Controle Preditivo Não-Linear Baseado em Modelo Obtido via Aprendizado de Máquina ⋆

André C. Trevas * **Marcelo A. Santos** ** **Guilherme V. Raffo** ***

* *Engenharia de Controle e Automação - Universidade Federal de Minas Gerais (e-mail: andretrevas@gmail.com).*
** *Programa de Pós-Graduação em Engenharia Elétrica - Universidade Federal de Minas Gerais (e-mail: cidomg32@ufmg.br)*
*** *Departamento de Engenharia Eletrônica - Universidade Federal de Minas Gerais (e-mail: raffo@ufmg.br)*

Abstract: Model-based predictive control (MPC) is a control design methodology based on the existence of a prediction model to solve a constraint optimal control problem with a finite receding horizon. However, in the absence of mathematical models to describe the system's dynamical behaviour, it is necessary to use identification techniques to enable the design of predictive controllers. In this context, this work addresses the predictive control problem using an identification technique for nonlinear dynamical systems based on a non-parametric machine learning method known as Lazily Adaptive Constant Kinky Inference (LACKI). Therefore, with the prediction process being inferred from a data set describing the system input to output relations, a nonlinear predictive control strategy is presented and corroborated through a numerical example.

Resumo: Controle preditivo baseado em modelo (MPC) é uma metodologia de projeto de controle baseada na existência de um modelo de predição para a solução de um problema de controle ótimo com restrições em um horizonte finito e deslizante. No entanto, na indisponibilidade de modelos matemáticos que descrevam o comportamento dinâmico do sistema, faz-se necessário utilizar técnicas de identificação para viabilizar o projeto de controladores preditivos. Nesse contexto, esse trabalho aborda o problema de controle preditivo utilizando uma técnica de identificação de sistemas dinâmicos não-lineares baseada em métodos não-paramétricos de aprendizado de máquina, conhecida como *Lazily Adaptive Constant Kinky Inference* (LACKI). Portanto, valendo-se da predição inferida por meio de um conjunto de dados que descrevem relações de entrada e saída do sistema, uma estratégia de controle preditivo não-linear é apresentada e validada através de exemplo numérico.

Keywords: MPC; Machine Learning; Kinky Inference; Nonlinear Systems.
Palavras-chaves: MPC; Aprendizagem de Máquina; Kinky Inference; Sistemas Não-Lineares.

1. INTRODUÇÃO

Os sistemas presentes na natureza são, em sua maioria, não-lineares. Devido a isso, existem muitos trabalhos focados na identificação e controle desses sistemas. Uma forma de lidar com sistemas não-lineares é através de modelos linearizados em torno de pontos de operação, para os quais existem diversas técnicas de controle linear, por exemplo: alocação de pólos (Lv et al., 2010), *internal model control* (Datta, 2012), entre outras. No entanto, apesar de simplificar o modelo e permitir o uso de técnicas de controle lineares, o processo de linearização limita a faixa de operação na qual o sistema é bem representado. Nesse contexto, outra forma de abordar o problema é utilizando técnicas de controle não-linear como, por exemplo, controle preditivo não-linear baseado em modelo (Grüne and Pannek, 2017).

Existem vários benefícios na utilização de controladores preditivos baseado em modelo (*MPC - Model Predictive Control*) que fazem com que eles sejam amplamente utilizados. Pode-se destacar alguns aspectos, por exemplo, possibilidade de lidar com restrições impostas nas entradas, nos estados e nas saídas, sejam elas rígidas ou flexíveis; possibilidade de trabalhar com sistemas multivariáveis de forma simples e com otimização online. Esses aspectos são destacados por Wang (2009) para justificar a versatilidade e popularização desse tipo de controlador. Em Rossiter (2017), cita-se o benefício de utilizar MPC para sistemas multivariáveis complexos, resolvendo-os de uma forma simples. Em grande parte, as vantagens citadas para o MPC se baseiam na existência de um modelo para a predição dos estados futuros, na solução de um problema de controle ótimo restrito com horizonte finito e na utilização de uma estratégia de horizonte deslizante para realimentação das saídas medidas e posterior cálculo da ação de controle. No entanto, na indisponibilidade de modelos matemáticos que descrevam o comportamento

⋆ Esse trabalho foi apoiado pelos projetos INCT InSAC e Universal do CNPq sob as bolsas 465755/2014-3 e 426392/2016-7, e pelas agências CAPES e FAPEMIG.

de um determinado sistema dinâmico, faz-se necessário utilizar técnicas de identificação orientadas a dados para viabilizar o projeto de controladores preditivos. Por exemplo, técnicas de identificação baseadas em aprendizado de máquina (Calliess, 2014) e redes neurais (Nelles, 2013).

Os algoritmos da classe de aprendizado de máquina são uma parte do conceito genérico de inteligência artificial, o qual pode ser definido como o estudo e projeto de agentes inteligentes (Russell and Norvig, 2016). Essa classe de algoritmos é caracterizada por tomar decisões baseadas em dados obtidos previamente e utilizados no seu treinamento. Em Mitchell (1997), a seguinte definição para aprendizado de máquina foi feita: "Diz-se que um programa de computador aprende pela experiência E, com respeito a algum tipo de tarefa T e desempenho P, se seu desempenho P nas tarefas T, na forma medida por P, melhoram com a experiência E". Ou seja, há uma melhora no desempenho do algoritmo com a evolução dos dados.

Na literatura, pode-se encontrar diversos tipos de algoritmos de aprendizado de máquina segmentados pelo tipo de iteração, pela saída fornecida, pela forma de construção do algoritmo, entre outros (James et al., 2013). A principal segmentação é com relação a natureza do sinal disponível e existem três categorias principais, sendo elas: o aprendizado supervisionado, o aprendizado não-supervisionado e o aprendizado por reforço. O foco deste trabalho será na classe de algoritmos de aprendizado supervisionado, onde será fornecido um conjunto de dados de entradas e saídas. Objetivo do algoritmo é conseguir aprender como uma entrada se relaciona com uma saída para realizar inferências de saídas a partir de entradas não contidas no conjunto de treinamento (Caruana and Niculescu-Mizil, 2006). As principais utilizações desses algoritmos estão na área de computação (Richardson et al., 2006; Cui et al., 2006; Semenov et al., 2016). Por outro lado, na engenharia eles têm sido menos utilizados, uma vez que existem dificuldades para garantir de maneira formal a estabilidade dos algoritmos, ou seja, de garantir que o resultado estará sempre dentro de um limite pré-estabelecido.

