

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROJETO DE GRADUAÇÃO**



MURILO SANTOLINI DE MENDONÇA

**CONVERSÃO DO MODELO AUTÔMATO HÍBRIDO
DISCRETO PARA O MODELO DE SISTEMAS LÓGICOS
DINÂMICOS MISTOS EM CÓDIGO ABERTO
APLICADO A SISTEMAS REAIS**

VITÓRIA-ES

2022

MURILO SANTOLINI DE MENDONÇA

**CONVERSÃO DO MODELO AUTÔMATO HÍBRIDO
DISCRETO PARA O MODELO DE SISTEMAS LÓGICOS
DINÂMICOS MISTOS EM CÓDIGO ABERTO
APLICADO A SISTEMAS REAIS**

Parte manuscrita do Projeto de Graduação do aluno **Murilo Santolini de Mendonça**, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Orientador: Prof. Dr. Danilo de Paula e Silva

Coorientadora: Profa. Dra. Jussara Farias Fardin

VITÓRIA-ES

2022


MURILO SANTOLINI DE MENDONÇA

CONVERSÃO DO MODELO AUTÔMATO HÍBRIDO DISCRETO PARA O MODELO DE SISTEMAS LÓGICOS DINÂMICOS MISTOS EM CÓDIGO ABERTO APLICADO A SISTEMAS REAIS


Parte manuscrita do Projeto de Graduação do aluno **Murilo Santolini de Mendonça**, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

COMISSÃO EXAMINADORA:


Prof. Dr. Danilo de Paula e Silva
Universidade Federal do Espírito Santo
Orientador

 Documento assinado digitalmente
DANILO DE PAULA E SILVA
Data: 18/08/2022 16:58:33-0300
Verifique em <https://verificador.iti.br>


Profa. Dra. Jussara Farias Fardin
Universidade Federal do Espírito Santo
Coorientadora

 Documento assinado digitalmente
JUSSARA FARIAS FARDIN
Data: 19/08/2022 12:41:30-0300
Verifique em <https://verificador.iti.br>

Prof. Dr. Gustavo Maia de Almeida
Instituto Federal de Educação, Ciência
e Tecnologia do Espírito Santo
Examinador

 Documento assinado digitalmente
GUSTAVO MAIA DE ALMEIDA
Data: 18/08/2022 18:41:59-0300
Verifique em <https://verificador.iti.br>

Me. Guilherme Lima Barbarioli
Examinador

 Documento assinado digitalmente
GUILHERME LIMA BARBARIOLI
Data: 18/08/2022 21:32:04-0300
Verifique em <https://verificador.iti.br>

Aos meus pais e à minha irmã, pelo incentivo e pelo apoio incondicional.

AGRADECIMENTOS

Agradeço imensamente aos meus pais, Vera e Suzano, pela minha criação, pelo exemplo que me deram e pelo incentivo à educação que sempre me proporcionaram. Do mesmo modo, também agradeço aos meus avós, aos meus tios e tias, aos meus primos, aos meus padrinhos e aos demais familiares por todo apoio e torcida. Especialmente, não poderia deixar de agradecer à minha irmã, Júlia, pelo companheirismo e forte presença na minha vida.

Agradeço aos meus amigos pelo momentos compartilhados e pela leveza trazida diante das dificuldades que apareceram ao longo do caminho. Principalmente, aos colegas de classe e aos companheiros de projetos, digo que o percorrimto da formação não se deu sozinho, por isso, o mérito da sua conclusão também deve ser partilhado.

Agradeço à professora Jussara, não só pelas aulas durante a graduação, como também pelo auxílio no projeto e pela indicação do professor Danilo. A ele, sou grato por todo conhecimento, pelo material e por todo tempo disponibilizado, mesmo quando o desenvolvimento do projeto se mostrou mais desafiador do que o esperado.

Finalmente, agradeço à Universidade Federal do Espírito Santo, exemplo de instituição que garante o ensino público, gratuito e de qualidade. Particularmente, sou grato pelos professores e funcionários do Departamento de Engenharia Elétrica e dos demais departamentos com os quais tive contato.

RESUMO

À medida que novas tecnologias são desenvolvidas e novas demandas são criadas, a Ciência precisa acompanhar esses avanços com ferramentas robustas o suficiente para tal. A modelagem de sistemas físicos reais é essencial para simulações, controle, otimização e outras tarefas que envolvem processos químicos, gerenciamento de energia, sistemas embarcados, dentre outros. Em especial, os sistemas híbridos, que se diferenciam por mesclarem variáveis contínuas e discretas, são particularmente úteis para muitas dessas aplicações. Dentre os tipos de modelos híbridos existentes, destacam-se o autômato híbrido discreto (DHA) e os sistemas lógicos dinâmicos mistos (MLD). Devido ao potencial descritivo do primeiro e às possibilidades de aplicações do segundo, faz-se necessário ter ferramentas que permitam a conversão de um modelo em outro. Apesar de já existirem opções para tal, elas dependem de *softwares* pagos e específicos que acabam por dificultar o trabalho realizado em pesquisas e usos em atividades de engenharia. Diante disso, esse projeto visa desenvolver um algoritmo próprio que usa uma linguagem aberta e de fácil uso que seja capaz de cumprir essa função. Como metodologia, foram estudadas as definições gerais do modelos DHA e MLD, bem como os passos necessários para a conversão do primeiro no segundo. Em seguida, os algoritmos e padrões matemáticos identificados foram convertidos em código, utilizando a linguagem *Python* para tal. O algoritmo gerado foi aplicado a um estudo de caso específico, o da microturbina eólica . As matrizes utilizadas pelo modelo MLD obtidas pelo algoritmo desenvolvido foram iguais às obtidas por outro *software* já existente, comprovando sua efetividade.

Palavras-chave: Sistemas híbridos. Autômato híbrido discreto. Sistemas dinâmicos lógicos mistos. Conversão de modelos híbridos.

ABSTRACT

As new technologies are developed and new demands are created, Science needs to follow those advancements with tools robust enough to do so. The modeling of real physical systems is essential for simulations, control, optimization and others tasks that involve chemical processes, energy management, embedded systems, among others. In particular, hybrid systems, which differ from the others due to combining continuous and discrete variables, are particularly useful for many of those applications. Among the types of existing hybrid models, the discrete hybrid automata (DHA) and the mixed logical dynamical systems (MLD) stand out. On account of the descriptive potential of the first one and the application possibilities of the latter, it is necessary to have tools that allow the conversion of a model into another. Even though there are already options for such, they depend on paid and specific softwares that end up making the work done by researchers and users in engineering activities difficult. Therefore, this project aims to develop its own application that uses an open language and of easy use that is capable of fulfilling that function. As for the methodology, the general definitions of the DHA and MLD models were studied, as well as the necessary steps for the conversion of the first into the latter. Then, the algorithms and mathematical patterns identified were converted into code, making use of the *Python* language for such. The generated product was applied to a specific case study, the wind microturbine. The matrices utilized by the MLD model obtained by the developed algorithm were equal to the ones obtained by the other already existing *software*, proving its effectivity.

Keywords: Hybrid systems. Discrete hybrid automata. Mixed logic dynamical systems. Hybrid models conversion.

LISTA DE FIGURAS

Figura 1 – Sistema de reservatório de água	18
Figura 2 – Estrutura do DHA	19
Figura 3 – Curva de operação da turbina eólica com controle de passo	29
Figura 4 – Comparativo entre a curva de potência real da turbina eólica e a obtida através do polinômio interpolador	30
Figura 5 – Fluxograma do modelo híbrido da microturbina eólica	34
Figura 6 – Fluxograma do algoritmo para cálculo das matrizes do modelo MLD	36
Figura 7 – Fluxograma da simulação da microturbina eólica	37
Figura 8 – Potência gerada e velocidade do vento - modelo gerado pelo algoritmo próprio	39
Figura 9 – Estados discretos - modelo gerado pelo algoritmo próprio	39
Figura 10 – Potência gerada e velocidade do vento - modelo gerado pelo HYSDEL	40
Figura 11 – Estados discretos - modelo gerado pelo HYSDEL	40
Figura 12 – Erro absoluto entre cada curva de potência simulada	41

LISTA DE TABELAS

Tabela 1 – Associação entre as linhas das matrizes E geradas pelo HYSDEL e as geradas pelo programa desenvolvido em <i>Python</i>	71
---	----

LISTA DE ABREVIATURAS E SIGLAS

CNF	Forma Normal Conjuntiva
DNF	Forma Normal Disjuntiva
DHA	Autômato Discreto Híbrido
EG	Gerador de Eventos
FSM	Máquina de Estados Finitos
HYSDEL	<i>HYbrid System DEscription Language</i>
MILP	Programação Linear Inteira Mista
MIQP	Programação Quadrática Inteira Mista
MLD	Sistemas Lógicos Dinâmicos Mistos
MPC	Controle Preditivo Baseado em Modelo
MS	Seletor de Modos
PLC	Controlador Lógico Programável
PHEVs	Veículos Elétricos Híbridos <i>Plug-In</i>
PWA	Sistemas Afins por Partes
SAS	Sistema Comutado por Partes
UFES	Universidade Federal do Espírito Santo

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Justificativa	14
1.2	Objetivos Geral e Específicos	15
1.2.1	Objetivo Geral	15
1.2.2	Objetivos Específicos	16
2	REFERENCIAL TEÓRICO	17
2.1	Sistemas Híbridos	17
2.2	Autômato Híbrido Discreto - DHA	19
2.2.1	Sistema Comutador por Partes - SAS	20
2.2.2	Gerador de Eventos - EG	20
2.2.3	Máquina de Estados Finitos - FSM	21
2.2.4	Seletor de Modos - MS	21
2.3	Sistemas Lógicos Mistos Dinâmicos - MLD	21
2.4	HYSDEL	23
2.5	Conversão do Modelo DHA para o MLD	23
3	METODOLOGIA E ETAPAS DE DESENVOLVIMENTO	28
3.1	Modelagem do sistema estudado	28
3.2	Obtenção das matrizes MLD	34
3.3	Simulações do modelo estudado	36
4	RESULTADOS	38
4.1	Matrizes MLD	38
4.2	Simulações do modelo estudado	38
5	CONCLUSÕES E PROJETOS FUTUROS	43
	REFERÊNCIAS	44
	APÊNDICES	46
	APÊNDICE A – MATRIZES DA MODELAGEM MLD	47
A.1	Matrizes geradas pelo HYSDEL	47
A.2	<i>Matrizes geradas pelo algoritmo próprio</i>	59
A.3	Equivalência das matrizes E	70

APÊNDICE B – MODELAGEM DO SISTEMA NO HYSDEL . . . 72

1 INTRODUÇÃO

Dentro do universo da engenharia e das ciências em geral, é comum lançar mão de modelos físicos e matemáticos para estudar sistemas. Esses podem ser definidos como agrupamentos de componentes que, a partir da sua ação conjunta, atingem um certo objetivo, sejam eles físicos, biológicos, econômicos ou outros (OGATA, 2011). Já o processo de modelagem pode ser visto como a representação de um sistema, podendo haver abstrações, uma vez que muitos deles são complexos e não se torna necessário esboçar todas as suas características. Desse modo, é esperado que o modelo adotado seja suficientemente simples, matematicamente condizente e capaz de ser verificado via experimentação (DICKERSON; MAVRIS, 2013).

O escopo, a acurácia e a precisão de um modelo dependem diretamente do tipo utilizado. Enquanto alguns derivam de formulações matemáticas precisas, outros resultam da reprodução do sistema em menor escala. Segundo Wellstead (1979), modelos dinâmicos são baseados em equações diferenciais ou de transformação, podendo ser classificados em modelos de análise dinâmica ou de identificação dinâmica. O primeiro é obtido por meio da análise física de sistemas em um nível fundamental, com simplificações que enquadrem a representação dentro do modelo de equações diferenciais. Em contrapartida, o segundo é obtido através da inferência, fazendo uso de dados estatísticos e da observação do comportamento do sistema.

Esses sistemas dinâmicos podem ser contínuos, discretos, lineares, não-lineares ou outros mais variados. Dentre deste trabalho, busca-se lidar diretamente com sistemas híbridos. Desde a cunhagem mais notória do termo por Witsenhausen (1966), esse tipo de abordagem tem crescido nos últimos anos e é relativamente nova se comparada a campos de estudo mais unicamente direcionados como a Mecânica e a Eletricidade. Em situações em que é preciso fazer uso de ferramentas multidisciplinares, como em sistemas médicos e automotivos, o *design* mecânico, eletrônico e de *software* podem tomar decisões que são vantajosas para uma área do processo em detrimento de outra, cabendo uma tecnologia robusta para combinar aspectos do controle físico e da computação de dados (LUNZE; LAMNABHI-LAGARRIGUE, 2009).

No passado, era mais comum lidar com a evolução de sistemas contínuos e discretos separadamente. No entanto, com o surgimento dos sistemas híbridos, esses provaram ser particularmente úteis em casos onde há interação entre dinâmicas que operam ao longo do tempo e as que variam com base em eventos pontuais, amplamente utilizadas quando se faz uso de atributos como controladores lógico programável (PLC - *Programmable Logic Controller*), máquinas de estados finitos (FSM - *Finite State Machine*) e redes de Petri,

por exemplo. Sobretudo, em controles embarcados, mas, não apenas neles, torna-se mais nítida uma outra vantagem de se usar sistemas híbridos: a hierarquização. A complexidade de sistemas de múltiplas camadas torna mais simples a abstração de modelos detalhados por meio da discretização. Nesse contexto, a interação das componentes discreta e contínua é facilitada se abordada em conjunto e a organização das etapas em uma hierarquia bem definida reduz a complexidade geral do sistema. Exemplos dessa situação podem ser vistos em processos químicos, *smart grids*, sistemas de trânsito inteligentes, controle de tráfego aéreo, redes de comunicação, etc. (ANTSAKLIS, 2000).

