiro.

BOEING, G., 2016, "Visual Analysis of Nonlinear Dynamical Systems: Chaos, Fractals, Self-Similarity and the Limits of Prediction", *Systems*, v. 4 (11), pp. 37. doi: 10.3390/systems4040037.

BUENO, E. I., PEREIRA, I. M., SILVA, A. T., 2016, *GMDH-BASED POLYNOMIAL NEURAL NETWORK ALGORITHM IN MATLAB*. Imperial College Press. ISBN: 978-1-78326-612-8. doi: 10.1142/9781783266135_0004.

CAMPANHARO, S. L. O., MACAU, E. E. N., RAMOS, F. M., 2014, "Detectando a presença de caos em uma série temporal." *Sociedade Brasileira de Matemática Aplicada e Computacional*. Disponível em: <`http://www.sbmac.org.br/eventos/cnmac/cd_xxviii_cnmac/resumos%20estendidos/andriana_campanharo_ST21.pdf`>. Acessado: 2019-05-28.

CAMPOS, R. J., 2008, *Previsão de séries temporais com aplicações a séries de consumo de energia elétrica*. Tese de Mestrado, Universidade Federal de Minas Gerais, Belo Horizonte.

CORDEIRO JR., O. A., 2007, *Utilizando Séries Temporais na Previsão da Arrecadação do Imposto de Renda*. Relatório técnico, ISC / CENTRO DE FORMAÇÃO, TREINAMENTO E APERFEIÇOAMENTO - CEFOR, Distrito Federal.

COSTA, A., OLIVEIRA, V., PEREIRA, A., 2015, "ESTUDO DO CLIMA NA REGIÃO DO BREJO PARAIBANO UTILIZANDO TÉCNICAS DE SÉRIES TEMPORAIS, PARA PREVISÃO COM O MODELO SARIMA", *GAIA SCIENTIA*, v. 9, n. 1, pp. 127–133.

DE S. CAVALCANTE, H. L. D., ORIÁ, M., SORNETTE, 2013, "Predictability and Suppression of Extreme Events in a Chaotic System", *Phys. Rev. Lett.*, v. 111 (Nov), pp. 198701. doi: 10.1103/PhysRevLett.111.198701.

EHLERS, R., 2005, "Análise de Séries Temporais", *Departamento de Estatística, UFPR*, (Nov). Disponível em: <`http://www.each.usp.br/rvicente/AnaliseDeSeriesTemporais.pdf`>. Acessado: 2019-05-18.

ELLACOTT, S., MASON, J., ANDERSON, I., 2012, *Mathematics of Neural Networks: Models, Algorithms and Applications*. Operations Research/-Computer Science Interfaces Series. Springer US. ISBN: 9781461560999.

FAWCETT, T., PROVOST, F., 2018, *Data Science para negócios: O que você precisa saber sobre mineração de dados e pensamento analítico de dados*. Alta Books. ISBN: 9788550803906.

FITZPATRICK, R., 2013, "Oscillations and Waves", Acessado: 2019-05-18. Disponível em: <`http://farside.ph.utexas.edu/teaching/315/Waves/node13.html`>.

FONTES, N. R. M., 2013, *SISTEMAS DINÂMICOS, ANÁLISE NUMÉRICA DE SÉRIES TEMPORAIS E APLICAÇÕES ÀS FINANÇAS*. Tese de Mestrado, Universidade de Lisboa, Lisboa.

FRANCO, N., 2006, *Cálculo numérico*. PRENTICE HALL BRASIL. ISBN: 9788576050872.

GRUS, J., 2018, *Data Science do Zero: Primeiras regras com o Python*. Alta Books. ISBN: 9788550803876.

KANTZ, H., SCHREIBER, T., 2004, *Nonlinear Time Series Analysis*. Cambridge nonlinear science series. Cambridge University Press. ISBN: 9780521529020.

LU, Z., HUNT, B. R., OTT, E., 2018, "Attractor reconstruction by machine learning", *Chaos: An Interdisciplinary Journal of Nonlinear Science*, v. 28, n. 6, pp. 061104. doi: 10.1063/1.5039508.

MACHADO, K. F., 2003, *Módulo de auto-localização para um agente exploratório usando Filtro de Kalman*. Tese de Mestrado, Universidade Federal do Rio Grande do Sul.

OLIVEIRA JR., G. F. D., 2012, *Caos e sincronização em circuitos eletrônicos com realimentação atrasada*. Tese de Mestrado, Universidade Federal da Paraíba, João Pessoa.