Algumas das principais estratégias utilizadas na literatura são: aprendizado baseado em árvores de decisão (Dietterich, 2000), rede neural artificial (Anthony and Bartlett, 2009), aprendizado profundo (LeCun et al., 2015) e algoritmos genéticos (Thrift, 1991). Nesse trabalho, porém, será abordado uma estratégia conhecida como métodos baseados em *Nonlinear Set-Membership* (NSM) (Milanese and Novara, 2004; Raissi et al., 2004). Nessa abordagem, assume-se que podem ser definidos limites rígidos ou conjuntos limitados que contenham o conjunto de dados com os erros. Essa suposição leva a conjuntos de estados estimados onde é garantido que os estados reais estarão contidos neles (Scholte and Campbell, 2003). Logo, essa classe de métodos parte do pressuposto que existe um conjunto que contém os dados reais e as suas incertezas, sendo que é necessário saber apenas o limite máximo de erro contido nesse conjunto (Milanese and Vicino, 1991).

Alguns trabalhos na literatura utilizam NSM para aprender funções de classes definidas e as utilizarem para predição em estratégias MPC não-linear. Em Canale et al. (2014), utilizando os métodos propostos em Sukharev (1978) e Milanese and Novara (2004), consegue-se aprender funções Lipschitz contínuas, porém é necessário assumir que a constante de Lipschitz é conhecida. Em Limon et al. (2017), estima-se funções Hölder contínuas para predição baseado no método conhecido como *Lazily Adaptive Constant Kinky Inference* (LACKI) (Calliess et al., 2020). Nesse trabalho, a constante de Hölder é estimada a partir dos dados e, além disso, garante-se factibilidade recursiva e estabilidade para o controlador preditivo não-linear proposto a partir do conhecimento do pior caso para os erros de estimação.

Baseado nos resultados de Limon et al. (2017) e Calliess et al. (2020), nesse trabalho será utilizada a técnica não-paramétrica de aprendizado de máquina LACKI para a identificação de sistemas. O preditor apresentado será obtido a partir de um conjunto de dados coletados contendo as entradas e saídas do sistema de modo que não há necessidade de conhecer as equações dinâmicas que descrevem seu comportamento. Com essa técnica será possível inferir saídas para entradas contidas ou não no conjunto de dados utilizado para o treinamento. Posterior a isso, será possível projetar um controlador preditivo não-linear utilizando o preditor obtido com o LACKI. Por fim, para avaliação do algoritmo, se propõe um estudo de caso considerando o controle de um pêndulo invertido.

2. INFERÊNCIA COM LIMITES CONHECIDOS PARA PIOR CASO

Aplicando algoritmos de aprendizado de máquina é possível inferir/aprender o comportamento de sistemas dinâmicos a partir de um conjunto de dados de saídas e entradas, utilizando esse conhecimento para o projeto de controladores. Sabe-se que, dentre os objetivos de um sistema de controle, estão estabilidade e robustez em malha fechada. Desse modo, é necessário que a técnica de aprendizado de máquina utilizada forneça garantias de predição com os limites bem conhecidos. Nessa seção, apresenta-se o método *Kinky Inference* proposto em Calliess et al. (2020) com o objetivo de se obter um algoritmo capaz de inferir as saídas de um sistema dinâmico com o menor erro possível e com limites conhecidos de incerteza para o pior caso.

2.1 Kinky inference

O objetivo do método é aprender uma função $f(x)$, combinando o conjunto de dados fornecidos de entradas e saídas e utilizando o pressuposto da função ser Hölder contínua. Serão inferidas predições $\hat{f}_n(x)$ de $f(x)$ para dados de entrada não observados (não contidos) no conjunto fornecido, ou seja, será possível predizer a saída para entradas diferentes das fornecidas no conjunto de dados inicial.

Seja $(\mathcal{X}, \sigma_x), (\mathcal{Y}, \sigma_y)$ dois espaços métricos equipados, respectivamente, com pseudo-métricas σ_x e σ_y; e seja $I \subset \mathcal{X}$ um conjunto aberto. A função $f : \mathcal{X} \to \mathcal{Y}$ é chamada (L-p-) Hölder contínua em $I \subset \mathcal{X}$ se existir uma constante de Hölder $L \geq 0$ e um expoente de Hölder $p \geq 0$ tal que

$$\forall x,x' \in I : \sigma_y(f(x), f(x')) \leq L(\sigma_x(x, x'))^p. \qquad (1)$$

Ainda, dado $p \in (0, 1]$, para $0 \leq L_1 \leq L_2$, tem-se que $\mathcal{H}_x(L_1, p) \subseteq \mathcal{H}_x(L_2, p)$, sendo $\mathcal{H}_x(L,p)$ o conjunto de todas as funções que são Hölder contínuas com respeito ao espaço métrico $(\mathcal{X}, \sigma_x)$. Assim, o menor valor de $L^* \geq 0$, tal que a

função $f(x)$ é L*-p-Hölder, é chamado de melhor constante de Hölder para $f(x)$.

A função $f : \mathcal{X} \to \mathcal{Y}$ é a função que queremos inferir e os espaços $\mathcal{X}$ e $\mathcal{Y}$ representam, respectivamente, os espaços de entrada e de saída. O conjunto de dados utilizado nas predições será definido por $\mathcal{D}_n := \{(s_i, \tilde{f}_i) \mid i = 1, \ldots, N_n\}$ contendo N_n vetores $\tilde{f}_i \in \mathcal{Y}$ com amostras da função f para $s_i \in \mathcal{X}$ entradas. Os vetores $\tilde{f}_i$ podem ter erro de observação, ou ruído, dentro do intervalo limitado por e_j, onde e_j denota a confiança sobre o erro real da observação. Ou seja, sabe-se que $f(s_i) \in O_i := [\underline{f_1}(s_i), \overline{f_1}(s_i)] \times \cdots \times [\underline{f_d}(s_i), \overline{f_d}(s_i)]$, onde $\underline{f_j}(s_i) := \tilde{f}_{i,j} - e_j(s_i)$, $\overline{f_j}(s_i) := \tilde{f}_{i,j} + e_j(s_i)$, sendo $\tilde{f}_{i,j}$ a j-ésima componente do vetor $\tilde{f}_i$.

O método proposto para a identificação de sistemas é considerado um método da classe de aprendizado de máquina uma vez que ele melhora o seu desempenho com o aumento da quantidade de dados, ou seja, o modelo inferido a partir do conjunto de dados $\mathcal{D}_{n+1}$, com $\mathcal{D}_n \subset \mathcal{D}_{n+1}$, é mais próximo do modelo real que o modelo inferido a partir do conjunto de dados $\mathcal{D}_n$. Ainda, o aumento do conjunto de dados não aumenta a incerteza de predição.