Há casos aparentemente menos intuitivos em que também convém usar sistemas dinâmicos. A título de exemplo, em sistemas de colisão, o baixo intervalo de tempo do impacto pode fazer com que ele seja modelado como um evento discreto. Em circuitos eletrônicos, os intervalos de condução ou bloqueio de corrente por um diodo também configuram eventos discretos, tal como ocorre nos tempos de operação de um motor de combustão, dentre outros (LUNZE; LAMNABHI-LAGARRIGUE, 2009). Existem múltiplas maneiras de se modelar um sistema híbrido. Uma delas é o uso do autômato híbrido discreto (DHA - *Discrete Hybrid Automata*). Sob a ótica do DHA, o sistema pode ser dividido em subsistemas bem definidos que dialogam entre si combinando variáveis discretas e contínuas. A formalização dessa segmentação resulta em um forte poder descritivo dessa abordagem. Em virtude dessa característica, o DHA é comumente utilizado para representar sistemas híbridos, contudo, ele nem sempre é usado diretamente, uma vez que ele pode ser convertido para demais modelos que se adequam melhor a tarefas de controle e automação (TORRISI; BEMPORAD, 2004).

Uma das variações de modelos para sistemas híbridos que se enquadra nesse escopo são os sistemas dinâmicos lógicos mistos (MLD - *Mixed Logical Dynamic Systems*). Devido à maneira como essa estrutura é definida, por equações matriciais e inequações envolvendo estados mistos (contínuos e discretos), ela é especialmente propícia para aplicações computacionais, podendo ser implementada com relações *if-then-else*. O modelo pode ser usado para propósitos de estabilização em ferramentas de controle e otimização, incluindo modelos por controle preditivo (MPC - *Model Predictive Control*), cumprindo limitações previamente à medida que segue a trajetória almejada (BEMPORAD; MORARI, 1999).

Em razão da alta capacidade descritiva de sistemas híbridos usando DHA, é prático usar essa abordagem para modelar sistemas. Entretanto, diante dos benefícios já expostos do MLD, faz-se necessário converter um modelo híbrido para o outro. Essa ação já foi desenvolvida por Torrisi e Bemporad (2004) ao apresentarem o HYSDEL (*HYbrid System Description Language*) que faz uso do *software* MATLAB para tal. Apesar disso, a gama

de projetos imersos nessa esfera e a necessidade de autonomia diante de *softwares* externos deixa uma lacuna para métodos alternativos que buscam realizar essa conversão entre diferentes modelos, oportunidade que será discutida neste trabalho.

Neste capítulo são apresentados conceitos gerais sobre modelagem, sistemas híbridos, DHA e MLD. Além disso, são postas as motivações para a realização deste projeto e os objetivos almejados com este trabalho. O Capítulo 2 apresenta a base teórica de conceitos já apresentados no Capítulo 1, porém, de forma mais aprofundada. Já o Capítulo 3 descreve as metodologias e as etapas de desenvolvimento deste projeto, ao passo que, no Capítulo 4, são mostrados os resultados obtidos. Por fim, as conclusões finais são discutidas no capítulo 5.

1.1 Justificativa

Como exposto na seção anterior, o modelo MLD apresenta uma série de vantagens para aplicações envolvendo sistemas híbridos, fazendo com que seu uso tenha ganhado espaço na comunidade científica. De modo a salientar essa tendência, pode-se citar o trabalho de Lian et al. (2017), em que foi desenvolvido um modelo MPC para veículos elétricos híbridos *plug-in* (PHEVs - *Plug-In Hybrid Electrical Vehicles*) fazendo uso do MLD em combinação com programação linear inteira mista (MILP - *Mixed Integer Linear Programming*). A partir desse artigo, foi possível propor uma estratégia de controle de energia que se provou eficaz na melhoria da economia de combustível mediante simulações. Ademais, Zhang et al. (2019), em sua pesquisa, criaram um sistema de modelagem e controle utilizando MLD em conjunto com MILP e programação quadrática inteira mista (MIQP - *Mixed Integer Quadratic Programming*). O foco foi o ajuste dinâmico de um sistema de refrigeração de água para ar-condicionado, onde obteve-se sucesso ao alcançar a temperatura ótima de operação e economizar energia em tempo real.

De fato, existe uma série de trabalhos disponíveis que dispõem do MLD para controle de sistemas dinâmicos. Todavia, o grande motivador para esse projeto de graduação foi a tese de doutorado desenvolvida por Silva (2020). Em vista das perspectivas futuras de microrredes, que podem ser definidas como um grupo de cargas e fontes distribuídas que podem operar conectadas ou não à rede principal (PESIN, 2017), gerenciá-las se torna um processo complexo. Portanto, na tese em questão, foi apresentada uma nova formulação de controle preditivo econômico híbrido para gerenciamento de microrredes. Para tal fim, foram desenvolvidos modelos híbridos originais de banco de baterias, gerador solar fotovoltaica e gerador eólico, utilizando a modelagem MLD.

O grande cerne da questão a ser discutida é como obter os modelos MLD. A ferramenta HYSDEL é uma *toolbox* do MATLAB que, dentre seus diversos atributos, é capaz de gerar um modelo MLD mediante a inserção de parâmetros baseados no modelo DHA (TORRISI; BEMPORAD, 2004). Porém, apesar da capacidade do HYSDEL, utilizado por Silva (2020), ele ainda opera utilizando um *software* que é pago, apresenta um custo computacional considerável e não é *open source* (código aberto). Essas questões ocasionam vários problemas, principalmente quando se trata de implementar modelos híbridos em aplicações em tempo real. Posto isso, essas características impõem limites que impedem desenvolver sistemas práticos que dependam de amostragem simultânea à execução da aplicação.

Buscando contornar essa situação, foi verificado que, no trabalho realizado por Hejri, Giua e Mokhtari (2018), é exibida uma sequência de passos matemáticos para a produção de modelos MLD, bem como variações do mesmo, a partir da definição com base no DHA para sistemas dinâmicos. Analisando o que é discorrido ao longo do respectivo trabalho, pode-se produzir um algoritmo próprio que cumpra a mesma função do HYSDEL, diferenciando-se por não estar atrelado a um *software* externo, em razão dos motivos já descritos. Dessa maneira, é possível ganhar robustez e a emancipação em relação a tecnologias extrínsecas para a continuidade de produções e pesquisas científicas que lançam mão do modelo MLD.

É importante citar também uma outra contribuição do artigo escrito por Hejri, Giua e Mokhtari (2018), o EMLD. Esse modelo apresenta matrizes de dimensões reduzidas, se comparadas às do MLD comum, o que pode diminuir a complexidade dos processos e o tempo levado pelos *softwares* para efetuarem os cálculos necessários.

1.2 Objetivos Geral e Específicos

1.2.1 Objetivo Geral

Este trabalho tem como finalidade a produção de um algoritmo computacional utilizando uma linguagem de código aberto que seja capaz de gerar os argumentos necessários para a modelagem MLD de sistemas dinâmicos.

1.2.2 Objetivos Específicos

- Analisar os fundamentos matemáticos para geração de modelos MLD e sua implementação a partir de modelos DHA;
- Desenvolver um programa computacional *open source* que seja capaz de gerar matrizes e vetores utilizados para modelar sistemas via MLD;
- Aplicar o algoritmo computacional desenvolvido ao caso prático da microturbina eólica;
- Comparar os modelos MLD gerados pelo algoritmo próprio com os produzidos pelo HYSDEL em relação aos valores das matrizes;
- Simular o sistema da microturbina eólica fazendo uso das matrizes do modelo MLD geradas pelo HYSDEL e das produzidas pelo algoritmo próprio, comparando os resultados de cada caso.

2 REFERENCIAL TEÓRICO

2.1 Sistemas Híbridos

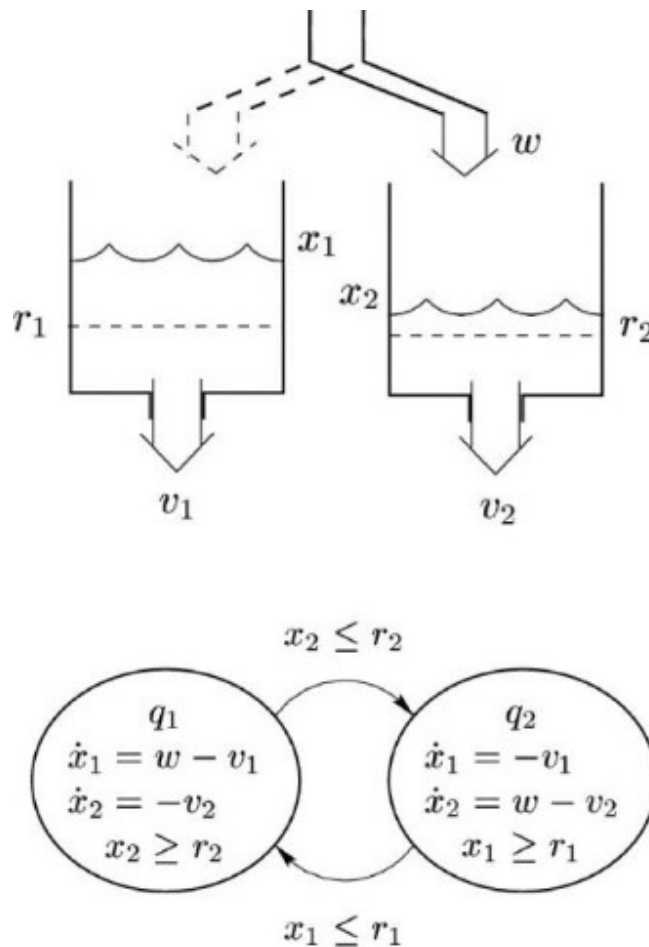
Usualmente, sistemas podem ser classificados como híbridos quando combinam dados e dinâmicas de tipos distintos, destacando-se os casos em que ocorre a interação de dinâmicas contínuas e discretas. Situações do gênero são comuns quando as variáveis envolvidas são alteradas em escalas temporais muito diferentes, como quando certos comportamentos ocorrem em um intervalo de tempo desprezível, podendo ser compreendidos como discretos. Outro fator é a mesclagem de valores analógicos e digitais, presente na ampla maioria dos processos atuais, que fazem uso de circuitos lógicos e recursos computacionais (JOHANSSON; LYGEROS; SASTRY, 2004).

Dentro do universo de sistemas que podem ser modelados utilizando a estrutura híbrida, pode-se citar: colisões mecânicas, processos químicos regulados por válvulas, bioengenharia, manufatura, controle de tráfego, controle automotivo, navegação, robótica e sistemas embarcados em geral (JOHANSSON; LYGEROS; SASTRY, 2004).

Basicamente, sistemas híbridos podem apresentar entradas e saídas contínuas ou discretas, onde eventos previstos mudam o modo de operação definido para cada estado, como em uma FSM. Esses eventos podem ser acionados por meio de interruptores, chaves, condições lógicas ou temporais. Assim, cada modo de operação contém suas próprias regras e equações que ditam a evolução das variáveis ao longo do tempo, geralmente sendo equações diferenciais ou de diferença (BEMPORAD, 2003).

Na Fig. 1, é possível ver um exemplo claro de um sistema híbrido. O objetivo é manter ambos os reservatórios com nível superior a r_1 e r_2 , respectivamente. Para tal, os eventos $x_1 \leq r_1$ e $x_2 \leq r_2$ ativam os estados q_1 e q_2 , onde o abastecimento do tanque é comutado para onde o limite está abaixo do desejado. Dentro de cada estado, são descritas as equações diferenciais que regem os valores das variáveis contínuas, no caso, as alturas x_1 e x_2 dos tanques.

Figura 1 – Sistema de reservatório de água



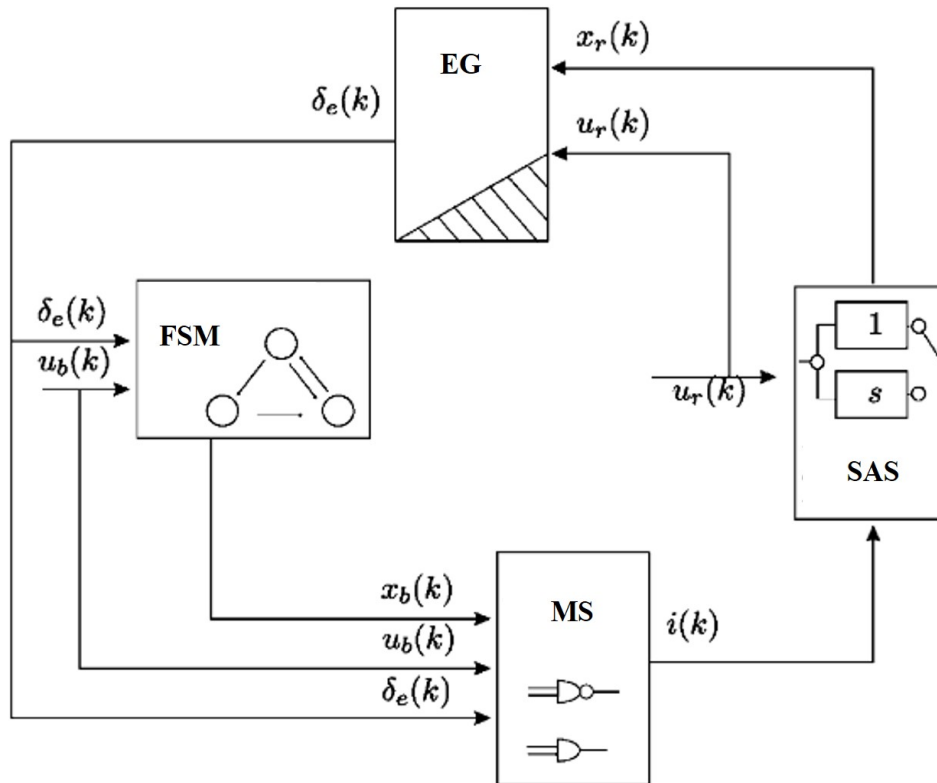
Fonte: Johansson, Lygeros e Sastry (2004)

Existem várias classes de sistemas híbridos possíveis de serem descritas, como as citadas abaixo:

- Sistemas afins por partes (PWA - *Piecewise Affine Systems*) (SONTAG, 1981);
- Autômato híbrido (RASKIN, 2005);
- Autômato híbrido discreto (DHA - *Discrete Hybrid Automata*) (BEMPORAD, 2003);
- Sistemas lógicos mistos dinâmicos (MLD - *Mixed Logical Dynamic*) (BEMPORAD; MORARI, 1999);
- Sistemas lineares por complementaridade (HEEMELS, 1999);
- Sistemas lineares por complementaridade estendidos (SCHUTTER; MOOR, 1997);
- Sistemas *max-min-plus-scaling* (SCHUTTER; BOOM, 2000).