PATHAK, J., HUNT, B., GIRVAN, M., 2018, "Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach", *Physical Review Letters*, v. 120 (01). doi: 10.1103/PhysRevLett.120.024102.

PECORA, L. M., MONIZ, L., NICHOLS, J., 2007, "A unified approach to attractor reconstruction", *Chaos: An Interdisciplinary Journal of Nonlinear Science*, v. 17, n. 1, pp. 013110. doi: 10.1063/1.2430294.

PEREIRA, L. R., 2014, *Ajuste de curva B-spline fechada com peso*. Tese de Mestrado, Universidade Federal de Uberlândia, Uberlândia.

POLLOCK, D., POLLOCK, D., GREEN, R., 1999, *A Handbook of Time-series Analysis, Signal Processing and Dynamics*. Signal processing and its applications. Academic. ISBN: 9780125609906.

PRESS, W., TEUKOLSKY, S., VETTERLING, W., 2007, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press. ISBN: 9780521880688.

RUGGIERO, M., DA ROCHA LOPES, V., 1996, *Cálculo numérico: aspectos teóricos e computacionais*. Pearson Makron Books. ISBN: 9788534602044.

SCHOPF, E. C., 2007, *MÉTODO NEURO-ESTATÍSTICO PARA PREDIÇÃO DE SÉRIES TEMPORAIS RUIDOSAS*. Tese de Mestrado, Universidade Federal do Rio Grande do Sul.

SCIPY, 2007, "scipy optimize least squares", Acessado: 2019-05-30. Disponível em: <`https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least_squares.html#scipy.optimize.least_squares`>.

STROGATZ, S., 1994, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Studies in Nonlinearity. PERSEUS BOOKS.

TRAN, G., WARD, R., 2016, "Exact Recovery of Chaotic Systems from Highly Corrupted Data", *Cornell Univerty*.

WADE, W., 2013, *Introduction to Analysis: Pearson New International Edition*. Pearson Education Limited. ISBN: 9781292055893.

WANG, W.-X., LAI, Y.-C., GREBOGI, C., 2016, "Data based identification and prediction of nonlinear and complex dynamical systems", *Physics Reports*, v. 644, pp. 1 – 76. ISSN: 0370-1573. doi: https://doi.org/10.1016/j.

physrep.2016.06.004. Data based identification and prediction of nonlinear and complex dynamical systems.

YANG, R., LAI, Y.-C., GREBOGI, C., 2012, "Forecasting the future: Is it possible for adiabatically time-varying nonlinear dynamical systems?" *Chaos: An Interdisciplinary Journal of Nonlinear Science*, v. 22, n. 3, pp. 033119. doi: 10.1063/1.4740057.

Apêndice A

Algumas Demonstrações

A.1 Método de Runge-Kutta de quarta ordem

Na aproximação:

$$x[n+1] = x[n] + \Delta t d,$$

onde Δt é o tamanho do passo e d será o valor da derivada obtida pela média das derivadas em 4 pontos diferentes e com pesos diferentes

Começa-se por calcular a derivada no ponto inicial, tal como no método de Euler:

$$d_1 = f(x[0], t[0]).$$

A seguir, realiza-se um deslocamento na direção dessa derivada, avançando uma distancia $\frac{\Delta t}{2}$ no tempo, ate um ponto 1 (Ver figura).

Nesse ponto 1, calcula-se um segundo valor da derivada:

$$d_2 = f\Big(x[0] + \frac{\Delta t d_1}{2}, t[0] + \frac{\Delta t}{2}\Big).$$

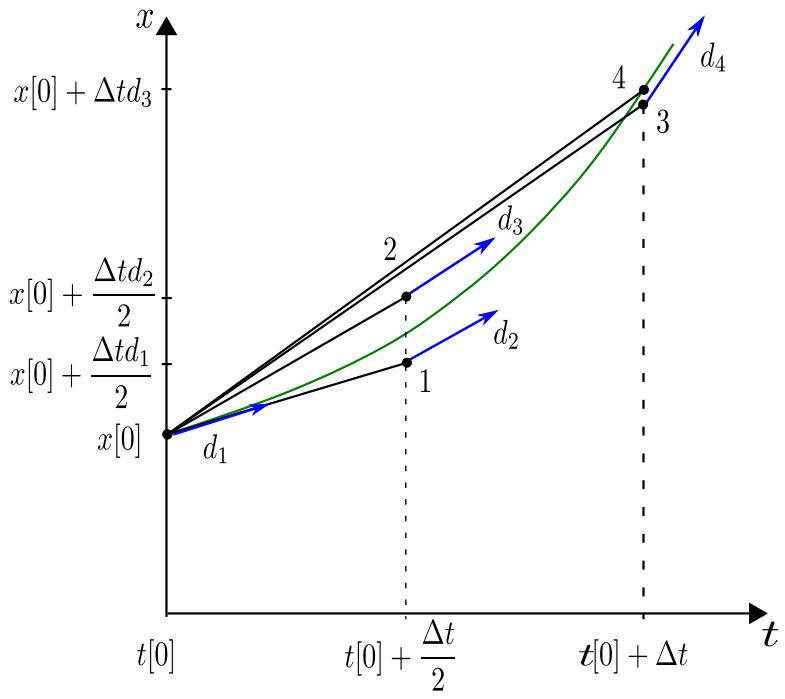
Esse novo valor da derivada é usado novamente para realizar outro deslocamento a partir do ponto inicial, avançando $\frac{\Delta t}{2}$ no sentido do tempo, ate um outro ponto 2, onde e calculado um terceiro valor da derivada:

$$d_3 = f\Big(x[0] + \frac{\Delta t d_2}{2}, t[0] + \frac{\Delta t}{2}\Big).$$

Seguindo o a direção da derivada d_3, realiza-se um terceiro deslocamento, a partir do ponto inicial, desta vez avançando uma distancia Δt no eixo do tempo, para chegar ate um ponto 3, onde se calcula um quarto valor da derivada:

$$d_4 = f\Big(x[0] + \Delta t d_3, t[0] + \Delta t\Big).$$

Figura A.1: Runge-Kutta de quarta ordem



Fonte: Elaborada pelo autor.

Pode mostrar-se que o valor da derivada que conduz a um erro mínimo é a combinação linear:

$$d = \frac{d_1 + 2(d_2 + d_3) + d_4}{6}.$$

Logo, Em cada ponto $(x[n], t[n])$, calcula-se o valor médio da derivada usando o mesmo processo, e com esse valor médio d, obtém-se o ponto seguinte na forma habitual:

$$x[n+1] = x[n] + \Delta t \frac{d_1 + 2(d_2 + d_3) + d_4}{6}.$$

A fim de tornar as equações mais simples para resolver computacionalmente, definimos $k_1 = \Delta t d_1$, $k_2 = \Delta t d_2$, $k_3 = \Delta t d_3$, $k_4 = \Delta t d_4$, então:

$$\begin{aligned}
k_1 &= \Delta t f(x[n], t[n]), \\
k_2 &= \Delta t f(x[n] + \tfrac{k_1}{2}, t[n] + \tfrac{\Delta t}{2}), \\
k_3 &= \Delta t f(x[n] + \tfrac{k_2}{2}, t[n] + \tfrac{\Delta t}{2}), \\
k_4 &= \Delta t f(x[n] + k_3, t[n] + \Delta t), \\
x[n+1] &= x[n] + \tfrac{k_1 + 2(k_2 + k_3) + k_4}{6}.
\end{aligned}$$

O método pode se estender a qualquer sistema do tipo 2.1. Bastando definir o método para cada variável. Sendo assim,

$$\begin{aligned}
\mathbf{k}_1 &= \Delta t \mathbf{f}(\mathbf{x}[n], t[n]), \\
\mathbf{k}_2 &= \Delta t \mathbf{f}(\mathbf{x}[n] + \tfrac{\mathbf{k}_1}{2}, t[n] + \tfrac{\Delta t}{2}), \\
\mathbf{k}_3 &= \Delta t \mathbf{f}(\mathbf{x}[n] + \tfrac{\mathbf{k}_2}{2}, t[n] + \tfrac{\Delta t}{2}), \\
\mathbf{k}_4 &= \Delta t \mathbf{f}(\mathbf{x}[n] + \mathbf{k}_3, t[n] + \Delta t), \\
\mathbf{x}[n+1] &= \mathbf{x}[n] + \tfrac{\mathbf{k}_1 + 2(\mathbf{k}_2 + \mathbf{k}_3) + \mathbf{k}_4}{6}.
\end{aligned}$$