Seja $\mathbb{R}_\infty := \mathbb{R} \cup \{-\infty, \infty\}$ e $\mathcal{X}$ espaços equipados com pseudo-métricas σ_x. Ainda, seja $\underline{B}, \overline{B} : \mathcal{X} \to \mathcal{Y} \subseteq \mathbb{R}_\infty^m$ limitadores inferior e superior, com $\underline{B}(x) \leq \overline{B}(x)\ \forall x \in \mathcal{X}$. Assim, dado um conjunto de dados $\mathcal{D}_n$, a função de predição $\hat{f}_n : \mathcal{X} \to \mathcal{Y}$ e a função de incerteza $\hat{\sigma}_n : \mathcal{X} \to \mathbb{R}_{\geq 0}^m$ capazes de inferir valores da função alvo $f(x)$ são dadas por

$$\begin{aligned}\hat{f}_{n,j}(x) :=& w_1 \min\{(\overline{B}_j(x), \mu_{n,j}(x; L(n))\} + \\ & w_2 \max\{(\underline{B}_j(x), \iota_{n,j}(x; L(n))\},\end{aligned} \tag{2}$$

$$\begin{aligned}\hat{\sigma}_{n,j}(x) :=& w_1 \min\{(\overline{B}_j(x), \mu_{n,j}(x; L(n))\} - \\ & w_2 \max\{(\underline{B}_j(x), \iota_{n,j}(x; L(n))\},\end{aligned} \tag{3}$$

onde j representa a j-ésima componente do vetor. Além disso, $\mu_{n,j}(x; L(n)) : \mathcal{X} \to \mathbb{R}^m$ e $\iota_{n,j}(x; L(n)) : \mathcal{X} \to \mathbb{R}^m$ são, respectivamente, funções *ceiling* e *floor* e w_1 e w_2 são pesos para a predição.

A predição obtida pelas equações (2) e (3) serão valores entre as funções *ceiling* e *floor* de acordo com a escolha de w_1 e w_2. Os limites superior, $\overline{B}$, e inferior, $\underline{B}$, podem ser escolhidos baseados nos conhecimentos a priori do sistema ou podem ser definidos como $-\infty$ e $+\infty$ caso não haja restrições.

As funções *ceiling* e *floor* são definidas como

$$\mu_{n,j}(x; L(n)) := \min_{i=1,\ldots,N_n} \tilde{f}_{i,j} + L_j(n)\sigma_x^p(x, s_i) + e_j(x), \tag{4}$$

$$\iota_{n,j}(x; L(n)) := \max_{i=1,\ldots,N_n} \tilde{f}_{i,j} - L_j(n)\sigma_x^p(x, s_i) - e_j(x), \tag{5}$$

onde $L(n)$ é uma constante de Hölder, $\sigma_x^p(x, s_i)$ é uma pseudo-métrica que pode ser escolhida a partir do conhecimento do sistema ou de maneira *ad hoc*, *e.g.*, norma 1, norma 2, norma infinita, entre outras; e o erro $e_j(x)$ descreve a precisão dos dados que foram fornecidos para a estimação do modelo. Ademais, os valores de L e p determinam o formato das funções *ceiling* e *floor* e devem ser escolhidos pelo projetista baseado em conhecimentos prévios da função. Os valores dessas variáveis podem mudar de acordo com cada componente da função de predição, porém, elas podem ser mantidas com valores constantes visando simplificar a implementação, ou seja, $L(n) = L$, $e_j(x) = e$, $\overline{B}_j(x) = \overline{B}$ e $\underline{B}_j(x) = \underline{B}$.

A função $\hat{f}_n(x)$ é o preditor que será utilizado para predizer saídas a partir de entradas que podem ou não estar no conjunto de dados utilizado, enquanto a função $\hat{\sigma}(x)$ é utilizada para quantificar a incerteza do preditor $\hat{f}_n(x)$. Em outras palavras, deseja-se fornecer uma função de predição para a incerteza $\hat{\sigma}(x) : \mathcal{X} \to \mathcal{Y}$ tal que acredita-se que para qualquer dado de entrada $x \in \mathcal{X}$, o valor da função real $f(x)$ está contido no hiper-retângulo

$$\hat{H}_n(x) := \{y \in \mathcal{Y} | \forall_j \in \{1, \ldots, \dim(\mathcal{Y})\} : y_j \in \hat{H}_{n,j}(x)\}, \tag{6}$$

onde

$$\hat{H}_{n,j}(x) := [\hat{f}_{n,j}(x) - \hat{\sigma}_{n,j}(x), \hat{f}_{n,j}(x) + \hat{\sigma}_{n,j}(x)], \tag{7}$$

refere-se ao j-ésimo intervalo predito.

2.2 Lazily adaptive constant kinky inference

Têm-se assumido até o momento que as constantes de Hölder são conhecidas. Todavia, em muitos casos se tem pouco ou nenhum conhecimento sobre essas constantes. Devido a isso, em Calliess et al. (2020), o método *Lazily Adaptive Constant* (LAC) é proposto para encontrar o valor de L baseado no conjunto de dados $\mathcal{D}_n$. Além disso, o LAC pode ser utilizado para melhorar a estimativa de L de acordo com a disponibilidade dos dados. A escolha correta da constante de Hölder é importante uma vez que se for tomado um valor conservativo (valor alto), o preditor pode ter predições abruptas, levando a um baixo desempenho. Por outro lado, sub-estimar L pode levar o preditor a ser incapaz de inferir de maneira correta a real forma da função $f(x)$.

Assumindo o conjunto de dados $\mathcal{D}_n \subset \mathcal{D}_{n+1}$, a constante de Hölder pode ser atualizada utilizando o LAC por meio da equação

$$L(n+1) := \max\left\{L(n), \max_{i=1,\ldots,N_n} \frac{\sigma_y(\tilde{f}_{N+1}, \tilde{f}_i) - 2\overline{e}}{\sigma_x^p(s_{N+1}, s_i)}\right\}. \tag{8}$$

Todavia, não havendo atualização do conjunto de dados em tempo de execução, pode-se utilizar

$$L(n) := \max_{i,j=1,\ldots,N_n, \sigma_x(s_i,s_j)>0} \frac{\sigma_y(\tilde{f}_i, \tilde{f}_j) - 2\overline{e}}{\sigma_x^p(s_i, s_j)}, \tag{9}$$

sendo $\sigma_y(.,.)$ e $\sigma_x(.,.)$ pseudo-métricas quaisquer e $\overline{e} := \sup|e(x)|$ o erro máximo contido no conjunto de dados.