2.2 Autômato Híbrido Discreto - DHA

Figura 2 – Estrutura do DHA



Fonte: Bemporad e Morari (1999)

Nota: Adaptado pelo autor.

O modelo autômato híbrido discreto pode ser compreendido como sendo uma derivação do autômato híbrido, havendo poucas diferenças entre os dois. Em teoria, enquanto o autômato híbrido admite números reais para valores de tempo, o DHA recebe apenas números naturais. Além disso, o DHA é incompatível com transições instantâneas, é um modelo determinístico e não permite comportamentos que apresentam o efeito Zeno, isto é, trocas de estado infinitas em um espaço finito de tempo (BEMPORAD; MORARI, 1999). A respeito da formalização de sua modelo, a Figura 2 exibe os componentes da sua estrutura: o sistema comutado por partes (SAS - *Switched Affine System*), o gerador de eventos (EG - *Event Generator*), o seletor de modos (MS - *Mode Selector*) e a máquina de estados finitos (FSM), que serão descritos em seguida.

2.2.1 Sistema Computador por Partes - SAS

O sistema é definido pelas equações vetoriais de estados e de saídas contínuos:

$$x_r'(k) = A_{i(k)}x_r(k) + B_{i(k)}u_r(k) + f_{i(k)} \quad (2.1a)$$

$$y_r(k) = C_{i(k)}x_r(k) + D_{i(k)}u_r(k) + g_{i(k)} \quad (2.1b)$$

Em (2.1a) e (2.1b), $k \in \mathbb{N}$ indica o tempo, $x_r \in \mathcal{X}_r \subseteq \mathbb{R}^{n_r}$ é o vetor de estados contínuos, $u_r \in \mathcal{U}_r \subseteq \mathbb{R}^{m_r}$ é o vetor de entradas contínuas, $y_r \in \mathcal{Y}_r \subseteq \mathbb{R}^{p_r}$ é o vetor de saídas contínuas, $\{A_i, B_i, f_i, C_i, D_i, g_i\}$ são matrizes de dimensões adequadas e o modo $i(k) \in I := \{1, \dots, s\}$ é um sinal de entrada proveniente do gerador de eventos que define a dinâmica contínua a ser seguida. O estado é preservado quando ocorre a comutação, no entanto, é possível implementar um mapa de reconfiguração (*reset*) (BEMPORAD, 2003). A troca das dinâmicas é mais evidente na visualização a seguir:

$$z_1(k) = \begin{cases} A_1x_r(k) + B_1u_r(k) + f_1, & \text{if } (i(k) = 1), \\ 0, & \text{caso contrário,} \end{cases} \quad (2.2a)$$

⋮

$$z_s(k) = \begin{cases} A_sx_r(k) + B_su_r(k) + f_s, & \text{if } (i(k) = s), \\ 0, & \text{caso contrário,} \end{cases} \quad (2.2b)$$

$$x_r'(k) = \sum_{i=1}^s z_i(k), \quad (2.2c)$$

Onde $z_i(k) \in \mathbb{R}^{n_r}$ e $i = 1, \dots, s$.

2.2.2 Gerador de Eventos - EG

O gerador de eventos é responsável por gerar um sinal lógico $\delta_e(k)$ em função de $x_r(k)$, $u_r(k)$ e k , a partir do vetor de funções $f_H \in \mathcal{D} \subseteq \{0, 1\}^{n_e}$. Os eventos de tempo são definidos tais que $[\delta_i^e(k) = 1] \leftrightarrow [kT_s \geq t_0]$, sendo T_s o tempo de amostragem e t_0 o tempo dado.

2.2.3 Máquina de Estados Finitos - FSM

Uma FSM é responsável por gerar o próximo vetor de estados binários $x'_b(k) \in \mathcal{X}_b \subseteq \{0, 1\}^{n_b}$ e, caso exista, o vetor de saídas binárias $y'_b(k) \in \mathcal{Y}_b \subseteq \{0, 1\}^{p_b}$. Os vetores de funções f_B e g_B responsáveis por esses vetores e dependem do vetor de estados binários $x_b(k)$, do vetor de entradas binárias $u_b(k)$ e de δ_e .

2.2.4 Seletor de Modos - MS

O MS recebe como entradas $x_b(k)$, $u_b(k)$ e δ_e , gerando, como resultado, a dinâmica $i(k)$ a ser seguida pelo SAS. Caso a dinâmica para um instante atual seja diferente do instante anterior, ou seja, se $i(k) \neq i(k-1)$, ocorre uma comutação de modos, supondo que a mudança ocorra no tempo de amostragem definido.

2.3 Sistemas Lógicos Mistos Dinâmicos - MLD

Os sistemas MLD são modelos de sistemas híbridos que mesclam conjuntos de equações lineares de diferença e inequações restritivas que relacionam tanto variáveis contínuas quanto discretas. Uma vez que o MLD apresenta equações que unem os componentes do DHA, ele se torna mais simples de ser aplicado, em especial, para usos computacionais. As equações que o definem podem ser descritas abaixo:

$$x'(k) = Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k) + B_4, \quad (2.3a)$$

$$y(k) = Cx(k) + D_1u(k) + D_2\delta(k) + D_3z(k) + D_4, \quad (2.3b)$$

$$E_2\delta(k) + E_3z(k) \leq E_1u(k) + E_4x(k) + E_5, \quad (2.3c)$$

$$\bar{E}_2\delta(k) + \bar{E}_3z(k) = \bar{E}_1u(k) + \bar{E}_4x(k) + \bar{E}_5. \quad (2.3d)$$

Ao contrário dos vetores contínuos e binários separados presentes no DHA, aqui, tem-se $x \in \mathbb{R}^{n_r} \times \{0, 1\}^{n_b}$ e $y \in \mathbb{R}^{p_r} \times \{0, 1\}^{p_b}$. Além disso, $z \in \mathbb{R}^{r_r}$, $\delta \in \{0, 1\}^{r_b}$ e $A, B_1, B_2, B_3, C, D_1, D_2, D_3, E_1, \dots, E_5$ são matrizes de dimensões adequadas (BEMPORAD, 2003).

Outros atributos associados ao modelo MLD são a inicialização dos estados $Init$, os grupos de invariantes Inv , as guardas controladas G_c , as guardas descontroladas G_{uc} , os eventos controlados E_c , os eventos descontrolados E_{uc} e o *reset* R_{xc} (HEJRI; GIUA; MOKHTARI, 2018).

A inicialização $Init$ é definida como os valores iniciais dos vetores de estados x_r e x_b na partida do sistema. Já os invariantes Inv dizem respeito às condições atendidas pelos estados e entradas quando o sistema se encontra em um modo específico. De forma análoga, as guardas G_c e G_{uc} indicam as condições dos estados e entradas para que haja troca de modo pelo sistema. As guardas apresentam como parâmetro os eventos E_c e E_{uc} , caracterizados pela passagem de um modo para outro. A diferença entre eventos e guardas controlados ou descontrolados encontra-se na origem da condição. Se ela é derivada do valor de estados e entradas recebidos, tem-se uma guarda ou evento descontrolado. Contudo, se a condição é definida externamente pelo usuário ou mediante uma ação direta no sistema que independe dos estados e entradas, há uma guarda ou evento controlado. A lógica de *reset* R_{xc} descreve o retorno do sistema a um estado específico mediante um determinado evento.

O MLD assume que o sistema é bem-posto, isto é, ele apresenta existência, unicidade e dependência da solução em relação às condições iniciais e de contorno. Outrossim, o MLD possui algumas especificidades, como a incorporação de fatores ϵ nas equações e inequações, números positivos pequenos que representam a precisão adotada para o processo a ser resolvido computacionalmente.

A estrutura com a qual um sistema MLD é definido mostra o que já foi discutido anteriormente: sua praticidade e seu potencial de aplicação computacional. No entanto, assim como também já foi apresentado, o DHA é organizado de uma maneira mais fácil de ser descrita, o que torna desejável métodos para a conversão de um modelo para o outro. A possibilidade dessa permutação é exposta a seguir:

Seja \sum_{DHA} um modelo DHA bem-posto definido em um conjunto de estados $\mathcal{X} \subseteq \mathbb{R}_m$, e um conjunto de saídas $\mathcal{Y} \subseteq \mathbb{R}_p$. Logo para qualquer conjunto limitado $\bar{\mathcal{X}}, \bar{\mathcal{U}}$, existe um modelo bem-posto MLD \sum_{DHA} equivalente a \sum_{DHA} em $\bar{\mathcal{X}}, \bar{\mathcal{U}}, \mathcal{Y}$. (BEMPORAD, 2003) (tradução do autor).

Um ponto a ser ressaltado é a apresentação de um novo modelo derivado do MLD por Hejri, Giua e Mokhtari (2018), o EMLD, ou MLD estendido. Nesse novo modelo, as dimensões dos vetores e matrizes utilizados pelo MLD são reduzidas, o que pode resultar em um maior desempenho computacional ao fazer uso do EMLD para estudo de sistemas híbridos.

2.4 HYSDEL

O HYSDEL foi apresentado por Torrisi e Bemporad (2004) como uma ferramenta de modelagem e conversão de sistemas híbridos. Como existe uma variedade de modelos de sistemas híbridos, cada um apresenta uma vantagem de acordo com a aplicação almejada. Enquanto alguns são mais propícios para descrição ou simulação, como o DHA, outros são mais adaptados a estudos de verificação de estabilidade, identificação e verificação, como o PWA. Outros ainda são mais vantajosos para objetivos do tipo detecção de faltas, estimativa e controle, tal qual o MLD.

Essa possibilidade de intercâmbio fornecida pelo HYSDEL é de grande utilidade, assim como a maneira de utilizá-lo. Mais especificamente, a ferramenta pode ser definida como uma linguagem textual e descritiva de alto nível para estabelecimento de sistemas com modelos que dialogam entre si.

A versão levada em consideração neste trabalho é a 2.0.6 (TORRISI et al., 2011). Em relação a sua sintaxe, a escrita é dividida em duas partes: *INTERFACE* e *IMPLEMENTATION*. A primeira porção é reservada para a declaração de parâmetros e variáveis, já a segunda é onde são descritos os processos que relacionam essas variáveis. Essa descrição é baseada no DHA, mas o sistema pode ser traduzido para PWA, MLD, sistemas lineares por complementaridade (extendidos ou não) e sistemas *max-min-plus-scaling*. O acervo de funções do HYSDEL também permite a simulação do sistema definido, com opções variadas para cálculo dos vetores a partir das matrizes encontradas (TORRISI; BEMPORAD, 2004).

2.5 Conversão do Modelo DHA para o MLD

Ao passo que Torrisi e Bemporad (2004) tenham formulado um método para a conversão do modelo DHA para o MLD e produzido uma aplicação que realize essa função (HYSDEL), os passos para esse processo não foram explicitados completamente. Já, no trabalho realizado por Hejri, Giua e Mokhtari (2018), existe um desenvolvimento teórico matemático para essa conversão.

Além disso, ambos os modelos DHA e MLD contém uma gama de definições, lemas e teoremas que descrevem o funcionamento e as relações existentes entre os dois modelos, que são descritos neste trabalho. A composição desses fatores ajuda a definir lógicas de combinações dos parâmetros dos vetores definidos no sistema MLD juntamente com as equações e inequações que o descrevem, sendo possível, portanto, obter as matrizes do

modelo MLD.

Os passos de obtenção das matrizes $A, B_1, B_2, B_3, B_4, B_5, C, D_1, D_2, D_3, D_4$ e D_5 (matrizes de igualdade), envolvem as equações de atualização das saídas e dos próximos estados, que dependem das entradas, dos estados atuais e dos parâmetros estáticos pré-definidos.

Para tal, são utilizadas as equações (2.1a), (2.1b), (2.2a), (2.2b) e (2.2c). As matrizes A_i e B_i que multiplicam os vetores de estados e entradas são utilizadas para comporem as matrizes de igualdade A e B_1 nas linhas e colunas onde ocorre a multiplicação com cada respectivo vetor. Os vetores que não fazem parte da equação acarretam coeficientes nulos das matrizes, ao passo que os modos definidos no MS, juntamente com a lógica SAS, indicam os coeficientes das variáveis auxiliares z_i que integram cada modo para determinação do próximo estado, utilizados para encontrar a matriz B_3 . As definições apresentadas na FSM são utilizadas para compor a matriz B_2 , ao passo que os valores constantes incluídos no SAS formam a matriz B_4 .