A.2 Implementações em Python

```
# -*- coding: utf-8 -*-

#=======================================
#Bibliotecas usadas
#=======================================


import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import least_squares
from numpy.polynomial.polynomial import polyval3d
from numpy.random import normal
from scipy.interpolate import splrep,splder
from scipy.interpolate import splev


#=======================================
#Implemetação do sistema de Lorenz
#=======================================

sigma = 10.0
r = 28.0
b = 2.667

#Definimos o sitema de Lorenz
def f1(x1,x2,x3):
    x1dot = sigma*(x2-x1)
    return x1dot

def f2(x1,x2,x3):
    x2dot = r*x1-x2-x1*x3
    return x2dot

def f3(x1,x2,x3):
    x3dot = x1*x2-b*x3
    return x3dot
```

```
#=============================================
#Implemetação do Método de Runge-Kutta
#=============================================

dt = 0.01
t0 = 0.0
tf = 100.0

N = int(((tf)-t0)/dt) + 1
t = np.linspace(t0,tf,N)

N_trans = int(40.0/dt)+1
t_trans = np.linspace(t0,tf,N_trans)

x1_RK4_trans = np.zeros_like(t_trans)
x2_RK4_trans = np.zeros_like(t_trans)
x3_RK4_trans = np.zeros_like(t_trans)

##### condições iniciais #######
x1_RK4_trans[0]= 0.1
x2_RK4_trans[0]= 0.1
x3_RK4_trans[0]= 0.1

# roda uma vez, num tempo, para eliminar o transiente
for j in range(1,N_trans):
    k1 = dt*f1(x1_RK4_trans[j-1],x2_RK4_trans[j-1],
       x3_RK4_trans[j-1])
    l1 = dt*f2(x1_RK4_trans[j-1],x2_RK4_trans[j-1],
       x3_RK4_trans[j-1])
    m1 = dt*f3(x1_RK4_trans[j-1],x2_RK4_trans[j-1],
       x3_RK4_trans[j-1])

    k2 = dt*f1(x1_RK4_trans[j-1]+k1*0.5, x2_RK4_trans[j-1]+
       l1*0.5, x3_RK4_trans[j-1]+m1*0.5)
    l2 = dt*f2(x1_RK4_trans[j-1]+k1*0.5, x2_RK4_trans[j-1]+
       l1*0.5, x3_RK4_trans[j-1]+m1*0.5)
    m2 = dt*f3(x1_RK4_trans[j-1]+k1*0.5, x2_RK4_trans[j-1]+
       l1*0.5, x3_RK4_trans[j-1]+m1*0.5)
```

```
    k3 = dt*f1(x1_RK4_trans[j-1]+k2*0.5,x2_RK4_trans[j-1]+l2
        *0.5, x3_RK4_trans[j-1]+m2*0.5)
    l3 = dt*f2(x1_RK4_trans[j-1]+k2*0.5,x2_RK4_trans[j-1]+l2
        *0.5, x3_RK4_trans[j-1]+m2*0.5)
    m3 = dt*f3(x1_RK4_trans[j-1]+k2*0.5,x2_RK4_trans[j-1]+l2
        *0.5, x3_RK4_trans[j-1]+m2*0.5)

    k4 = dt*f1(x1_RK4_trans[j-1]+k3,x2_RK4_trans[j-1]+l3,
        x3_RK4_trans[j-1]+m3)
    l4 = dt*f2(x1_RK4_trans[j-1]+k3,x2_RK4_trans[j-1]+l3,
        x3_RK4_trans[j-1]+m3)
    m4 = dt*f3(x1_RK4_trans[j-1]+k3,x2_RK4_trans[j-1]+l3,
        x3_RK4_trans[j-1]+m3)

    x1_RK4_trans[j] = x1_RK4_trans[j-1] + (k1 + 2*k2 + 2*k3
        + k4)/6
    x2_RK4_trans[j] = x2_RK4_trans[j-1] + (l1 + 2*l2 + 2*l3
        + l4)/6
    x3_RK4_trans[j] = x3_RK4_trans[j-1] + (m1 + 2*m2 + 2*m3
        + m4)/6

# pega  último ponto como nova condição inicial
x1_0, x2_0, x3_0 = x1_RK4_trans[N_trans-1], x2_RK4_trans[
    N_trans-1], x3_RK4_trans[N_trans-1]



#%%

#=============================================================