2.3 Escolha da constante p

A definição do valor da constante p fica a cargo do projetista, baseado na experiência e em critérios analíticos. Esse valor é utilizado na obtenção das funções *ceiling* e *floor*, e, em conjunto com a constante L, define o formato dessas funções. Desse modo, a escolha incorreta de p pode piorar o desempenho do preditor.

Dado que o valor de p deve estar entre 0 e 1, pode-se avaliar valores em simulação por meio do erro médio quadrático (MSE - *Mean Squared Error*) para se obter o valor de p que minimiza essa métrica. A escolha pode ser feita tomando valores de p igualmente espaçados dentro do intervalo (0,1]. Porém, vale ressaltar que essa abordagem é possível apenas

quando se tem algum conhecimento do modelo, mesmo que simplificado, o que possibilita comparar a função predita com a função original. Não havendo nenhum conhecimento sobre o modelo, p ainda pode ser escolhido de maneira arbitrária desde que $p \in (0, 1]$.

2.4 Construção do conjunto de dados $\mathcal{D}_n$

Para que haja uma boa representação do modelo real é necessário obter saídas para diferentes entradas com várias condições iniciais, ou seja, para a condição inicial c_1 deve-se obter saídas $\tilde{f}_1, \ldots, \tilde{f}_n$ para as entradas $s_1, \ldots, s_n$, onde as entradas e saídas podem ser de qualquer dimensão e o valor n é definido pelo tamanho do conjunto de entradas. As saídas devem ser obtidas após um determinado tempo t da aplicação da entrada, ou seja, será armazenado apenas os valores das saídas após t segundos e não a evolução do sistema desde a aplicação da entrada até t segundos depois. O processo é repetido para as condições iniciais $c_1, \ldots, c_m$, onde m deve ser escolhido para que haja uma boa quantidade de condições iniciais que representem o sistema. Por exemplo, para um pêndulo invertido, o ideal é que a condição inicial varie entre $[-2\pi, 2\pi]$, cobrindo todo o espaço de trabalho do sistema. Portanto, o conjunto $\mathcal{D}_n$ é formado por m subconjuntos, onde cada subconjunto tem n pares de entradas e saídas. O aumento de m e n tem efeito sobre o tempo de processamento do algoritmo e sobre o MSE da estimação. Ainda, quanto mais dados, melhor é o desempenho do preditor, porém isso aumenta o tempo de execução do algoritmo.

Com essa formulação, os dados obtidos representam bem o sistema, porém tem-se m subconjuntos. De modo que, executar o algoritmo para todos eles fica inviável devido ao alto tempo de execução. Dessa forma, faz-se necessário escolher um dos subconjuntos para cada entrada que se deseja predizer a saída e deve-se utilizar o conjunto de dados que possui valor de condição inicial mais próximo da entrada fornecida. Ou seja, para cada entrada será utilizado apenas um subconjunto de $\mathcal{D}_n$ com n dados de entradas e saídas.

3. CONTROLE PREDITIVO NÃO-LINEAR

Controladores preditivos são ditos não-lineares quando seu problema de controle ótimo associado é um problema de programação não-linear, com exceção daqueles que resultam em um problema de otimização quadrático puro. Comumente os controladores preditivos se enquadram nessa categoria por utilizarem modelos não-lineares para realizar predições. A utilização de modelos não-lineares trazem a vantagem de representar com maior precisão o sistema dinâmico, todavia tem a desvantagem de aumentar o custo computacional necessário para a resolução do problema de otimização.

O conceito de horizonte deslizante é fundamental e consiste na ideia de que os estados serão preditos em N_p instantes futuros dado os estados atuais do sistema. Tais predições serão utilizadas para obter N_c ações de controle que levam o sistema de maneira ótima para a referência desejada dentro do horizonte considerado (Camacho and Bordons, 2007). Na estratégia de horizonte deslizante, o primeiro valor da sequência ótima de controle obtida é aplicado ao sistema e então a janela de predição é deslocada em uma amostra a frente e o problema de controle ótimo é resolvido novamente considerando como condição inicial as medições atuais. Na literatura, as variáveis N_p e N_c são chamadas de horizonte de predição e horizonte de controle.

3.1 MPC não-linear com modelo físico

Considere um sistema não-linear em tempo discreto com dimensão finita da forma

$$x_{k+1} = f(x_k, u_k), \tag{10}$$

onde $x \in \mathbb{R}^m$ representa o vetor de estados e $u \in U \subset \mathbb{R}^n$ representa o vetor de entradas, com U sendo um conjunto convexo admissível de entradas de controle.

Para o caso de controle regulatório, pode-se considerar um problema de controle ótimo que penaliza os erros entre os estados, x, e os estados desejados, x^{ref} e o sinal de controle, u. Desse modo, considerando as restrições do sistema, o seguinte problema de otimização pode ser proposto

$$\begin{aligned} \min_{\boldsymbol{u},\boldsymbol{x}} \quad & \sum_{i=0}^{N_p} ||x_{k+i} - x_{k+i}^{ref}||_Q^2 + \sum_{j=0}^{N_c-1} ||u_{k+j}||_R^2 \\ \text{sujeito a} \quad & x_{k+i+1} = f(x_{k+i}, u_{k+i}), \forall i = 0, \ldots, N_p - 1, \\ & u_{k+i} \in U, \forall i = 0, \ldots, N_c - 1, \end{aligned} \tag{11}$$

sendo Q e R, respectivamente, matrizes de ponderação do erro dos estados e dos sinais de controle. Ainda, $\boldsymbol{u} = \{u_k, \ldots, u_{k+N_c-1}\}$ e $\boldsymbol{x} = \{x_k, \ldots, x_{k+N_p}\}$ denotam, respectivamente, as sequências ótimas de controle e de estados preditos.