No que tange às matrizes E_1, E_2, E_3, E_4 e E_5 (matrizes de desigualdade), faz-se necessário analisar as lógicas dispostas em cada componente do DHA para a geração de inequações que descrevam as restrições do comportamento do sistema.

No EG, são descritas as equações que determinam o valor lógico discreto de cada variável do vetor $\delta(k)$ mediante uma inequação em função dos estados e das entradas contínuos. Para que sejam geradas as inequações gerais que definem essa relação, faz-se necessário utilizar o conceito de programação linear, segundo descrito por Hejri, Giua e Mokhtari (2018). É definida uma tolerância ϵ , relacionada intrinsecamente à precisão computacional adotada para que os cálculos sejam realizados. A formulação para a geração das inequações pode ser vista a seguir:

$$[\delta_{ci}(k) = 1] \leftrightarrow H_i x_r(k) + J_i u_r(k) \leq K_i \quad (2.4)$$

$$M_i = \max(H_i x_r(k) + J_i u_r(k) - K_i) \quad (2.5)$$

$$m_i = \min(H_i x_r(k) + J_i u_r(k) - K_i) \quad (2.6)$$

$$H_i x_r(k) + J_i u_r(k) - K_i - M_i(1 - \delta_{ci}(k)) \leq 0 \quad (2.7)$$

$$-H_i x_r(k) - J_i u_r(k) + K_i + \epsilon + (m_i - \epsilon)\delta_{ci}(k) \leq 0 \quad (2.8)$$

Nas equações acima, K_i é uma matriz de constantes e $max()$ e $min()$ são funções genéricas que retornam os valores máximo e mínimo possíveis para a expressão contida entre parênteses. Os demais termos utilizados já foram apresentados.

Ademais, é preciso utilizar as equações lógicas descritas na FSM e no EG com o objetivo de se adquirir as inequações que regem esses módulos. Diferentemente das inequações em função de valores contínuos, neste ponto, são buscadas as equações que definem as variáveis discretas a partir da combinação de outras variáveis discretas.

É fundamental citar que as técnicas matemáticas que serão mostradas até o fim desta seção foram propostas pelo próprio autor. Seu desenvolvimento deu-se pelos estudos acerca dos modelos DHA e MLD em conjunto com exemplos e testes realizados com auxílio do HYSDEL.

O método estudado e desenvolvido para realizar essa conversão constitui-se na obtenção da equação lógica na sua forma normal conjuntiva (CNF - *Conjunctive Normal Form*) e na sua forma normal disjuntiva (DNF - *Disjunctive Normal Form*). A CNF de uma equação lógica a reduz a uma agrupamento de operações lógicas E de operações lógicas OU. Simetricamente, a DNF de uma equação lógica a reduz a uma agrupamento de operações lógicas OU de operações lógicas E (HOOS; STÜTZLE, 2005). É crucial afirmar que o método aplicado consiste na transformação das equações lógicas em inequações mistas inteiras tanto convertendo-as em CNF, quanto em DNF.

O processo para obtenção das inequações que correspondem às equações do tipo CNF pode ser visto adiante. São descritas o expressões l que possuem elementos que se relacionam por meio da operação lógica E. Dentro de cada conjunto de variáveis que se relacionam por meio da operação lógica OU, os m elementos i representam as variáveis não negadas, enquanto os n elementos j representam as variáveis as negadas. Entende-se variável negada por uma variável v qualquer onde ela é disposta como \bar{v} .

$$\delta_r = \bigwedge_{l=1}^o \left(\bigvee_{i=1}^m \delta_{il} \vee \bigvee_{j=1}^n \delta_{jl} \right)_l \quad (2.9)$$

$$\begin{cases} \sum_{i=1}^m (-\delta_{il=1}) + \sum_{j=1}^n (\delta_{jl=1}) + \delta_r \leq n_{l=1} \\ \vdots \\ \sum_{i=o}^m (-\delta_{il=o}) + \sum_{j=o}^n (\delta_{jl=o}) + \delta_r \leq n_{l=o} \end{cases} \quad (2.10)$$

A seguir, vê-se o processo para obtenção das inequações que correspondem às equações do tipo DNF. São descritas o expressões l que possuem elementos que se relacionam por meio da operação lógica OU. Dentro de cada conjunto de variáveis que se relacionam por meio da operação lógica E, os m elementos i representam variáveis não negadas, enquanto os n elementos j representam as variáveis negadas.

$$\delta_r = \bigvee_{l=1}^o \left(\bigwedge_{i=1}^m \delta_{il} \wedge \bigwedge_{j=1}^n \delta_{jl} \right)_l \quad (2.11)$$

$$\left\{ \begin{array}{l} \sum_{i=1}^m (\delta_{il=1}) + \sum_{j=1}^n (-\delta_{jl=1}) - \delta_r \leq m_{l=1} - 1 \\ \vdots \\ \sum_{i=o}^m (\delta_{il=o}) + \sum_{j=o}^n (-\delta_{jl=o}) - \delta_r \leq m_{l=o} - 1 \end{array} \right. \quad (2.12)$$

As equações descritas 2.2a, 2.2b e 2.2c definem as lógicas de *IF-THEN-ELSE*, aplicadas às variáveis z_i . Essas expressões também devem ser convertidas em inequações com lógica mista. O processo pode ser resumido na formulação desenvolvida abaixo:

$$x = IF \text{ boolexp } THEN \text{ exp}_1 \text{ ELSE } \text{ exp}_2 \quad (2.13)$$

$$M_1 = \max(\text{exp}_1) \quad (2.14)$$

$$M_2 = \max(\text{exp}_2) \quad (2.15)$$

$$m_1 = \min(\text{exp}_1) \quad (2.16)$$

$$m_2 = \max(\text{exp}_2) \quad (2.17)$$

$$(M_1 - m_2)\text{boolexp} + \text{exp}_1 - x \leq (M_1 - m_2) \quad (2.18)$$

$$(M_2 - m_1)boolexp - exp_1 + x \leq (M_2 - m_1) \quad (2.19)$$

$$-(M_2 - m_1)boolexp + exp_1 - x \leq 0 \quad (2.20)$$

$$-(M_1 - m_2)boolexp - exp_2 + x \leq 0 \quad (2.21)$$

No equacionamento mostrado acima, exp_1 e exp_2 representam uma variável ou constante contínua qualquer e $boolexp$ é uma expressão lógica booleana. As funções $max()$ e $min()$ têm o mesmo funcionamento que apresentam nas equações (2.5) e (2.6).

Após serem determinadas todas as inequações buscadas para o detalhamento do funcionamento do sistema definido na modelagem DHA, deve ser analisada a correspondência de cada variável da inequação com a sua respectiva posição do vetor que multiplica cada matriz E . Dessa maneira, pode-se determinar o coeficiente a ser atribuído à cada coluna para uma linha da matriz. Assim, repetindo o processo para cada inequação, completam-se as matrizes de desigualdade. Vale ressaltar que inequações iguais são desnecessárias, cabendo eliminar múltiplos registros, caso hajam.

3 METODOLOGIA E ETAPAS DE DESENVOLVIMENTO

De modo a alcançar os objetivos desejados, foram estudados os modelos híbridos DHA e MLD. A fim de poder definir de maneira acurada o modelo em DHA, foi analisada a estrutura utilizada no HYSDEL (TORRISI et al., 2011). Já, para a conversão em MLD, foram analisada a teoria descrita por Hejri, Giua e Mokhtari (2018). A combinação dessas duas fontes juntamente com exemplos de sistemas e o próprio uso do HYSDEL para geração de matrizes como exemplo foi o que possibilitou o desenvolvimento de um programa em *Python*(ROSSUM; JR, 1995) que retornasse as matrizes MLD para os sistemas escolhidos. A escolha da linguagem *Python* deu-se pela sua natureza *open source*, sua variedade de bibliotecas, seu crescente uso em atividades de pesquisa e de engenharia e sua facilidade de interação com diversos ambientes. Além disso, foi necessário definir um sistema em que fosse possível aplicar o modelo DHA e MLD. Para tal, dispôs-se do modelo de microturbina eólica descrito por Silva (2020).

As etapas de desenvolvimento são descritas com maior detalhe em seguida.

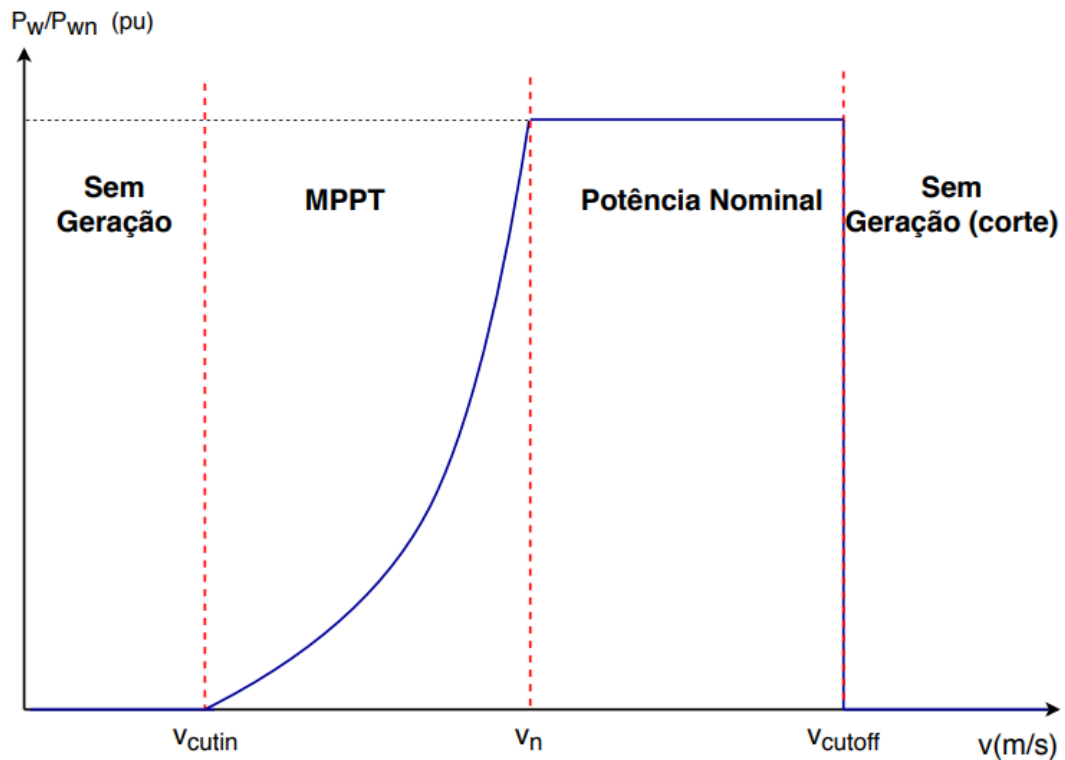
3.1 Modelagem do sistema estudado

O modelo de turbina eólica utilizado foi o GERAR246 da Enersud de 1000 W. Ele apresenta uma tensão de saída de 220 V e 2,46 m de diâmetro de pá. Segundo as especificações do fornecedor, ela visa atender às exigências de operação em sítios de telecomunicações. O equipamento é considerado robusto, já que mantém a geração mesmo com ventos de baixa velocidade e tem a capacidade de conservar a geração sob ventos de alta velocidade (ENERSUD, 2017).

A turbina opera com controle de passo, a partir do qual definem-se as regiões em que ela opera, conforme é mostrado na figura 3. Com base nessa curva, Silva (2020) desenvolveu uma função polinomial através da interpolação de pontos do gráfico. A comparação entre a curva fornecida na documentação da turbina e a produzida pelo polinômio gerado pode ser vista na Figura 4. O polinômio caracteriza a operação da turbina em regime de máxima eficiência (MPPT),

O sistema de microturbina apresenta eólica duas entradas contínuas, a potência de operação P_g , em W, e a velocidade do vento vel , em m/s. A potência de operação é descrita através do polinômio que define seus valores na região de máxima eficiência. Como o modelo DHA aceita apenas funções afins, a potência de operação deve ser calculada externamente e incluída como parâmetro de entrada. Quanto aos estados contínuos, existe apenas a