# roda pela segunda vez e daí obtemos as séries temporais
#=============================================================
x1_RK4 = np.zeros_like(t)
x2_RK4 = np.zeros_like(t)
x3_RK4 = np.zeros_like(t)

x1_RK4[0], x2_RK4[0], x3_RK4[0] = x1_0, x2_0, x3_0
for j in range(1,N):
```

```
    k1 = dt*f1(x1_RK4[j-1],x2_RK4[j-1],x3_RK4[j-1])
    l1 = dt*f2(x1_RK4[j-1],x2_RK4[j-1],x3_RK4[j-1])
    m1 = dt*f3(x1_RK4[j-1],x2_RK4[j-1],x3_RK4[j-1])

    k2 = dt*f1(x1_RK4[j-1]+k1*0.5, x2_RK4[j-1]+l1*0.5,
       x3_RK4[j-1]+m1*0.5)
    l2 = dt*f2(x1_RK4[j-1]+k1*0.5, x2_RK4[j-1]+l1*0.5,
       x3_RK4[j-1]+m1*0.5)
    m2 = dt*f3(x1_RK4[j-1]+k1*0.5, x2_RK4[j-1]+l1*0.5,
       x3_RK4[j-1]+m1*0.5)

    k3 = dt*f1(x1_RK4[j-1]+k2*0.5,x2_RK4[j-1]+l2*0.5, x3_RK4
       [j-1]+m2*0.5)
    l3 = dt*f2(x1_RK4[j-1]+k2*0.5,x2_RK4[j-1]+l2*0.5, x3_RK4
       [j-1]+m2*0.5)
    m3 = dt*f3(x1_RK4[j-1]+k2*0.5,x2_RK4[j-1]+l2*0.5, x3_RK4
       [j-1]+m2*0.5)

    k4 = dt*f1(x1_RK4[j-1]+k3,x2_RK4[j-1]+l3, x3_RK4[j-1]+m3
       )
    l4 = dt*f2(x1_RK4[j-1]+k3,x2_RK4[j-1]+l3, x3_RK4[j-1]+m3
       )
    m4 = dt*f3(x1_RK4[j-1]+k3,x2_RK4[j-1]+l3, x3_RK4[j-1]+m3
       )

    x1_RK4[j] = x1_RK4[j-1] + (k1 + 2*k2 + 2*k3 + k4)/6
    x2_RK4[j] = x2_RK4[j-1] + (l1 + 2*l2 + 2*l3 + l4)/6
    x3_RK4[j] = x3_RK4[j-1] + (m1 + 2*m2 + 2*m3 + m4)/6

X = np.array([x1_RK4, x2_RK4, x3_RK4])
x1, x2, x3 = X[0], X[1], X[2]
x1_use, x2_use, x3_use = X[0], X[1], X[2]

#%%
#=======================================
#Implemetação das funções B-Spline
# s é o valor de suavização e
# k é o valor do grau da função
#=======================================
```

```
s=0

spl1 = splrep(t,x1,k=5,s=0)
P1 = splev(t,spl1)
dev1 = splder(spl1)
V1 = splev(t,dev1)

spl2 = splrep(t, x2,k=5,s=0)
P2 = splev(t, spl2)
dev2 = splder(spl2)
V2 = splev(t,dev2)

spl3 = splrep(t, x3,k=5,s=0)
P3 = splev(t, spl3)
dev3 = splder(spl3)
V3 = splev(t,dev3)

#%%

#=====================================
#Implemetação da função polinomial
# multidimensional de grau 2 para o uso do
# método de mínimos quadrados
#=====================================

def residue_3D(Coef,dados1,dados2,dados3,v):
    C3D = np.zeros((3,3,3))
    C3D[0,0,0] = Coef[0]
    C3D[1,0,0] = Coef[1]
    C3D[0,1,0] = Coef[2]
    C3D[0,0,1] = Coef[3]
    C3D[1,1,0] = Coef[4]
    C3D[1,0,1] = Coef[5]
    C3D[0,1,1] = Coef[6]
    C3D[2,0,0] = Coef[7]
    C3D[0,2,0] = Coef[8]
    C3D[0,0,2] = Coef[9]
    p3d_values =  polyval3d(dados1,dados2,dados3,C3D)
```

```
    return (p3d_values - v)


def convert(Coef):
    C3D = np.zeros((3,3,3))
    C3D[0,0,0] = Coef[0]
    C3D[1,0,0] = Coef[1]
    C3D[0,1,0] = Coef[2]
    C3D[0,0,1] = Coef[3]
    C3D[1,1,0] = Coef[4]
    C3D[1,0,1] = Coef[5]
    C3D[0,1,1] = Coef[6]
    C3D[2,0,0] = Coef[7]
    C3D[0,2,0] = Coef[8]
    C3D[0,0,2] = Coef[9]
    return C3D

#%%

#=====================================
#Implemetação do método de minimos
# quadrados, loss é o tipo de norma
#=====================================