3.2 MPC não-linear via aprendizado de máquina

O problema apresentado na subseção anterior pode ser adaptado para utilizar o preditor LACKI proposto no Seção 2 para fazer as predições ao longo do horizonte considerado. Desse modo, o controlador preditivo passa a não depender de um modelo para o sistema, pois requer apenas uma base de dados representativa capaz de relacionar as entradas e saídas do sistema. Portanto, será necessário ajustar as equações (10) e (11) para utilizar o preditor LACKI ao invés do modelo, levando em conta que as saídas inferidas mapeiam uma dada entrada em um vetor de estados. Com isso, a nova equação para predição é dada por

$$\hat{x}_{k+1} = \hat{f}_n(u_k), \tag{12}$$

onde o termo $\hat{x}_{k+1}$ indica o próximo estado predito baseado no preditor $\hat{f}_n$, dado o sinal de controle u_k. Assim, o problema de otimização pode ser reescrito como

$$\begin{aligned} \min_{\boldsymbol{u},\hat{\boldsymbol{x}}} \quad & \sum_{i=0}^{N_p} ||\hat{x}_{k+i} - x_{k+i}^{ref}||_Q^2 + \sum_{j=0}^{N_c-1} ||u_{k+j}||_R^2 \\ \text{sujeito a} \quad & \hat{x}_{k+i+1} = \hat{f}_n(u_{k+i}), \forall i = 0, \ldots, N_p - 1, \\ & u_{k+i} \in U, \forall i = 0, \ldots, N_c - 1. \end{aligned} \tag{13}$$

3.3 Problema de otimização

Os problemas de otimização (11) e (13) podem ser tratados com a predição feita de forma explícita, na forma de restrição de igualdade, ou de forma implícita, incluindo o modelo de predição diretamente na função objetivo.

Uma vantagem de se trabalhar formulando o problema na forma implícita é a possível simplificação de não utilizar restrições não-lineares que podem ser não-convexas e, além disso, pode-se evitar os problemas de factibilidade. Todavia, a abordagem implícita traz a desvantagem de aumentar os problemas com mínimos locais ao incluir não-linearidades adicionais à função objetivo. Além disso, vale ressaltar que espera-se que o problema de otimização para o MPC não-linear via aprendizado de máquina seja mais difícil de ser resolvido por possuir não-linearidades duras na formulação do preditor, como funções de máximo e de mínimo.

4. EXEMPLO NUMÉRICO - PÊNDULO INVERTIDO

Para avaliar os algoritmos propostos, será apresentado um estudo de caso considerando um sistema pêndulo invertido. Para tanto, inicialmente será avaliado o desempenho do preditor utilizando dados de entrada e saída para o sistema em malha fechada com um controlador regulador linear quadrático (LQR). Posteriormente, o preditor obtido será utilizado para implementação do MPC não-linear via aprendizado de máquina e os resultados serão comparados com um MPC não-linear baseado no modelo físico.

4.1 Sistema pêndulo invertido

O sistema pêndulo invertido, ilustrado na Figura 1, possui como estados a posição, θ, e a velocidade angular, $\dot{\theta}$, sendo atuado em torque, u. A dinâmica do sistema pode ser representada através da equação diferencial de segunda ordem

$$l^2 M\ddot{\theta} = Mglsin(\theta) - \beta\dot{\theta} + u, \tag{14}$$

onde $g = 9{,}81\ m/s^2$ é a aceleração da gravidade, $l = 1\ m$ é o comprimento da haste, $M = 1\ kg$ é a massa do pêndulo e $\beta = 0{,}5\ N.m.s$ é o atrito viscoso na junta, proporcional à velocidade angular. Ademais, assume-se o torque aplicado ao sistema dentro do intervalo $u \in [-10, 10]\ N.m$.

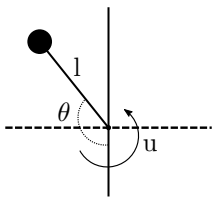

Figura 1. Sistema pêndulo invertido.

4.2 Predição via aprendizado de máquina

Sabe-se que o sistema (14) é instável em malha aberta e, portanto, o preditor inferido pelo LACKI deve ser avaliado em malha fechada. Para tanto, implementou-se um controlador não-linear através da técnica de controle com linearização por realimentação de saída (*feedback linearization*), sendo o sistema linear resultante controlado por meio de um controlador LQR.

Definindo uma variável de controle artificial v, pode-se escrever a lei de controle não-linear como

$$u = l^2 M\left(\frac{-g}{l}sin\theta + v\right), \tag{15}$$

de modo que o modelo resultante, aplicando (15) em (14), é dado por

$$\ddot{\theta} = \frac{-\beta}{l^2 M}\dot{\theta} + v. \tag{16}$$

Por fim, representando (16) no espaço de estados, obtêm-se

$$\begin{bmatrix}\dot{\theta}\\ \ddot{\theta}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & \frac{-\beta}{l^2 M}\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix} v. \tag{17}$$

Utilizando a função do Matlab® *dlqr* para obter a matriz de ganho para a versão discreta de (17) obtida pela aproximação de Euler, com parâmetros $Q = \text{diag}([20\ 1]), R = 0{,}01$ e $t_s = 0{,}05$, onde Q e R são pesos para a matriz de ganho, diag([·]) é uma matriz diagonal com valores dados pelo vetor [·] e t_s é o tempo de amostragem para a discretização, tem-se

$$v = [12{,}2638\ \ 5{,}2034]\,(\theta^{ref} - \theta). \tag{18}$$

Com o sistema em malha fechada, utilizou-se para teste uma referência em degrau de amplitude π aplicada no instante t_a:

$$x(k) = \begin{cases} 0; \text{ se } t < t_a, \\ \pi; \text{ se } t \geq t_a; \end{cases}$$
$$\dot{x}(k) = 0.$$

O conjunto de dados $\mathcal{D}_n$ é composto dos dados de entrada, torque, e saídas, posição e velocidade angular. Para a coleta de dados, o torque foi amostrado com incrementos de $0{,}2\ N.m$ dentro do intervalo $[-10, 10]\ N.m$. Para as condições iniciais, optou-se por uma variação de $[-2\pi, 2\pi]$, espaçados de $\pi/80\ rad$, para a posição e a velocidade fixa em 0. O algoritmo foi implementado com parâmetros $\underline{B} = -3\pi, \overline{B} = 3\pi, e = [0{,}2\ 6{,}5]^T, w_1 = 0{,}5, w_2 = 0{,}5, p = 1, \overline{e} = 0{,}1$ e L variando de acordo com o conjunto de dados selecionado. Ainda, considerou-se a pseudo-métrica $\sigma_x(x, x') = ||x - x'||_\infty$. O valor de L variou de 0,04 até 3,89, dependendo do conjunto utilizado. No total foram gerados 321 conjuntos de dados, sendo que cada um possui 101 amostras de entrada e saídas. Com o conjunto de dados obtido, utilizou-se o preditor para inferir as saídas do sistema dinâmico dada uma entrada qualquer.