Figura 3 – Curva de operação da turbina eólica com controle de passo



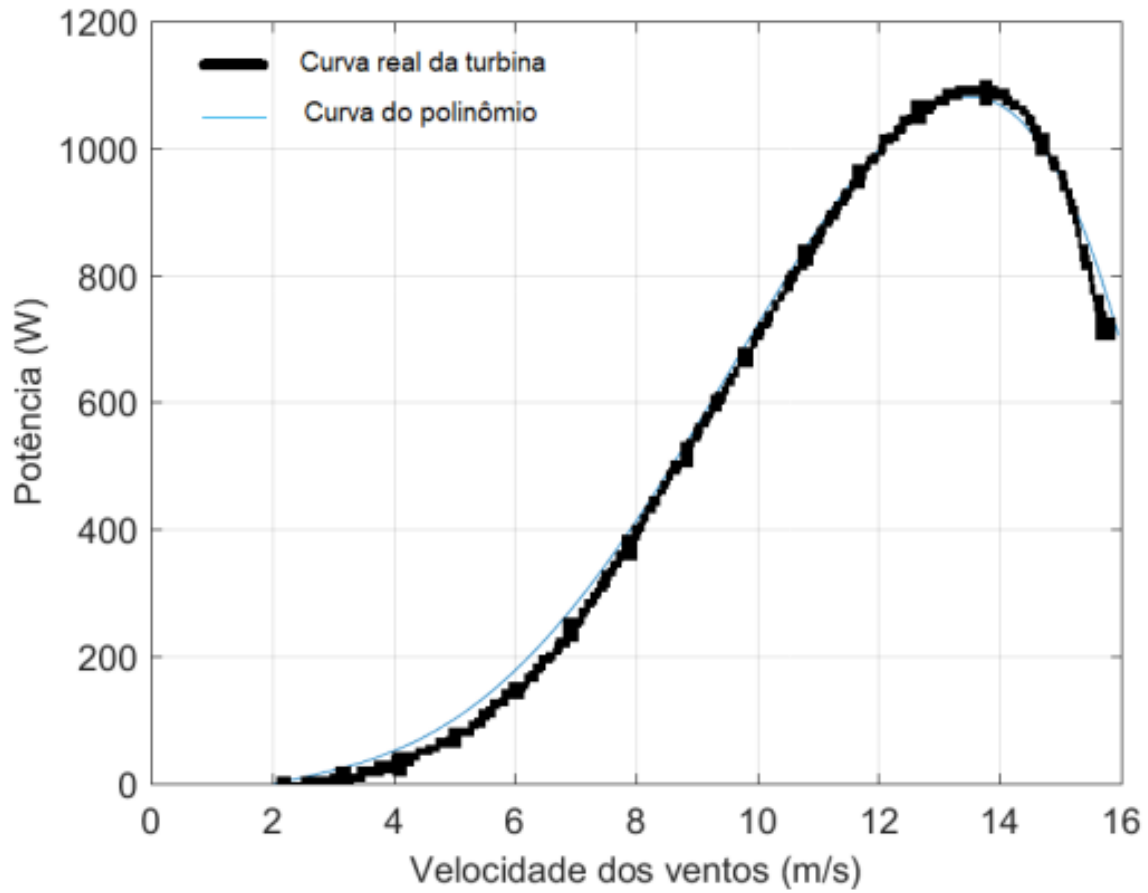
Fonte: (SILVA, 2020).

potência atual da microturbina, indicada por P_w , em W.

Os estados discretos são divididos em quatro: *parado*, *operando*, *nominal* e *corte*. Cada um deles indica a região de operação da microturbina. No estado parado, não há geração. A partir de uma certa velocidade, definida como v_{cutin} , a microturbina entra no estado *operando*, onde a potência é dada por P_g . A partir da velocidade nominal v_n , o sistema entra em regime nominal, apresentando uma potência fixa P_n . Caso a velocidade exceda a velocidade definida como v_{cutoff} , o valor da potência decai para um valor constante residual P_{cutoff} . São definidas variáveis auxiliares d_1 , d_2 , d_3 , op e pn . Elas são utilizadas para, com base na transição de velocidades, chavear o sistema para um certo estado de operação (SILVA, 2020).

Com base na teoria posta por Bemporad e Morari (1999) e Hejri, Giua e Mokhtari (2018), pode-se definir o modelo DHA do sistema descrito.

Figura 4 – Comparativo entra a curva de potência real da turbina eólica e a obtida através do polinômio interpolador



Fonte: (SILVA, 2020).

$$SAS : x_r(k + 1) = \begin{cases} 0 \leftrightarrow [i(k) = 1] & (3.1a) \\ P_g \leftrightarrow [i(k) = 2] & (3.1b) \\ P_n \leftrightarrow [i(k) = 3] & (3.1c) \\ P_{cutoff} \leftrightarrow [i(k) = 4] & (3.1d) \end{cases}$$

$$EG = \begin{cases} [d_1 = 1] \leftrightarrow [vel \geq v_{cutin}] & (3.2a) \\ [d_2 = 1] \leftrightarrow [vel \geq v_n] & (3.2b) \\ [d_3 = 1] \leftrightarrow [vel \geq v_{cutoff}] & (3.2c) \\ [op = 1] \leftrightarrow (d_1 \wedge \bar{d}_2) & (3.2d) \\ [pn = 1] \leftrightarrow (d_2 \wedge \bar{d}_3) & (3.2e) \end{cases}$$

$$FSM = \begin{cases} [parado = 1][\leftrightarrow (\bar{d}_1 \wedge operando) \vee (parado \wedge \bar{d}_1) \\ \vee (corte \wedge \bar{d}_1) \vee (nominal \wedge \bar{d}_1)] & (3.3a) \\ [operando = 1][\leftrightarrow (operando \wedge op) \vee (parado \wedge d_1) \\ \vee (nominal \wedge \bar{d}_2) \vee (corte \wedge op)] & (3.3b) \\ [nominal = 1][\leftrightarrow (nominal \wedge pn) \vee (operando \wedge d_2) \\ \vee (corte \wedge \bar{d}_3) \vee (parado \wedge pn)] & (3.3c) \\ [corte = 1][\leftrightarrow (corte \wedge d_3) \vee (nominal \wedge d_3) \\ \vee (operando \wedge d_3) \vee (parado \wedge d_3)] & (3.3d) \end{cases}$$

$$MS : i(k) = \begin{cases} 1 \leftrightarrow [parado = 1] & (3.4a) \\ 2 \leftrightarrow [operando = 1] & (3.4b) \\ 3 \leftrightarrow [nominal = 1] & (3.4c) \\ 4 \leftrightarrow [corte = 1] & (3.4d) \end{cases}$$

Diante da modelagem apresentada, podem ser definidos vetores, matrizes, eventos e variáveis que a compõem.

O vetor de estados contínuos x_r é composto por apenas um estado x_{r1} , igual a P_w , que representa a potência gerada pela turbina. Dessa maneira, define-se o seu tamanho $n_r = 1$. Quanto aos limites inferior e superior definidos para P_w , tem-se que $P_w \in \mathbb{R} \mid 0 \leq P_w \leq 1500$.

É importante citar que a turbina opera com valor de potência entre 0 W e 1000 W. No entanto, o modelo MLD escolhido leva um passo para recalcular os próximos valores, o que faz com que a ultrapassagem do valor máximo pelo polinômio acarrete um estado fora dos limites para um determinado k . Em virtude disso, u_{r1} e x_{r1} têm seus limites máximos aumentados para além da potência máxima real. Esse processo é conhecido como relaxamento de restrições.

O vetor de estados discretos x_b é composto por quatro estados, x_{b1} , x_{b2} , x_{b3} , x_{b4} , onde $x_{b1} = parado$, $x_{b2} = operando$, $x_{b3} = nominal$, $x_{b4} = corte$. No caso, o tamanho do vetor n_b é o igual ao número de estados discretos s , 4. Assim, cada modo pode ser definido pelos vetores $[1 \ 0 \ 0 \ 0]^T$, $[0 \ 1 \ 0 \ 0]^T$, $[0 \ 0 \ 1 \ 0]^T$ e $[0 \ 0 \ 0 \ 1]^T$.

O sistema conta com duas entradas contínuas. Logo, tem-se $u_r = [u_{r1} \ u_{r2}]^T$, onde $u_{r1} = P_g$ e $u_{r2} = vel$. Os limites das variáveis são, respectivamente, $P_g \in \mathbb{R} \mid 0 \leq P_w \leq 1500$ e

$vel \in \mathbb{R} \mid 0 \leq P_w \leq 50$. Disso, tem-se que o tamanho do vetor u_r é $m_r = 2$. Como não há entradas contínuas, $u_b = []$ e $m_b = 0$.

O vetor δ é composto pelos auxiliares discretos e pelos próximos estados discretos. Portanto, define-se que $\delta_{aux} = [\delta_{aux1} \delta_{aux2} \delta_{aux3} \delta_{aux4} \delta_{aux5}]^T$, onde $\delta_{aux1} = d_1$, $\delta_{aux2} = d_2$, $\delta_{aux3} = d_3$, $\delta_{aux4} = op$ e $\delta_{aux5} = pn$. Logo, o tamanho do vetor δ_{aux} é $a_b = 5$. Os próximos estados discretos são definidos como $\delta_f = [\delta_{f1} \delta_{f2} \delta_{f3} \delta_{f4}]^T$. O vetor δ é, por conseguinte, dado por $\delta = [\delta_{aux}^T \delta_f^T]^T$.

O vetor z contém as variáveis contínuas auxiliares utilizadas pela componente SAS do modelo MLD. Ele é dado por $z = [z_1 \ z_2 \ z_3 \ z_4]^T$, onde $z_1 = 0$, $z_2 = P_g$, $z_3 = P_n$ e $z_4 = P_{cutoff}$, com tamanho $a_r = 4$.

As variáveis estáticas são definidas como $P_n = 1000W$, $P_{cutoff} = 50W$, $v_{cutin} = 2m/s$, $v_n = 12m/s$ e $v_{cutoff} = 16m/s$.

O sistema é definido como partindo do estado discreto *parado* e com o estado contínuo zerado. Portanto, tem-se que $Init = \{(x_r = [0]), (x_b = [1 \ 0 \ 0 \ 0]^T)\}$.

Os invariantes são definidos pela relação entre a velocidade vel , tida como u_{r2} e os limites pré-definidos. Isso implica as equações expostas a seguir:

$$Inv(x_b = [1 \ 0 \ 0 \ 0]^T) = \{x_r(k), u_r(k)\} \mid u_r(k) \leq v_{cutin} \quad (3.5)$$

$$Inv(x_b = [0 \ 1 \ 0 \ 0]^T) = \{x_r(k), u_r(k)\} \mid v_{cutin} < u_r(k) \leq v_n \quad (3.6)$$

$$Inv(x_b = [0 \ 0 \ 1 \ 0]^T) = \{x_r(k), u_r(k)\} \mid v_n < u_r(k) < v_{cutoff} \quad (3.7)$$

$$Inv(x_b = [0 \ 0 \ 0 \ 1]^T) = \{x_r(k), u_r(k)\} \mid u_r(k) \geq v_{cutoff} \quad (3.8)$$

Quanto aos eventos possíveis. Foi definido que qualquer um dos modos pode ser seguido por qualquer outro no próximo instante k . Dessa maneira, os eventos descontrolados englobam todas as mudanças de modo, ou seja: $E_{uc} = \{(parado, parado) \vee (parado, operando) \vee (parado, nominal) \vee (parado, corte) \vee (operando, parado) \vee (operando, operando)\}$

\vee (*operando, nominal*) \vee (*operando, corte*) \vee (*nominal, parado*) \vee (*nominal, operando*) \vee (*nominal, nominal*) \vee (*nominal, corte*) \vee (*corte, parado*) \vee (*corte, operando*) \vee (*corte, nominal*) \vee (*corte, corte*) \vee }. Não há eventos controlados, já que a turbina depende de fenômenos meteorológicos que independem de qualquer ação externa.

Assim como os invariantes, as guardas descontroladas são definidas pela mudanças da velocidade do vento. Não há guardas controladas estabelecidas, logo, tem-se G_{uc} é composto por:

$$g_{uc}(\textit{parado}, \textit{operando}) = \{(x_r(k), u_r(k)) \mid u_r(k) \geq v_{cutin}\} \quad (3.9)$$

$$g_{uc}(\textit{parado}, \textit{nominal}) = \{(x_r(k), u_r(k)) \mid (u_r(k) \geq v_n) \wedge (u_r(k) < v_{cutoff})\} \quad (3.10)$$

$$g_{uc}(\textit{parado}, \textit{corte}) = \{(x_r(k), u_r(k)) \mid u_r(k) \geq v_{cutoff}\} \quad (3.11)$$

$$g_{uc}(\textit{operando}, \textit{parado}) = \{(x_r(k), u_r(k)) \mid u_r(k) < v_{cutin}\} \quad (3.12)$$

$$g_{uc}(\textit{operando}, \textit{nominal}) = \{(x_r(k), u_r(k)) \mid u_r(k) \geq v_n\} \quad (3.13)$$

$$g_{uc}(\textit{operando}, \textit{corte}) = \{(x_r(k), u_r(k)) \mid u_r(k) \geq v_{cutoff}\} \quad (3.14)$$

$$g_{uc}(\textit{nominal}, \textit{parado}) = \{(x_r(k), u_r(k)) \mid u_r(k) < v_{cutin}\} \quad (3.15)$$

$$g_{uc}(\textit{nominal}, \textit{operando}) = \{(x_r(k), u_r(k)) \mid u_r(k) < v_n\} \quad (3.16)$$

$$g_{uc}(\textit{nominal}, \textit{corte}) = \{(x_r(k), u_r(k)) \mid u_r(k) \geq v_{cutoff}\} \quad (3.17)$$