C_v_ini = np.zeros(10)


F1_lq =  least_squares(residue_3D, C_v_ini, args=(P1,P2,P3,
   V1),loss='soft_l1')
F1_coef = F1_lq.x
#
F2_lq =  least_squares(residue_3D, C_v_ini, args=(P1,P2,P3,
   V2),loss='soft_l1')
F2_coef = F2_lq.x

F3_lq =  least_squares(residue_3D, C_v_ini, args=(P1,P2,P3,
   V3),loss='soft_l1')
F3_coef = F3_lq.x
```

```
F1_coeff = convert(F1_coef)
F2_coeff = convert(F2_coef)
F3_coeff = convert(F3_coef)

F1 = polyval3d(P1,P2,P3,F1_coeff)
F2 = polyval3d(P1,P2,P3,F2_coeff)
F3 = polyval3d(P1,P2,P3,F3_coeff)

#%%
#=========================================
#Implemetação da função recuperada
# C são os coeficientes que precisamos
#=========================================

def ff(C,dados1,dados2,dados3):
    result = polyval3d(dados1,dados2,dados3,C)
    return result

#%%
#=========================================
#Implemetação do ruído observacional
#=========================================

O = 0.0
D1 = 0.01*np.amax(np.fabs(x1))
D2 = 0.01*np.amax(np.fabs(x2))
D3 = 0.01*np.amax(np.fabs(x3))
r1 = normal(O,D1,N)
r2 = normal(O,D2,N)
r3 = normal(O,D3,N)

x1_r = x1 + r1
x2_r = x2 + r2
x3_r = x3 + r3

#%%
#=========================================
#Implemetação do método de coordenadas
```

```
#atrasadas
#=========================================


tau = 0.12
alpha = tau/dt
mu = 3
x = x1[:int(len(t)-(mu-1)*alpha)]
x_tau = x1[int(alpha):int(len(t)-alpha)]
x_2tau = x1[int(2*alpha):]
t_a = t[:int(len(t)-(mu-1)*alpha)]

#%%
#=========================================
#Implemetação da função polinomial
# multidimensional de grau 3 para o uso do
# método de mínimos quadrados
#=========================================



def residue_3D_gr3(Coef,dados1,dados2,dados3,v):
    C3D = np.zeros((4,4,4))
    C3D[0,0,0] = Coef[0]
    C3D[1,0,0] = Coef[1]
    C3D[0,1,0] = Coef[2]
    C3D[0,0,1] = Coef[3]
    C3D[1,1,0] = Coef[4]
    C3D[1,0,1] = Coef[5]
    C3D[0,1,1] = Coef[6]
    C3D[1,1,1] = Coef[7]
    C3D[2,0,0] = Coef[8]
    C3D[0,2,0] = Coef[9]
    C3D[0,0,2] = Coef[10]
    C3D[2,1,0] = Coef[11]
    C3D[2,0,1] = Coef[12]
    C3D[1,2,0] = Coef[13]
    C3D[1,0,2] = Coef[14]
    C3D[0,2,1] = Coef[15]
    C3D[0,1,2] = Coef[16]
    C3D[3,0,0] = Coef[17]
```

```
    C3D[0,3,0] = Coef[18]
    C3D[0,0,3] = Coef[19]
    p3d_values =  polyval3d(dados1,dados2,dados3,C3D)
    return (p3d_values - v)


def convert_gr3(Coef):
    C3D = np.zeros((4,4,4))
    C3D[0,0,0] = Coef[0]
    C3D[1,0,0] = Coef[1]
    C3D[0,1,0] = Coef[2]
    C3D[0,0,1] = Coef[3]
    C3D[1,1,0] = Coef[4]
    C3D[1,0,1] = Coef[5]
    C3D[0,1,1] = Coef[6]
    C3D[1,1,1] = Coef[7]
    C3D[2,0,0] = Coef[8]
    C3D[0,2,0] = Coef[9]
    C3D[0,0,2] = Coef[10]
    C3D[2,1,0] = Coef[11]
    C3D[2,0,1] = Coef[12]
    C3D[1,2,0] = Coef[13]
    C3D[1,0,2] = Coef[14]
    C3D[0,2,1] = Coef[15]
    C3D[0,1,2] = Coef[16]
    C3D[3,0,0] = Coef[17]
    C3D[0,3,0] = Coef[18]
    C3D[0,0,3] = Coef[19]
    return C3D
```