Na Figura 2 é aplicado um degrau de referência de amplitude $\pi\ rad$ para que o sistema seja controlado com a haste

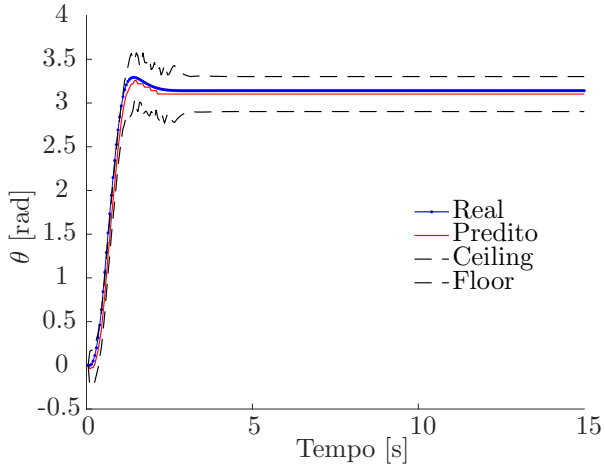

Figura 2. Predição da posição para um degrau de referência de amplitude $\pi\ rad$.

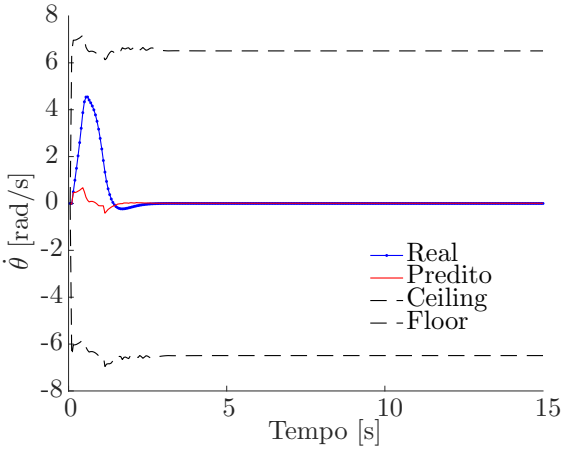

Figura 3. Predição da velocidade para um degrau de referência de amplitude π *rad*.

para cima. Pode-se verificar que a posição simulada e a predita possuem valores próximos. Na Figura 3, por outro lado, os sinais de velocidade apresentam uma diferença no início da simulação, mas depois as funções se aproximam e a estimação é satisfatória.

Com o objetivo de se avaliar o efeito da quantidade de dados no MSE e no tempo de execução do algoritmo, efetuou-se testes para avaliar o desempenho do algoritmo com o aumento do conjunto de dados $\mathcal{D}_n$. Para o primeiro teste, o valor do torque varia constantemente entre $[-10, 10]$ $N.m$, enquanto, para o segundo teste, o valor da posição inicial varia entre $[-2\pi, 2\pi]$ *rad*. Em ambos os testes o valor da velocidade é fixo em 0 rad/s.

A variação da quantidade de dados de torque afeta o tempo de execução do algoritmo, visto que cada subconjunto de dados passa a ter mais dados. Os testes foram realizados para variação na posição inicial fixa em $\pi/8$ *rad*, 33 subconjuntos, e diferentes espaçamentos no torque, com variações de 1, 0,5 e 0,1, ou seja, 21, 41 e 201 amostras por subconjunto, respectivamente. Os tempos de execução e o MSE são exibidos na Tabela 1. O aumento do tamanho de cada subconjunto faz com que a constante de Hölder seja melhor estimada levando a uma redução do MSE. Como todos os testes já possuem uma constante L bem estimada, a redução no MSE é pequena. Por outro lado, percebe-se que o aumento do tempo é não-linear em relação a quantidade de amostras e isso é explicado pelo custo da função de estimação da constante L, o qual é $O(n^2)$. Logo, o ideal é escolher a menor quantidade de amostras de torque que forneçam uma boa estimação da constante L.

Variação	Tempo máximo	Tempo médio	MSE $(\theta; \dot{\theta})$
1 N	82 ms	2,3 ms	(0,1686; 0,3839)
0,5 N	49,3 ms	6,9 ms	(0,1681; 0,3883)
0,1 N	262,2 ms	123,5 ms	(0,1677; 0,3929)

Tabela 1. Influência do tamanho dos conjuntos no tempo de execução do algoritmo.

A variação da quantidade de dados da posição inicial, por outro lado, afeta o MSE, visto que existem mais subconjuntos de dados e a variação das entradas dentro de cada subconjunto é menor. Os testes foram realizados para variação no torque fixa em 0,5 $N.m$, 41 amostras, e diferentes espaçamentos na posição inicial, com variações de $\pi/4$ *rad*, $\pi/8$ *rad* e $\pi/80$ *rad*, ou seja, 17, 33 e 321 subconjuntos, respectivamente. Os tempos de execução e o MSE são exibidos na Tabela 2. O tempo de execução praticamente não apresenta alteração, visto que as funções *ceiling* e *floor* possuem custo $O(n)$ e a seleção do conjunto a ser utilizado também possui o mesmo custo, ou seja, a tendência é que não haja variações significativas com o aumento da quantidade de subconjuntos. Por outro lado, o MSE é muito afetado por esse aumento, uma vez que quanto maior a quantidade de conjuntos mais escalonada está a condição inicial, ou seja, o erro entre a entrada e a posição inicial do subconjunto escolhido será minimizado. Portanto, o ideal é ter o máximo de condições iniciais possíveis tal que o tempo total não seja um problema.

Variação	Tempo máximo	Tempo médio	MSE $(\theta; \dot{\theta})$
$\pi/4$ *rad*	30,9 ms	7,7 ms	(0,6492; 0,4860)
$\pi/8$ *rad*	11,6 ms	6,1 ms	(0,1681; 0,3883)
$\pi/80$ *rad*	14,6 ms	6,5 ms	(0,0031; 0,3461)

Tabela 2. Influência da quantidade de conjuntos no MSE.

4.3 MPC não-linear via aprendizado de máquina

O controle do pêndulo invertido será feito utilizando as técnicas da Seção 3. Os parâmetros utilizados para o preditor LACKI e para o modelo são os mesmos da subseção anterior. Os ganhos das matrizes Q e R foram definidos como diag([1 1]) e 0,01, respectivamente, e o período de amostragem foi definido como 0,1 segundo. A restrição no sinal de controle foi considerada para que ele esteja no intervalo $[-10, 10]$ $N \cdot m$, a posição inicial considerada é $x_0 = [0{,}1\ 0]$ e $N_p = 4$ e $N_c = 2$ são os horizontes de predição e controle considerados. Para fins de comparação, um controlador MPC não-linear baseado no modelo do sistema representado na equação (14) é obtido utilizando os mesmos parâmetros de ajuste. Para resolução do problema de otimização formulado na forma implícita, considerou-se a heurística patternsearch do Matlab®.