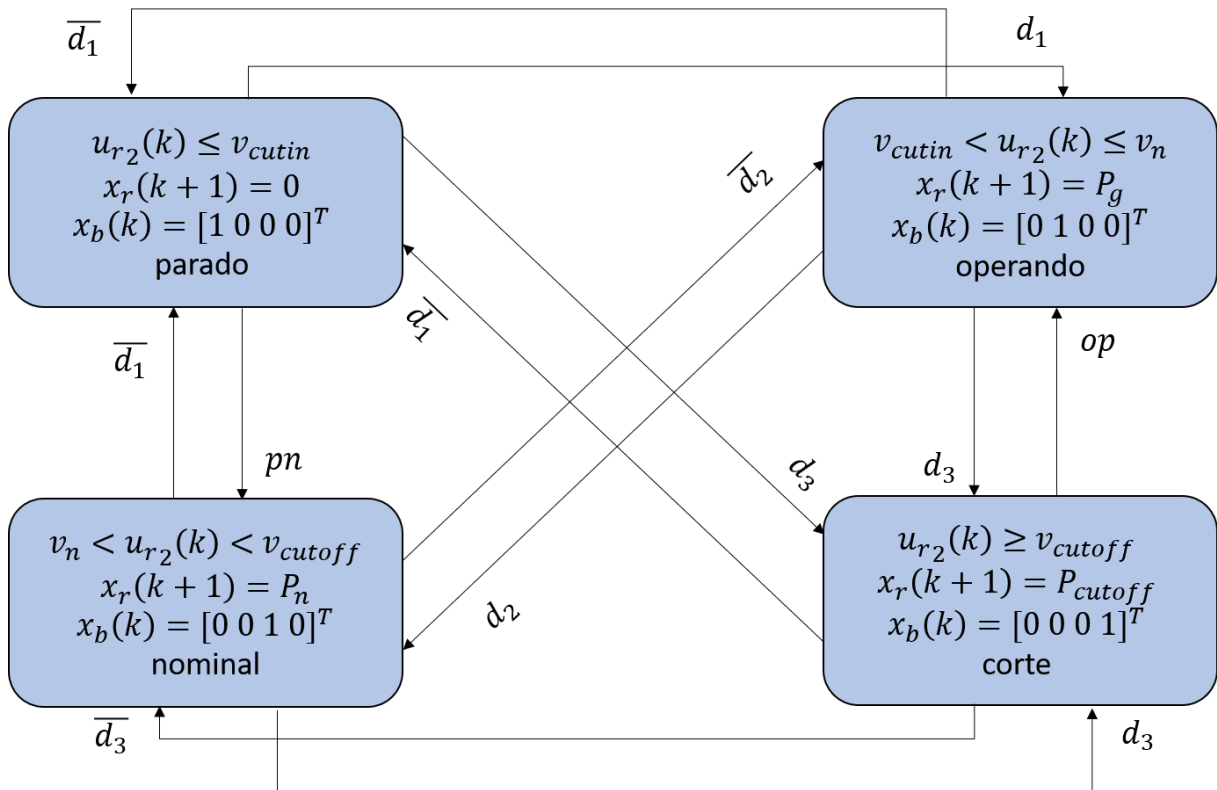
$$g_{uc}(corte, parado) = \{(x_r(k), u_r(k)) \mid u_r(k) < v_{cutin}\} \quad (3.18)$$

$$g_{uc}(corte, operando) = \{(x_r(k), u_r(k)) \mid (u_r(k) \geq v_{cutin}) \wedge (u_r(k) < v_n)\} \quad (3.19)$$

$$g_{uc}(corte, nominal) = \{(x_r(k), u_r(k)) \mid u_r(k) < v_{cutoff}\} \quad (3.20)$$

Como não foi instituída lógica de *reset*, tem-se que $R_{xc} = \emptyset$. O fluxograma abaixo explicita o funcionamento do sistema conforme a descrição realizada.

Figura 5 – Fluxograma do modelo híbrido da microturbina eólica



Fonte: Produção do próprio autor.

3.2 Obtenção das matrizes MLD

As matrizes MLD geradas pelo HYSDEL foram obtidas utilizando o código exibido no Apêndice B. Já as matrizes obtidos pelo algoritmo computacional desenvolvido foram geradas em um arquivo *.txt*. Tanto as matrizes geradas pelo HYSDEL quanto as resultantes do código produzido em *Python* são apresentadas integralmente no Apêndice A.

A formulação do algoritmo computacional para a geração das matrizes MLD compreendeu alguns passos. Primeiramente, foram definidas todas as variáveis utilizadas no sistema, conforme descritas na seção anterior. A abordagem escolhida para a declaração de cada variável segmenta cada uma conforme seus tipos, isto é, se é discreta ou contínua, se é uma entrada, uma saída, um estado ou um auxiliar e registrando seus limites superiores e inferiores, caso existam. Posteriormente, elas são concatenadas em vetores segundo a modelagem já apresentada.

Após a definição das variáveis, fez-se necessário descrever o funcionamento do modelo. Para tal, cada equação que o compõe foi incluída em um dos seguintes casos:

- Contínua para discreta
- Discreta para discreta
- Discreta para contínua
- Atualização dos estados contínuos
- Atualização dos estados discretos
- Atualização das saídas contínuas
- Atualização das saídas discretas

Na lógica contínua para discreta, são definidas as equações (3.2a), (3.2b) e (3.2c). Aplicando a lógica descrita nas equações (2.4), (2.5), (2.6), (2.7), e (2.8), encontram-se as inequações. Foi definida a tolerância como ϵ como 10^{-6} , conforme o padrão seguido por Bemporad (2003).

Na lógica discreta para discreta, são definidas as equações (3.2d) e (3.2e). Primeiramente, as equações são convertidas para a sua forma CNF, onde são aplicadas as equações (2.9) e (2.10). Segundamente, as equações são convertidas para a sua forma DNF, onde são aplicadas as equações (2.11) e (2.12).

Na lógica discreta para contínua, são definidas as equações (3.1a), (3.1b), (3.1c), (3.1d), (3.4a), (3.4b), (3.4c) e (3.4d). Dessa maneira, faz-se uso das equações (2.13), (2.14), (2.15), (2.16), (2.17), (2.18), (2.19), (2.20) e (2.21). As inequações obtidas foram agrupadas e as repetidas foram eliminadas, restando apenas registros únicos.

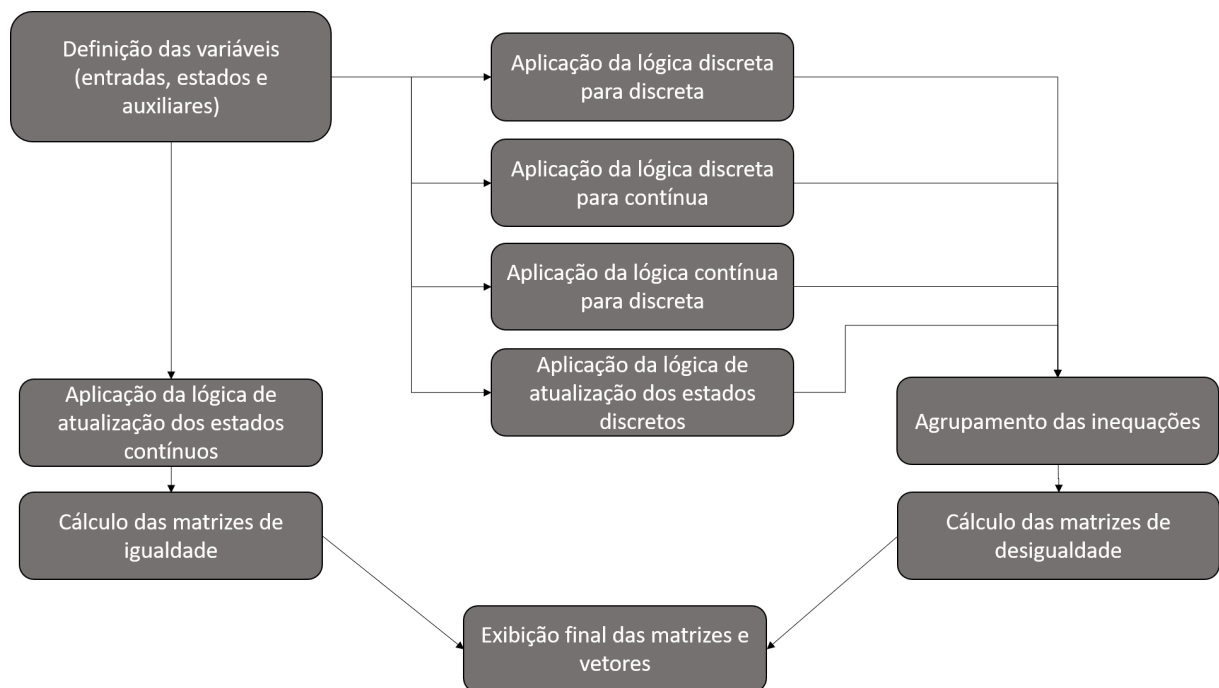
Na lógica de atualização dos estados discretos, são definidas as equações (3.3a), (3.3b), (3.3c) e (3.3d), já na atualização dos estados contínuos, são definidas as equações (3.1a),

(3.1b), (3.1c) e (3.1d). Agrupando cada definição, são preenchidas as matrizes A , B_1 , B_2 e B_2 . Vale ressaltar que o valor associado as vetor contínuo definido na SAS é armazenado no vetor de auxiliares contínuos $z(k)$, sendo utilizado para a construção da matriz B_3 . A matriz B_4 não é utilizada, uma vez que as parcelas constantes estão inclusas nas variáveis z_i .

Como não há saídas, não existem lógicas de atualização das saídas contínuas e discretas para o caso estudado.

Um resumo do processo pode ser visto no fluxograma a seguir:

Figura 6 – Fluxograma do algoritmo para cálculo das matrizes do modelo MLD



Fonte: Produção do próprio autor.

3.3 Simulações do modelo estudado

Para simular o sistema no MATLAB, fez-se necessário definir valores de entrada e do estado inicial. Como valores de entrada, foram utilizadas alguns dos dados utilizados por Silva (2020). No caso, foram selecionados valores de velocidade do vento para a cidade de Santa Vitória do Palmar-RS para um período de 72 horas, com discretização de um valor de velocidade por hora. Essa velocidade foi utilizada como parâmetro na função polinomial construída que retorna a potência de operação da microturbina quando a velocidade está entre v_{cutin} e v_n . Com esses valores foi possível construir os vetores de

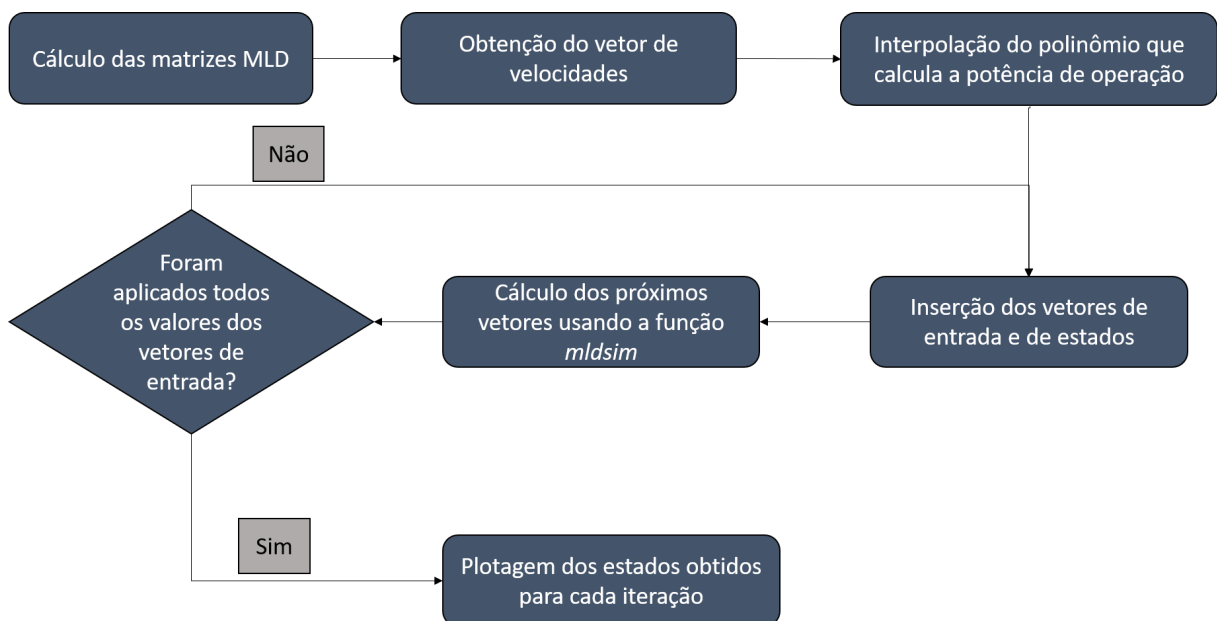
entrada. Já, para o estado inicial, optou-se por partir do ponto em que não há operação, ou seja a potência é zero e o estado lógico verdadeiro é o *parado*.

Primeiramente, foram levadas em conta as matrizes calculadas pelo HYSDEL. Desse modo, os dados de entrada e as matrizes MLD foram utilizadas como parâmetro em uma função do HYSDEL chamada *mldsim* (TORRISI et al., 2011). Assim, a função produz em retorno os próximos vetores do sistema, ou seja, $x(k + 1)$, $z(k + 1)$, $\delta(k + 1)$ e $y(k)$. Esse processo é incluído em um laço de repetição que sempre substituiu o vetor de estados atual pelo produzido na iteração anterior. Logo, pode ser construída uma matriz que contém todos os estados gerados pelo sistema ao longo do tempo.

Em seguida, o processo citado anteriormente foi repetido, contudo, no lugar das matrizes MLD geradas pelo HYSDEL, foram utilizadas as produzidas pelo programa desenvolvido em *Python*.

A seguir, pode ser visto um fluxograma que resume os passos adotados durante o processo de simulação do sistema no MATLAB.

Figura 7 – Fluxograma da simulação da microturbina eólica



Fonte: Produção do próprio autor.

4 RESULTADOS E DISCUSSÕES

4.1 Matrizes MLD

Fazendo uso da aplicação desenvolvida em *Python* e, utilizando como exemplo de sistema, o caso da microturbina eólica, conforme descrito por Silva (2020), foram extraídas as matrizes de igualdade ($A, B1, B2, B3, B4, B5, C, D1, D2, D3, D4$ e $D5$) e desigualdade ($E1, E2, E3, E4$ e $E5$), do sistema MLD correspondente. As matrizes podem ser encontradas no Apêndice A.

Analisando cada matriz, nota-se que as matrizes de igualdade geradas pelo HYSDEL são exatamente iguais às produzidas pelo algoritmo computacional desenvolvido em *Python*. Já, no que tange às de desigualdade, percebe-se que as ambas possuem as mesmas linhas, porém ordenadas de maneira diferente. Diferentemente das matrizes $A, B1, B2, B3, B4, B5, C, D1, D2, D3, D4$ e $D5$, a posição das linhas das matrizes $E1, E2, E3, E4$ e $E5$ não altera o funcionamento do sistema. Isso ocorre uma vez que as matrizes de desigualdade agrupam os coeficientes de inequações, transformando uma série de relações de desigualdade em uma inequação matricial. Em ambos os casos, foram encontradas matrizes com 58 linhas, isto é, existem 58 inequações onde $E1$ possui os coeficientes do vetor $u(k)$, $E2$ os do vetor $\delta(k)$, $E3$ os do vetor $z(k)$, $E4$ os do vetor $x(k)$ e $E5$ contém a parte estática da inequação.

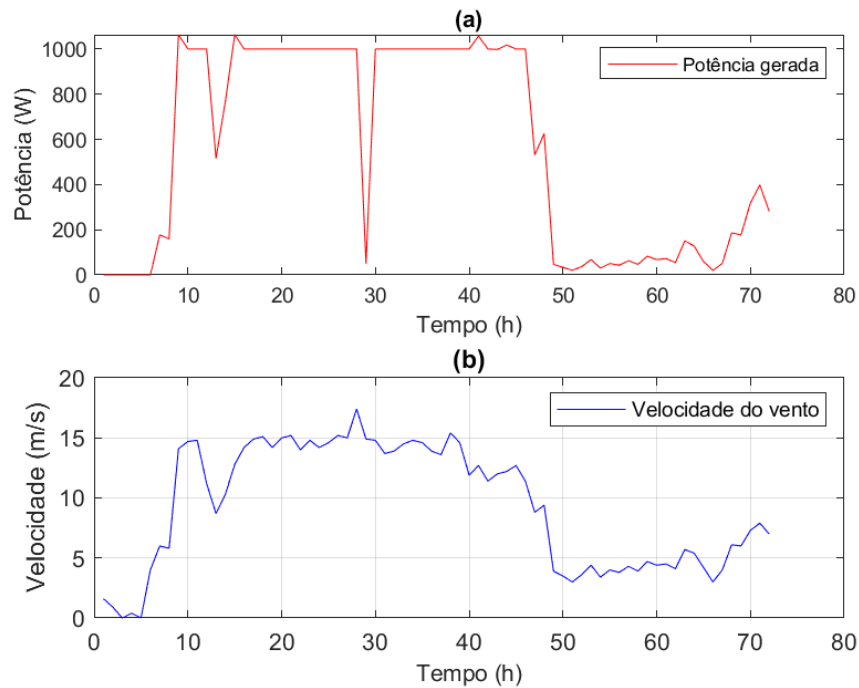
No Apêndice A, além das matrizes MLD produzidas, existe uma tabela que faz a associação de cada linha das matrizes de desigualdade geradas no programa em *Python* às obtidas por meio do HYSDEL. A permutação das linhas de cada matriz de desigualdade seguindo os índices da tabela resulta em grupos de matrizes iguais. Além disso, no Apêndice B, é exibido o código utilizado no HYSDEL para obtenção das matrizes MLD.

4.2 Simulações do modelo estudado

Foram gerados gráficos com auxílio do MATLAB que exibem os valores da potência gerada e da velocidade do vento para a faixa amostral determinada de 72h para as matrizes produzidas pelo algoritmo próprio e pelo HYSDEL. Além disso, também foram construídos gráficos que apontam o nível lógico em que cada estado discreto do sistema se encontra.

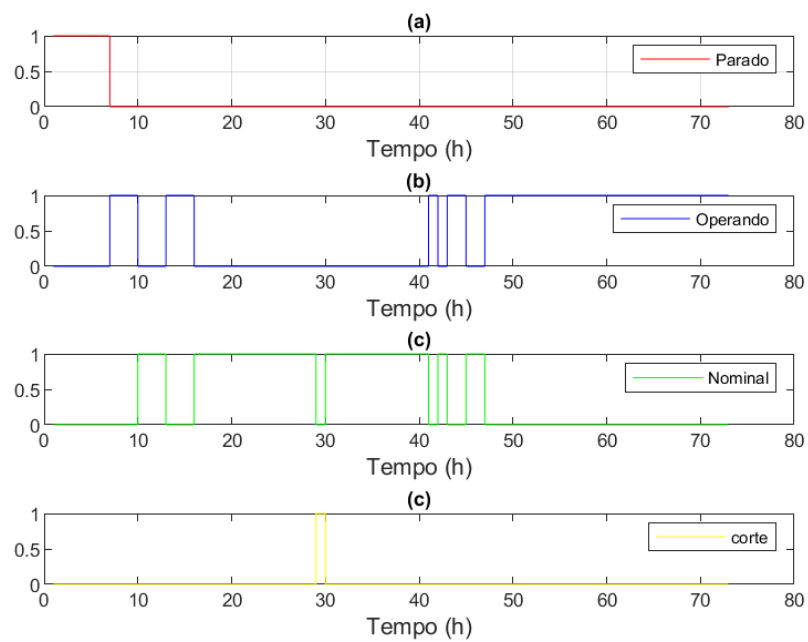
A seguir, podem ser vistos os resultados obtidos a partir das matrizes provenientes do algoritmo próprio desenvolvido em *Python*.

Figura 8 – Potência gerada e velocidade do vento - modelo gerado pelo algoritmo próprio



Fonte: Produção do próprio autor.

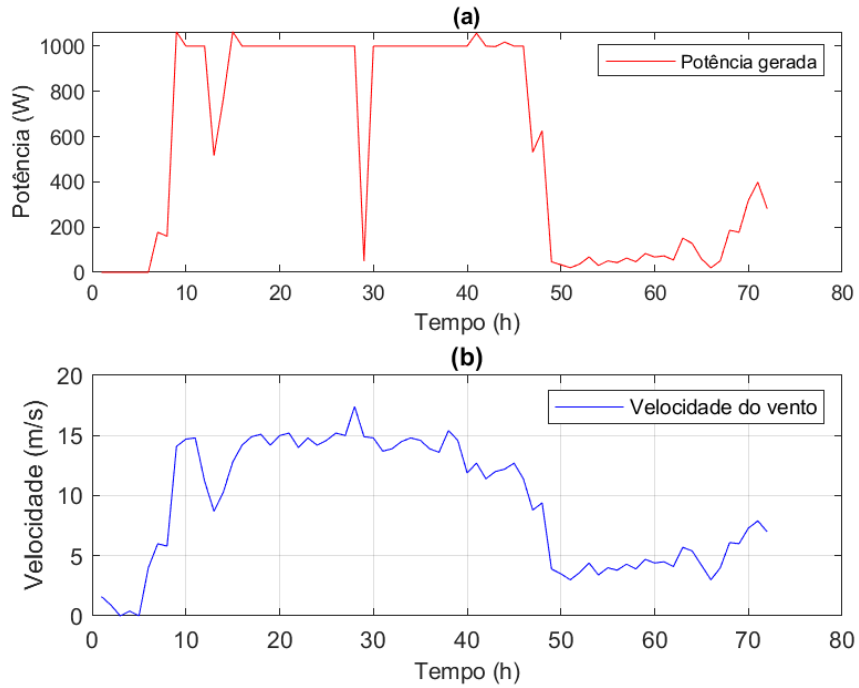
Figura 9 – Estados discretos - modelo gerado pelo algoritmo próprio



Fonte: Produção do próprio autor.

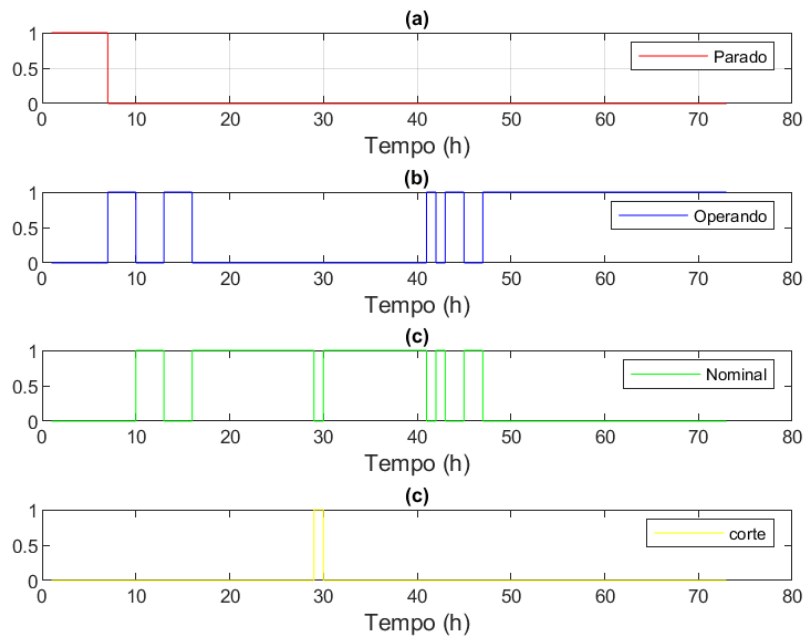
Em seguida, são apresentados os gráficos resultantes das matrizes geradas pelo HYSDEL.

Figura 10 – Potência gerada e velocidade do vento - modelo gerado pelo HYSDEL



Fonte: Produção do próprio autor.

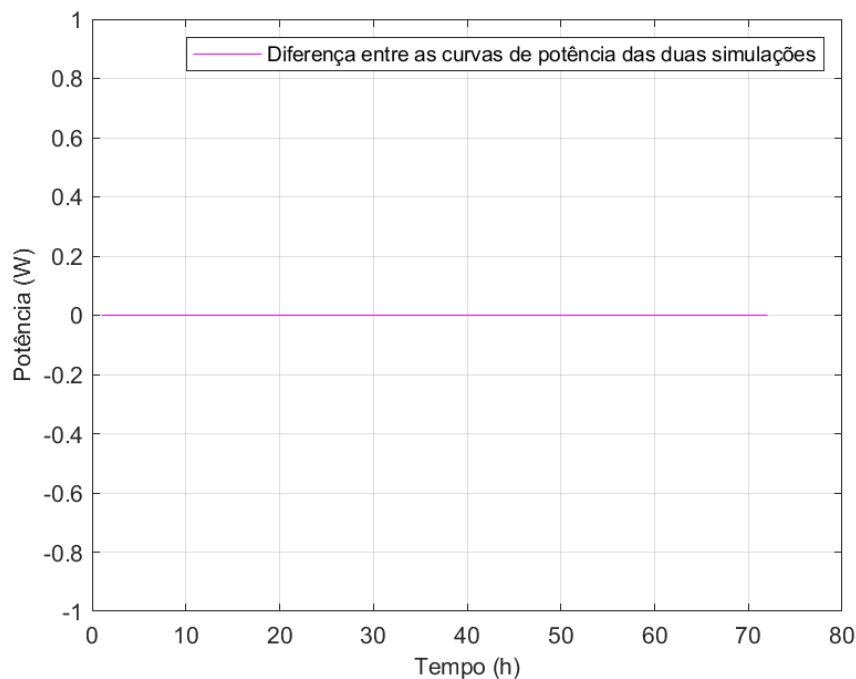
Figura 11 – Estados discretos - modelo gerado pelo HYSDEL



Fonte: Produção do próprio autor.

Evidencia-se que ambos os pares de gráficos são semelhantes. De fato, o resultado é esperado, uma vez que as matrizes de igualdade são iguais nos dois métodos de geração do modelo MLD. Já, em relação às de desigualdade, como explicado na seção anterior, ainda que, comparando as matrizes produzidas pelos dois métodos, haja uma diferença na posição das linhas, os parâmetros definidos nos modelos são os mesmos. A igualdade dos resultados obtidos para cada caso pode ser comprovada no gráfico seguinte, que mostra que o erro absoluto entre as duas curvas de potência é igual a zero.

Figura 12 – Erro absoluto entre cada curva de potência simulada



Fonte: Produção do próprio autor.

Fazendo uma análise do comportamento das curvas de potência gerada, velocidade do vento e dos estados discretos é possível fazer algumas observações. Nota-se que a turbina inicia-se no estado *parado*. À medida que a velocidade cresce, ela passa a atuar no estado *operando*, onde sua potência varia segundo uma função polinomial e opera na sua máxima eficiência. Quando a velocidade atinge a sua faixa nominal, ou seja, entre 12/s e 16 m/s, a potência gerada é a nominal (1000 W). Observa-se que a velocidade cai abaixo dessa faixa e, depois, retorna à mesma, estabilizando no estado *nominal* durante um período longo de tempo.

Após 28h, a velocidade ultrapassa os 16 m/s, o que faz com que ela opere no estado *corte* e, portanto, o valor da potência gerada decresça para 50 W. Um ponto importante a ser notado é a defasagem de um passo entre a potência gerada e a velocidade. Por exemplo, apesar da velocidade romper o limite da faixa de operação nominal no ponto 28h, a queda

da potência gerada só ocorre no ponto 29h. Isso é resultado da modelagem MLD adotada para a microturbina. Em aplicações em que se busque visualizar o impacto das entradas em relação às entradas de maneira mais rápida, pode-se diminuir o passo, aumentando a discretização.

Após o retorno à operação no estado *nominal* e, passadas algumas horas de oscilação, a velocidade sofre uma queda, fazendo com que o sistema fique no estado *operando* até o final do período amostral.