Para avaliar o sistema de controle, utilizou-se uma referência em degrau com amplitude π aplicado no instante de tempo t_a:

$$x(t) = \begin{cases} 0; & \text{se } t < t_a, \\ \pi; & \text{se } t \geq t_a; \end{cases}$$
$$\dot{x}(t) = 0.$$

Nas Figuras 4 e 5 são apresentados os resultados para o sistema pêndulo invertido utilizando o MPC não-linear via modelo, NMPC, e o MPC não-linear via aprendizado de máquina, l-NMPC. Verifica-se na Figura 4 que os controladores efetuam a estabilização do pêndulo na posição π *rad*. O NMPC não possui sobressinal, sendo o sistema estabilizado em π *rad*, enquanto o l-NMPC apresenta oscilações significativas, mas também consegue estabilizar o sistema. Ademais, o preditor para esse sistema apresenta erros na estimação da velocidade, o que prejudica o desempenho do controlador. Os sinais de controle apresentados na Figura 6 demonstram o motivo da oscilação do sistema controlado pelo l-NMPC, uma vez que ele apresenta grandes variações em um curto espaço de tempo. O controlador NMPC, por outro lado, fornece ao sistema um sinal de controle mais suave.

Na Figura 7, os parâmetros do controlador l-NMPC foram alterados, atribuindo um peso maior para o erro na velocidade com o intuito de melhorar a sua regulação, dado que ela é mal estimada pelo preditor. A nova sintonia, dada por $Q = \text{diag}([1\ 10])$, melhora o desempenho do algoritmo, fazendo com que a oscilação diminua e o problema de regulação de ambas as variáveis seja melhorado, visto que essa nova configuração pondera mais o erro na velocidade.

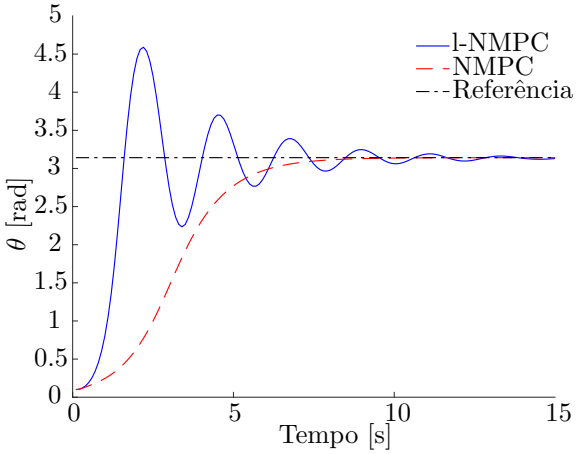

Figura 4. Evolução temporal da posição para uma referência em degrau de amplitude π.

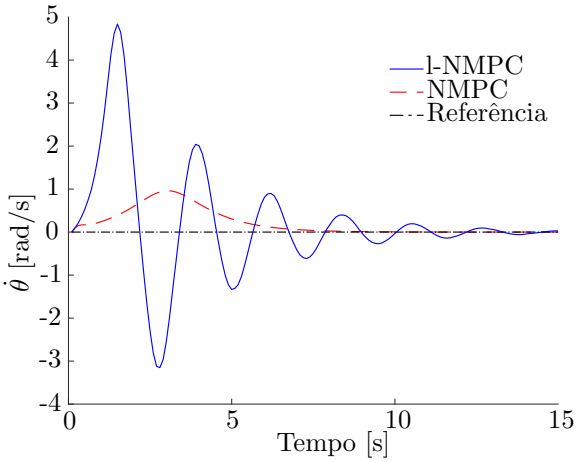

Figura 5. Evolução temporal da velocidade para uma referência em degrau de amplitude π.

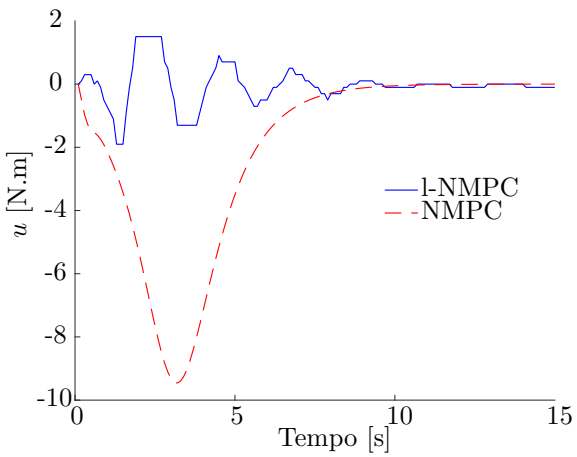

Figura 6. Sinal de controle para ambos os controladores.

A Figura 8 evidencia a melhora no controle da velocidade ao se ponderá-la melhor, refletindo no controle da posição. Para esta sintonia, a Figura 9 indica que o controlador ainda é agressivo, mas ele consegue ponderar melhor os erros e estabiliza mais rápido, sem tanta oscilação no sinal de controle.

Os resultados apresentados na Tabela 3 ressaltam a dificuldade do l-NMPC de regular a velocidade, uma vez que a estimação feita pelo preditor é ruim para essa variável. A

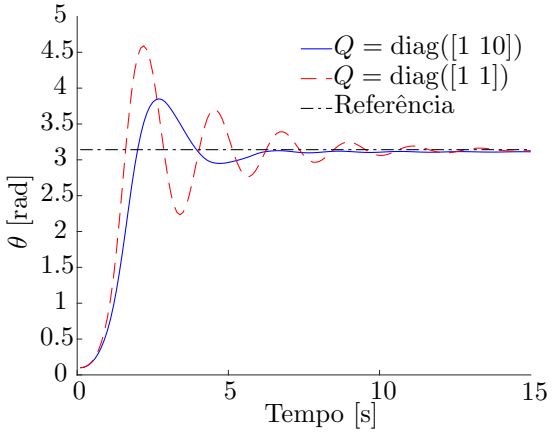

Figura 7. Evolução temporal da posição ponderando mais o erro na velocidade.

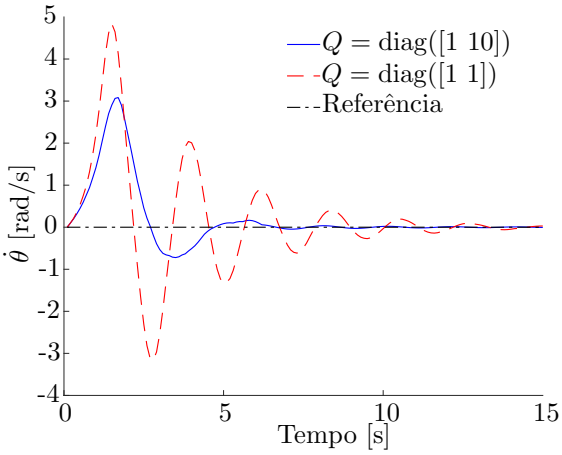

Figura 8. Evolução temporal da velocidade ponderando mais o erro na velocidade.