5 CONCLUSÕES E PROJETOS FUTUROS

A modelagem de sistemas é vital para o desenvolvimento de pesquisas científicas e atividades de engenharia. Com a complexidade crescente de muitos desses sistemas e maior interação entre si, é preciso lançar mão de modelos que possibilitem a integração de múltiplas estruturas e possam lidar com variáveis distintas. Em meio a esse quadro, as teorias de modelagens que mesclam lógicas discretas e contínuas têm uma grande gama de aplicação. Cabe, portanto, saber lidar com diferentes modelos e identificar a utilidade de cada um conforme a situação. Neste trabalho, foram estudados os modelos DHA e MLD. Mais especificamente, foi pesquisado o método de conversão do primeiro no segundo. Para tal, além do estudo teórico das modelagens, fez-se uso do HYSDEL como exemplo de aplicação que realiza a conversão almejada para propósitos de comparação e embasamento. Foi possível, portanto, identificar os padrões e passos matemáticos utilizados para geração de matrizes MLD e transcrevê-los com sucesso para um programa de código aberto em *Python*. Os resultados obtidos foram de encontro aos retornados pelo HYSDEL, o que pode ser visto como uma validação do processo. Em suma, em se tratando dos objetivos propostos pelo trabalho, afirma-se que o desenvolvimento de um programa em código aberto que converta o modelo DHA em MLD é promissor e apresenta resultados pertinentes.

Como trabalhos futuros, primeiramente, o algoritmo deve ser generalizado para ser capaz de calcular as matrizes do modelo MLD para qualquer sistema definido pelo usuário, visto que, apesar das metodologias desenvolvidas neste projeto, houve um foco no sistema da microturbina eólica. Ademais, sugere-se ir além do desenvolvimento de um *software* que converta o modelo DHA em MLD. O HYSDEL apresenta uma série de funcionalidades extras que podem ser utilizadas para aprimorar o estudo de sistemas híbridos, como funções próprias para simulação e otimização. Além disso, existem variadas edições da ferramenta, cabendo analisar os avanços de cada uma e, se possível, também implementá-las. Outro ponto importante é a questão de interface com o usuário. A aplicação desenvolvida em *Python* mostrou-se capaz de produzir as matrizes desejadas. Entretanto, é possível ainda definir uma sintaxe ainda mais robusta que facilite a interação com o utilizador da ferramenta e possibilite o registro de uma variedade de informações sobre o sistema descrito. Finalmente, é possível estudar com mais profundidade a generalização posta por Hejri, Giua e Mokhtari (2018). Dessa maneira, pode-se criar um algoritmo que permita a geração de matrizes, não apenas MLD, como também, EMLD e incluir as lógicas de *reset* e definição de parâmetros, que não foram utilizadas neste trabalho. Assim, pode-se comparar a eficiência e o ganho das matrizes MLD e EMLD e simular sistemas mais complexos, que apresentem a necessidade da lógica de *reset* ser implementada.

REFERÊNCIAS

ANTSAKLIS, P. J. A brief introduction to the theory and applications of hybrid systems. Proceedings of the IEEE, v. 88, n. 7, p. 879887, 2000., 2000. Citado na página 13.

BEMPORAD, A. Modeling, Control, and Reachability Analysis of Discrete-Time Hybrid Systems. 2003. Disponível em: <<http://www.dii.unisi.it/~bemporad>>. Citado 6 vezes nas páginas 17, 18, 20, 21, 22 e 35.

BEMPORAD, A.; MORARI, M. Control of systems integrating logic, dynamics, and constraints 1. 1999. 407-427 p. Citado 4 vezes nas páginas 13, 18, 19 e 29.

DICKERSON, C. E.; MAVRIS, D. A Brief History of Models and Model Based Systems Engineering and the Case for Relational Orientation. 2013. Disponível em: <<http://www.uml.org/#Links-Tutorials>> Citado na página 12.

ENERSUD. GERAR 246 instructions manual. 2017. Disponível em: <<http://www.enersud.com.br/produtos/turbina-eolica-gerar-246/>>. Citado na página 28.

HEEMELS, W. W. P. M. H. Linear complementarity systems : a study in hybrid dynamics. [S.l.]: Technische Universiteit Eindhoven, 1999. ISBN 9038616902. Citado na página 18.

HEJRI, M.; GIUA, A.; MOKHTARI, H. On the complexity and dynamical properties of mixed logical dynamical systems via an automaton-based realization of discrete-time hybrid automaton. International Journal of Robust and Nonlinear Control, John Wiley and Sons Ltd, v. 28, p. 4713–4746, 11 2018. ISSN 10991239. Citado 7 vezes nas páginas 15, 22, 23, 24, 28, 29 e 43.

HOOS, H. H.; STÜTZLE, T. 6 - propositional satisfiability and constraint satisfaction. In: HOOS, H. H.; STÜTZLE, T. (Ed.). Stochastic Local Search. San Francisco: Morgan Kaufmann, 2005, (The Morgan Kaufmann Series in Artificial Intelligence). p. 257–312. ISBN 978-1-55860-872-6. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B9781558608726500238>>. Citado na página 25.

JOHANSSON, K. H.; LYGEROS, J.; SASTRY, S. MODELING OF HYBRID SYSTEMS. 2004. Citado 2 vezes nas páginas 17 e 18.

LIAN, J.; LIU, S.; LI, L.; LIU, X.; ZHOU, Y.; YANG, F.; YUAN, L. A mixed logical dynamical-model predictive control (mld-mpc) energy management control strategy for plug-in hybrid electric vehicles (phevs). Energies, MDPI AG, v. 10, 2017. ISSN 19961073. Citado na página 14.

LUNZE, J.; LAMNABHI-LAGARRIGUE, F. F. Handbook of hybrid systems control : theory, tools, applications. [S.l.]: Cambridge University Press, 2009. 565 p. ISBN 9780521765053. Citado 2 vezes nas páginas 12 e 13.

OGATA, K. Engenharia de Controle Moderno - Katsuhiko Ogata - 5 Edição. [S.l.]: Pearson, 2011. Citado na página 12.

PESIN, M. "U.S. Department of Energy Electricity Grid Research and Development". 2017. Citado na página 14.

RASKIN, J.-F. An introduction to hybrid automata. In: _____. Handbook of Networked and Embedded Control Systems. Boston, MA: Birkhäuser Boston, 2005. p. 491–517. ISBN 978-0-8176-4404-8. Disponível em: <https://doi.org/10.1007/0-8176-4404-0_21>. Citado na página 18.

ROSSUM, G. V.; JR, F. L. D. Python reference manual. [S.l.]: Centrum voor Wiskunde en Informatica Amsterdam, 1995. Citado na página 28.

SCHUTTER, B. D.; BOOM, T. V. D. On Model Predictive Control for Max-Min-Plus-Scaling Discrete Event Systems. 2000. Citado na página 18.

SCHUTTER, B. D.; MOOR, B. D. The extended linear complementarity problem and the modeling and analysis of hybrid systems. MATHEMATICAL PROGRAMMING, v. 71, p. 289–325, 1997. Citado na página 18.

SILVA, D. D. P. E. UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO CENTRO TECNOLÓGICO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA GERENCIAMENTO DE UMA MICRORREDE UTILIZANDO CONTROLE PREDITIVO COM INCERTEZAS METEOROLÓGICAS. 2020. Citado 8 vezes nas páginas 14, 15, 28, 29, 30, 36, 38 e 72.

SONTAG, E. Nonlinear regulation: The piecewise linear approach. IEEE Transactions on Automatic Control, v. 26, n. 2, p. 346–358, 1981. Citado na página 18.

TORRISI, F. D.; BEMPORAD, A. Hysdel - a tool for generating computational hybrid models for analysis and synthesis problems. IEEE Transactions on Control Systems Technology, v. 12, p. 235–249, 3 2004. ISSN 10636536. Citado 3 vezes nas páginas 13, 15 e 23.

TORRISI, F. D.; BEMPORAD, A.; BERTINI, G.; HERTACH, P.; JOST, D.; MIGNONE, D. HYSDEL 2.0.6 User Manual. 2011. Citado 3 vezes nas páginas 23, 28 e 37.

WELLSTEAD, P. E. Introduction to physical system modelling. [S.l.]: Academic Press, 1979. 279 p. ISBN 0127443800. Citado na página 12.

WITSENHAUSEN, H. A class of hybrid-state continuous-time dynamic systems. IEEE Transactions on Automatic Control, v. 11, n. 2, p. 161–167, 1966. Citado na página 12.

ZHANG, Y.; CHU, X.; LIU, Y.; LIU, Y. A modelling and control approach for a type of mixed logical dynamical system using in chilled water system of refrigeration system. Mathematical Problems in Engineering, Hindawi Limited, v. 2019, 2019. ISSN 15635147. Citado na página 14.

Apêndices

APÊNDICE A – MATRIZES DA MODELAGEM MLD

Neste apêndice são mostradas as matrizes MLD calculadas para o sistema estudado.

A.1 Matrizes geradas pelo HYSDEL

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$B2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B3 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$C = []$$

$$D1 = []$$

$$D2 = []$$

$$D3 = []$$

$$D4 = []$$

$$D5 = []$$

$$E2 = \begin{bmatrix} -48.000001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -38.000001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -34.000001 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 16 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$E4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1500 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1500 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

$$E4 = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1500 & 0 \\ 0 & 0 & 0 & 1500 & 0 \\ 0 & 0 & 0 & 0 & 50 \\ 0 & 0 & 0 & 0 & 50 \end{bmatrix}$$

A.2 Matrizes geradas pelo algoritmo próprio

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$B2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B3 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$C = []$$

$$D1 = []$$

$$D2 = []$$

$$D3 = []$$

$$D4 = []$$

$$D5 = []$$

$$E2 = \begin{bmatrix} -48.000001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -38.000001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -34.000001 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 16 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

$$E2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$E4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1500 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1500 & 0 & 0 \\ 0 & 0 & 0 & -1500 & 0 \\ 0 & 0 & 0 & 1500 & 0 \\ 0 & 0 & 0 & 0 & 50 \\ 0 & 0 & 0 & 0 & 50 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & -1 & 0 & 0 & 0 \end{bmatrix}$$

$$E4 = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$E5 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

A.3 Equivalência das matrizes E

Nas tabela abaixo, é mostrada a equivalência das matrizes E geradas pelo HYSDEL e pelo programa desenvolvido. Como essas matrizes são construídas através dos coeficientes de uma lista de inequações, a ordem das linhas de cada matriz não altera seu uso, contanto que a permutação entre as linhas das matrizes ocorra para todas as matrizes E ($E1$, $E2$,

$E3$, $E4$ e $E5$).

Tabela 1 – Associação entre as linhas das matrizes E geradas pelo HYSDEL e as geradas pelo programa desenvolvido em *Python*.

Linha nas matrizes geradas em Python	Linha nas matrizes geradas pelo HYSDEL
1	1
...	...
12	12
13	53
14	54
15	13
16	14
17	15
18	16
19	55
20	56
21	57
22	58
23	19
24	20
25	17
26	18
27	21
28	22
29	24
30	26
31	23
32	25
33	33
34	31
35	34
36	32
37	29
38	27
39	30
40	28
41	36
42	35
43	38
44	37
45	45
46	43
47	46
48	44
49	41
50	39
51	42
52	40
53	47
54	48
55	49
56	50
57	52
58	51

Fonte: Produção do próprio autor

APÊNDICE B – MODELAGEM DO SISTEMA NO HYSDEL

Neste apêndice, são mostrados os códigos utilizados pelo HYSDEL para a geração das matrizes MLD. Os mesmos foram baseados no trabalho desenvolvido por Silva (2020).

```

1
2 SYSTEM eolico {
3   INTERFACE {
4     STATE {
5       /* Potência da microturbina eólica em W */
6       REAL Pw [0, 1500];
7       /* Estados de operação da turbina eólica */
8       BOOL parado, operando, nominal, falha; }
9
10    INPUT {
11      /* Potência de operação calculada externamente em W */
12      REAL Pg [0, 1500];
13      /* Velocidade do vento em m/s */
14      REAL vel [0, 50];
15      }
16  }
17
18  IMPLEMENTATION {
19    AUX {
20      /* Deltas auxiliares */
21      BOOL d1, d2, d3, op, pn;
22      /* Variáveis Z */
23      REAL xnominal, xfail, xo, xop;}
24
25    AD {
26      d1 = vel >= 2; /* Partida da microturbina */
27      d2 = vel >= 12; /* Início da operação nominal turbina */
28      d3 = vel >= 16; /* Operação com potência residual */
29      }
30
31    LOGIC {
32      op = d1 & ~d2;
33      pn = d2 & ~d3;
34      }
35
36    AUTOMATA {
37      parado = (~d1 & operando) | (parado & ~d1) | (falha & ~d1) | (
nominal & ~d1);
38      operando = (operando & op) | (parado & d1) | (nominal & ~d2) | (
falha & op);
39      nominal = (nominal & pn) | (operando & d2) | (falha & ~d3) | (
parado & pn);

```

```
40     falha = (falha & d3) | (nominal & d3) | (operando & d3) | (parado
41     & d3);
42     }
43     DA {
44         xo = {IF parado THEN
45             0
46             ELSE 0};
47
48         xop = {IF operando THEN
49             Pg
50             ELSE 0};
51
52         xnominal = {IF nominal THEN
53             1000
54             ELSE 0};
55         xfail = {IF falha THEN
56             50
57             ELSE 0};
58     }
59
60     CONTINUOUS{
61         x = xop + xfail + xo + xnominal;
62     }
63
64 }
65 }
```