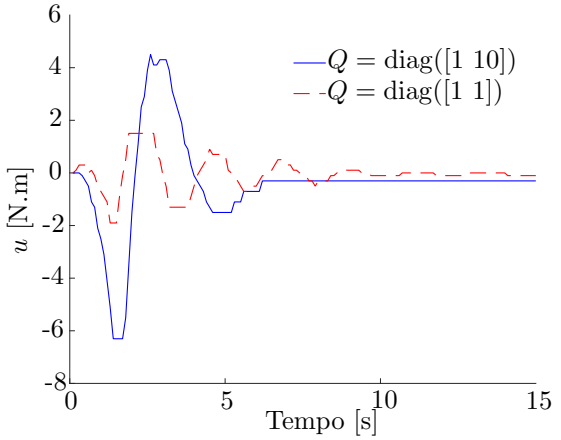

Figura 9. Sinal de controle para a sintonia com ponderação na velocidade.

segunda sintonia realizada reduz o MSE para a velocidade, mas ainda continua com um valor alto quando comparado ao NMPC, evidenciando a melhora proporcionada pelo novo controlador mas ressaltando a dificuldade em manter baixas acelerações angulares. Por outro lado, a posição alcança a referência, apesar das oscilações devido ao comportamento agressivo do controlador.

Controlador	MSE θ	MSE $\dot{\theta}$	IADU
NMPC	2,2499	0,1885	188,6700
l-NMPC	1,1274	2,5079	157,5079
l-NMPC melhorado	1,0924	0,8602	247,1483

Tabela 3. Métricas utilizadas para mensurar o desempenho dos controladores.

Por fim, cabe ressaltar que as estratégias de controle preditivo apresentadas para o caso de regulação podem ter erro em regime permanente devido às diversas fontes de incertezas e falta de ação integral. Esse problema poderia ser contornado, por exemplo, estendendo os controladores apresentados de acordo com a metodologia de *off-set free* (Morari and Maeder, 2012).

5. CONCLUSÃO

Neste trabalho foi apresentada uma estratégia de predição com regra de inferência baseada em algoritmo de aprendizado de máquina, bem como uma estratégia de controle preditivo não-linear que utiliza o método de aprendizado para realizar predições do comportamento do sistema. Utilizando o algoritmo LACKI foi possível inferir saídas do sistema garantindo que os seus valores se mantivessem dentro de um limite de incerteza conhecido. A partir do método de inferência, foi possível obter um controlador preditivo não-linear com o processo de predição baseado em um conjunto de dados representativo do sistema e não em um modelo fenomenológico. Para corroborar os algoritmos de inferência e controle, resultados numéricos foram apresentados considerando um sistema pêndulo invertido como estudo de caso. Em trabalhos futuros, espera-se desenvolver métodos para obter e aplicar os algoritmos em sistemas de alta ordem, e.g., veículos aéreos não-tripulados. Além disso, será estudado métodos para se resolver o problema de otimização de maneira eficiente.

REFERÊNCIAS

Anthony, M. and Bartlett, P.L. (2009). Neural network learning: Theoretical foundations.

Calliess, J.P. (2014). *Conservative decision-making and inference in uncertain dynamical systems*. Ph.D. thesis, University of Oxford.

Calliess, J.P., Roberts, S.J., Rasmussen, C.E., and Maciejowski, J. (2020). Lazily adapted constant kinky inference for nonparametric regression and model-reference adaptive control. *Automatica*, 122, 109216.

Camacho, E.F. and Bordons, C. (2007). Nonlinear model predictive control: An introductory review. 1–16.

Canale, M., Fagiano, L., and Signorile, M. (2014). Nonlinear model predictive control from data: a set membership approach. *International Journal of Robust and Nonlinear Control*, 24(1), 123–139.

Caruana, R. and Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. 161–168.

Cui, G., Wong, M.L., and Lui, H.K. (2006). Machine learning for direct marketing response models: Bayesian networks with evolutionary programming. *Management Science*, 52(4), 597–612.

Datta, A. (2012). Adaptive internal model control.

Dietterich, T.G. (2000). Ensemble methods in machine learning. 1–15.

Grüne, L. and Pannek, J. (2017). Nonlinear model predictive control. 45–69.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). An introduction to statistical learning. 112.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436–444.

Limon, D., Calliess, J., and Maciejowski, J.M. (2017). Learning-based nonlinear model predictive control. *IFAC-PapersOnLine*, 50(1), 7769–7776.

Lv, L., Duan, G., and Zhou, B. (2010). Parametric pole assignment and robust pole assignment for discrete-time linear periodic systems. *SIAM Journal on Control and Optimization*, 48(6), 3975–3996.

Milanese, M. and Novara, C. (2004). Set membership identification of nonlinear systems. *Automatica*, 40(6), 957–975.

Milanese, M. and Vicino, A. (1991). Optimal estimation theory for dynamic systems with set membership uncertainty: an overview. *Automatica*, 27(6), 997–1009.

Mitchell, T.M. (1997). Machine learning.

Morari, M. and Maeder, U. (2012). Nonlinear offset-free model predictive control. *Automatica*, 48(9), 2059–2067.

Nelles, O. (2013). Nonlinear system identification: from classical approaches to neural networks and fuzzy models.

Raissi, T., Ramdani, N., and Candau, Y. (2004). Set membership state and parameter estimation for systems described by nonlinear differential equations. *Automatica*, 40(10), 1771–1777.

Richardson, M., Prakash, A., and Brill, E. (2006). Beyond pagerank: machine learning for static ranking. 707–715.

Rossiter, J.A. (2017). Model-based predictive control: a practical approach.

Russell, S.J. and Norvig, P. (2016). Artificial intelligence: a modern approach.

Scholte, E. and Campbell, M.E. (2003). A nonlinear set-membership filter for on-line applications. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 13(15), 1337–1358.

Semenov, A., Romov, P., Korolev, S., Yashkov, D., and Neklyudov, K. (2016). Performance of machine learning algorithms in predicting game outcome from drafts in dota 2. 26–37.

Sukharev, A. (1978). Optimal method of constructing best uniform approximations for functions of a certain class. *USSR Computational Mathematics and Mathematical Physics*, 18(2), 21–31.

Thrift, P.R. (1991). Fuzzy logic synthesis with genetic algorithms. 509–513.

Wang, L. (2009). Model predictive control system design and implementation using matlab®.